

Oracle Financial Services Crime and Compliance Studio

Installation Guide

Release 8.0.8.0.0

September 2020

E91246-01

ORACLE
Financial Services

OFS Crime and Compliance Studio Kubernetes Installation Guide

Copyright © 2020 Oracle and/or its affiliates. All rights reserved.

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

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

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

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are “commercial computer software” pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

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

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

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

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

For information on third party licenses, click [here](#).

Document Control

Version Number	Revision Date	Change Log
1.0	September 2020	Revamp

Table of Contents

1	Preface	7
1.1	Audience	7
1.2	Access to Oracle Support	7
1.3	Related Documents	7
1.4	Conventions	7
1.5	Abbreviations	8
2	Introduction	9
2.1	Installation Check List when Studio is Deployed with OFSAA	9
2.2	Installation Check List when Studio is Deployed without OFSAA	10
3	Pre-installation	12
3.1	Prerequisites	12
3.2	Hardware and Software Requirements	13
3.3	Prerequisite Environmental Settings	14
3.4	Download the Big Data Files	15
3.5	Validation Checklist	16
3.6	Configure the Elastic Search Component	17
3.7	Configure the Interpreter Settings	19
3.7.1	Configure the fcc-jdbc Interpreter	20
3.7.2	Configuring the fcc-ore Interpreter	21
3.7.3	Configure the fcc-python Interpreter	21
3.7.4	Configure the Spark Interpreter	22
3.7.5	Configure the PySpark Interpreter	24
3.8	Create the Studio Schema	27
3.9	Assign Grants for the Database Schemas	27
4	Setup Password Stores with Oracle Wallet	28
4.1.1	Setup the Password Stores for Database User Accounts	28
4.1.2	Verify the Connectivity of the Wallet	31
4.2	Create the Credential Keystore	31
4.2.1	Copying and Adding Files	32

4.3	Download the Installer Kit	32
5	Installation.....	33
5.1	Extract the Installer Kit	33
5.2	Place Files in the Installation Directories	33
5.3	Generate an Encrypted Password	34
5.4	Generate the Public and Private Keys	34
5.5	Apply GDPR and Redaction Changes for FCC Studio.....	35
5.5.1	<i>Generate the Key Store File for Secure Batch Service.....</i>	35
5.5.2	<i>Add the Batch Service (SSL) to PGX Trust Store</i>	35
5.6	Configure the Preferred Services	36
5.7	Configure the studio-env.yml File.....	37
5.8	Configure the Extract Transfer and Load (ETL) Process	64
5.8.1	<i>Configure the Extract Transfer Load (ETL) Services</i>	64
5.9	Verify the Resource Allocation for FCC Studio Services	67
5.10	Resource Limits	67
5.11	Resource Types	68
5.12	Resource Parameters in FCC Studio.....	68
5.13	Deploy FCC Studio on the K8s Cluster	76
6	5 Post-installation Steps when OFSAA is Installed	77
6.1	Verify the Installation	77
6.1.1	<i>Check Logs of FCC Studio Services.....</i>	77
6.2	Configure the Schema Creation	80
6.2.1	<i>Configure the Schema Creation from the FCC Studio Server.....</i>	80
6.2.2	<i>Configure the Schema Creation from the OFSAA Server.....</i>	80
6.3	Configure the ICIJ Data.....	81
6.4	Restart the PGX Service	83
6.5	Access the FCC Studio Application	83
6.6	Perform the OFSAA Configuration for Batch Execution	84
6.7	Configure and Run Published Notebooks.....	85
7	Post-installation Steps when OFSAA is Not Installed.....	86
7.1	Verify the Installation	86
7.1.1	<i>Checking Logs of FCC Studio Services.....</i>	86

7.2	Restart the PGX Service	89
7.3	Access the FCC Studio Application	89
8	Upgrade	91
8.1	Cleanup for Upgrade	91
8.1.1	<i>Perform Extract Transfer and Load (ETL) Cleanup</i>	92
8.2	Upgrade Steps with OFSAA.....	92
8.2.1	<i>Start the FCC Studio Application</i>	93
8.3	Upgrade Steps without OFSAA.....	94
8.4	Configure the studio-env.yml File for Upgrade.....	95
9	Reinstall FCC Studio.....	96
9.1.1	<i>Cleanup for Studio Schema</i>	96
9.1.2	<i>Cleanup for BD or ECM Atomic Schema</i>	97
9.1.3	<i>Cleanup for BD or ECM Config Schema</i>	97
10	Appendix - Starting and Stopping Services.....	98
10.1	Start a Service	98
10.2	Stop a Service.....	98
11	Appendix A - Tables and Sequences.....	99
11.1	Studio Schema Tables.....	99
11.2	Studio Schema Sequences.....	100
12	Appendix B - Using Python Virtual Environments with PySpark	101
12.1	Create a Virtual Environment with Conda.....	101
12.2	Include a Virtual Environment in the Init Container	101
12.3	Update the Interpreter Properties.....	102
13	Frequently Asked Questions (FAQs).....	103

1 Preface

This section provides information about the Oracle Financial Services (OFS) Crime and Compliance Studio (FCC Studio) Installation and Configuration Guide with kubernetes deployment.

1.1 Audience

OFS FCC Studio Installation and Configuration Guide is intended for System Engineers who are responsible for installing and maintaining the application.

This document assumes that you have experience in installing Enterprise components and basic knowledge about the following:

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

1.2 Access to Oracle Support

OFS FCC Studio Installation and Configuration Guide is intended for System Engineers who are responsible for installing and maintaining the application.

Oracle customers have access to electronic support through [My Oracle Support](#). For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>.

Or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing-impaired.

1.3 Related Documents

We strive to keep this and all other related documents updated regularly; visit the [OHC Documentation Library](#) to download the latest version available there. The list of related documents is provided here.

- *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 Conventions

The following text conventions are used in this document.

Table 1: Document Conventions

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

1.5 Abbreviations

The following table lists the abbreviations used in this document.

Table 2: 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
FCDM	Financial Crime Data Model
ICIJ	International Consortium of Investigative Journalists
IDCS	Oracle Identity Cloud Service
SSO	Single Sign-On
SSH	Secure Shell

2 Introduction

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 (v8.0.8.0.0) of FCC Studio can be used for the following:

- To install a new instance of FCC Studio as follows:
- FCC Studio with OFSAA (Oracle Financial Services Analytical Application). Here, with OFSAA is with Behavior Detection (BD) or Enterprise Case Management (ECM).
- FCC Studio without OFSAA
- To upgrade an existing instance of FCC Studio on the K8s cluster as follows:
- Upgrade FCC Studio from v8.0.7.1.0 onwards to v8.0.8.0.0 with OFSAA.

Topics:

- [Installation Check List when Studio is Deployed with OFSAA](#)
- [Installation Check List when Studio is Deployed without OFSAA](#)

2.1 Installation Check List when Studio is Deployed with OFSAA

To complete the installation process, you must perform the steps listed in the following Checklist. Use this checklist to verify whether these steps are completed or not. Click the reference link to go to the topic.

Table 3: Installation Check List

Sl. No.	Activity
Pre-installation Steps	
1	Install all the prerequisite hardware and software.
2	Setup the environmental settings.
3	Download the files for Big Data
4	Configure the Elastic Search Component
5	Configure the Interpreter Settings
6	Create the Studio Schema
7	Assign Grants for the Database Schemas
8	Setup Password Stores with Oracle Wallet.
9	Create the Credential Keystore
10	Download the installer kit.
Installation Steps	
1	Extract the installer kit.

2	Place files in the Installation Directories
3	Generate an Encrypted Password
4	Generate the Public Key and Private Key
5	Generate the Key Store File for Secure Batch Service
6	Add the Batch Service (SSL) to PGX Trust Store
7	Configure the Preferred Services
8	Configure the studio-env.yml File
9	Configure the Extract Transfer and Load (ETL) Process
10	Verify the Resource Allocation for FCC Studio Services
11	Deploy FCC Studio on the K8s Cluster
Post-Installation Steps	
1	Verify the Installation
2	Configure the Schema Creation
3	Configure the ICIJ Data
4	Access the FCC Studio Application
5	Perform the OFSAA Configuration for Batch Execution
6	Configure and Run Published Notebooks

2.2 Installation Check List when Studio is Deployed without OFSAA

To complete the installation process, you must perform the steps listed in the following Checklist. Use this checklist to verify whether these steps are completed or not. Click the reference link to go to the topic.

Table 4: Installation Check List

Sl. No.	Activity
Pre-installation Steps	
1	Install all the prerequisite hardware and software.
2	Setup the environmental settings.
3	Download the files for Big Data
4	Configure the Elastic Search Component
5	Configure the Interpreter Settings
6	Create the Studio Schema
7	Assign Grants for the Database Schemas
8	Setup Password Stores with Oracle Wallet.

9	Create the Credential Keystore
10	Download the installer kit.
Installation Steps	
1	Extract the installer kit.
2	Place files in the Installation Directories
3	Generate an Encrypted Password
4	Generate the Public Key and Private Key
5	Generate the Key Store File for Secure Batch Service
6	Add the Batch Service (SSL) to PGX Trust Store
7	Configure the Preferred Services
8	Configure the studio-env.yml File
9	Configure the Extract Transfer and Load (ETL) Process
10	Verify the Resource Allocation for FCC Studio Services
11	Deploy FCC Studio on the K8s Cluster
Post-Installation Steps	
1	Verify the Installation
2	Access the FCC Studio Application

3 Pre-installation

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

Topics:

- [Prerequisites](#)
- [Hardware and Software Requirements](#)
- [Prerequisite Environmental Settings](#)
- [Download the Big Data Files](#)
- [Validation Checklist](#)
- [Configure the Elastic Search Component](#)
- [Configure the Interpreter Settings](#)
- [Create the Studio Schema](#)
- [Assign Grants for the Database Schemas](#)
- [Setup Password Stores with Oracle Wallet](#)
- [Create the Credential Keystore](#)
- [Download the Installer Kit](#)

3.1 Prerequisites

The Linux machine must satisfy the following conditions:

- BD (Behavior Detection) or the ECM (Enterprise Case Management) application pack must be 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 for this version of FCC Studio. The installation environment or setup must have these requirements for an application to run smoothly and efficiently.

Table 5: Hardware and Software Requirements

Hardware or Software Category	Component Version
Browser	<ul style="list-style-type: none"> • Chrome 57.x • Firefox 52.x
Java Version	Java 8
Docker Registry	<ul style="list-style-type: none"> • Docker registry must be present to store docker images • Min of 45GBspace is required to save docker images
Database Server	<ul style="list-style-type: none"> • 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-7.6.2
Elastic Search Hadoop Jars	ES - Hadoop v7.6.2 You must download the ZIP file from https://www.elastic.co/downloads/past-releases/elasticsearch-apache-hadoop-7-6-2
Kubernetes Cluster	
Processing Server	<ul style="list-style-type: none"> • RHEL 7.4+ • SFTP • Oracle JRE Standard Edition 1.8.x(with JCE) • Kubernetes(K8s) cluster
PGX (Graph) Server	<ul style="list-style-type: none"> • RHEL 7.4+ • Kubernetes(K8s) cluster • Minimum gcc library version 4.8.2
Big Data	

Cloudera Distribution Hadoop 6.x	<ul style="list-style-type: none"> • CDH Version 6.3.1 • Apache Hadoop Version 3.0.0 • Apache Hive Version 2.1.1 • Apache Spark Version 2.4.0 • Apache Sqoop Version 1.4.7 <p>The <code>.profile</code> file must be present with the <code>SPARK_HOME</code> and <code>PYTHON_HOME</code> parameters already set.</p>
Cloudera Hive Connectors	Hive JDBC Connectors V 2.5.15
Hadoop Security Protocol	<ul style="list-style-type: none"> • Kerberos R release 1.6.1 • Sentry-1.4.0

NOTE If Cloudera is not installed, the following versions are required:

- Hadoop Version 3.0.0
- Hive Version 2.1.1
- Spark Version 2.4.0
- Sqoop Version 1.4.7

3.3 Prerequisite Environmental Settings

The following prerequisite environmental settings must be set before beginning the installation of FCC Studio. These settings are the configuration that a system must have for an application to run smoothly and efficiently.

Table 6: Prerequisite Environmental Settings

Category	Expected Value
Java Settings	<p><code>PATH</code> in the <code>.profile</code> file must be set to include <code>kubectl</code> and the Java Runtime Environment (Java 8) absolute path.</p> <p>NOTE: Ensure the absolute path to <code>JRE/bin</code> is set at the beginning of the <code>PATH</code> variable.</p> <p>For example: <code>PATH=/usr/java/jre1.8/bin:\$PATH</code></p> <p>Ensure no <code>SYMBOLIC</code> links to Java installation are set in the <code>PATH</code> variable.</p>

Oracle Database Settings	<p>NOTE:</p> <p>This setting is required only if the Wallet has to be created on the same server as that of the FCC Studio server.</p> <p>Oracle Processing Server</p> <ul style="list-style-type: none"> • ORACLE_HOME must be set in the <code>.profile</code> file pointing to the appropriate Oracle DB Client installation. • PATH in the <code>.profile</code> file must be set to include the appropriate <code>\$ORACLE_HOME/bin</code> directory.
Download Directory	Indicates the directory where the product installer zip file is downloaded or copied. The user permission must be set to 755 for this 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 directory. <p>NOTE:</p> <p>The Installation and the Download Directory can be the same if the product installer zip file is not copied separately to another directory.</p>
OS Locale	Linux: <code>en_US.utf8</code> Execute the following command to check the locale: <code>locale -a grep -i 'en_US.utf'</code> The locale is displayed.

3.4 Download the Big Data Files

Download the following configuration files from the Cloudera installation setup:

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

Table 7: Required File Structure

File Category	File Names
Hadoop Cluster	<ul style="list-style-type: none"> • <code>core-site.xml</code> • <code>hadoop-env.sh</code> • <code>hdfs-site.xml</code> • <code>log4j.properties</code> • <code>ssl-client.xml</code> • <code>topology.map</code> • <code>topology.py</code>

Kerberos Files	<ul style="list-style-type: none"> • <code>krb5.conf</code> • keytab file name as mentioned in the <code>config.sh</code> file. <p>NOTE: Ensure to rename your <code>.keytab</code> file to <code>fccstudio.keytab</code>.</p>
Additional Jars	<ul style="list-style-type: none"> • <code>hive-exec-*.jar</code>. For example, <code>hive-exec-1.1.0-cdh5.13.0.jar</code>. • <code>HiveJDBC4.jar</code>. • <code>hive-metastore-*.jar</code>. For example, <code>hive-metastore-1.1.0-cdh5.13.0</code>. • <code>hive-service-*.jar</code>. For example, <code>hive-service-1.1.0-cdh5.13.0.jar</code>. <p>NOTE:</p> <ul style="list-style-type: none"> • The version of the jars is client or user-specific. These jars can be obtained from the existing jars of the Cloudera installation. • The <code>HiveJDBC4.jar</code> file is not available in the Cloudera installation setup. You must download the same from the Cloudera website.
ES-Hadoop Jars	<p><code>elasticsearch-spark-20_2.11-7.6.2.jar</code></p> <p>To download the <code>elasticsearch-spark-20_2.11-7.6.2.jar</code> file, follow these steps:</p> <ol style="list-style-type: none"> 1. Download the ZIP file from https://www.elastic.co/downloads/past-releases/elasticsearch-apache-hadoop-7-6-2. 2. Extract the downloaded file. 3. Navigate to the <code>dist</code> directory and download the <code>elasticsearch-spark-20_2.11-7.6.2.jar</code> file.

3.5 Validation Checklist

The Validation Checklist section provide you the information about the parameters that you can validate to avoid installation issues. This section explains the validation and actions that can be taken for some of the common parameters that are used in the `studio-env.yml` file for the installation. The parameters that can be validated are as follows:

Table 8: Required File Structure

Parameters	Validation
External Service (OFSAA_SERVICE URL)	The OFSAA_Service URL can be validated by clicking the URL for verification.
DB Details for Studio Schema	You can login to SQL developer and verify the DB Details for Studio Schema.
BD Config Schema	You can login to SQL developer and verify the BD Config Schema.

Studio BE Wallet Details	You can verify the Wallet details by reviewing the steps in Verify the Connectivity of the Wallet .
Atomic Wallet Details	You can verify the Wallet details by reviewing the steps in Verify the Connectivity of the Wallet .
SQL Scripts	You can login to FCC Studio using SQL developer and validate the Studio_DBLINK_BD, if link type is DBLINK, if Schema is not DBLINK, there is no validation required.
Cloudera	You can verify the Cloudera details and validate it by reviewing the steps in Create the Credential Keystore .
Cloudera (Keytab)	Run the command <code>kinit -V <KERBEROS_PRINCIPAL> -k -t <KEYTAB_FILEPATH></code> to verify the keytab.

3.6 Configure the Elastic Search Component

To configure the Elastic Search component, follow these steps:

NOTE

- 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-7.6.2/config` directory.
2. Configure the `elasticsearch.yml` with the following variables:

Table 9: elasticsearch.yml File

Interaction Variable 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 the elastic search service.
http.port	Indicates the port number where the elastic search service 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	<ul style="list-style-type: none"> 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 <code>OutOfMemory</code> error.

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

Table 10: elasticsearch.yml File

Interaction Variable Name	Significance
-Xms1g	<ul style="list-style-type: none"> 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 successful installation of the Elastic Search service.

```
{
  "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"
}

```

3.7 Configure the Interpreter Settings

You must configure the interpreter settings for the following interpreters before installing FCC Studio.

NOTE Ensure to configure the settings only for the interpreters that you require.

Table 11: Pre-installation Interpreter Settings

Interpreter	Prerequisite Settings
fcc-jdbc	No additional configuration is required. NOTE: The FCC Studio application installed without 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 Configure the fcc-ore Interpreter .
fcc-pyspark	<ul style="list-style-type: none"> 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 Configure 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.
pyspark	For the required configuration, see Configure the PySpark Interpreter .
spark	For the required configuration, see Configure the Spark Interpreter .

Topics:

- [Configure the fcc-jdbc Interpreter](#)
- [Configure the fcc-ore Interpreter](#)
- [Configure the fcc-python Interpreter](#)
- [Configure the Spark Interpreter](#)
- [Configure the PySpark Interpreter](#)

3.7.1 Configure the fcc-jdbc Interpreter

To create context for the fcc-jdbc interpreter, follow these steps:

1. Log in to Oracle Database as a **SYSDBA** user.
2. Grant Execute permission to the user using the following command:

```
grant execute dbms_ols to <Studio_DB_Username>;
```

The Execute permission is granted to user.

3. Grant Create permission to the context using the following command:

```
grant create any context to <Studio_DB_Username>;
```

The Create permission is granted to context.

3.7.2 Configuring the fcc-ore Interpreter

To configure the fcc-ore interpreter you must install the [Oracle R Distribution](#).

3.7.2.1 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 [Install Oracle R Distribution on Linux](#).

3.7.3 Configure 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
 - matplotlib-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, you must modify the Python image. As an example, follow the steps provided in the

Modifying the Python Images for the Python Interpreter section in the *OFS Crime and Compliance Studio Administration Guide*.

3.7.3.1 Prerequisites for Installing Python Libraries

Following are the prerequisites to install Python libraries:

- FCC Studio supports any-version of Python with the following required conditions:
 - The preferred Python version is installed on the Processing Server (Studio Notebook Server) and is accessible to the Linux user of FCC Studio.
 - The Python Library (py4j) package is present for the preferred Python version.
 - The prerequisites to install the Python 3.6 libraries that are packaged with the FCC Studio Installer are as follows:
 - The Python3.6 version is installed on the Processing Server (Studio Notebook Server) and is accessible to the Linux user of FCC Studio.
 - The Python Library (py4j) package is installed and is accessible to the Linux user of FCC Studio.

3.7.4 Configure the Spark Interpreter

FCC Studio uses spark 2.4.0. Before you begin the configurations, check the prerequisites depending on your operation mode.

- [Prerequisites](#)
- [Configuration](#)

3.7.4.1 Prerequisites for using the Spark Interpreter

To operate the Spark interpreter in local mode or yarn mode, perform the following:

- [Spark Master Settings](#)
- [Local Mode](#)
- [Yarn Mode](#)

3.7.4.1.1 Spark Master Settings

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.

3.7.4.1.2 Local Mode

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

3.7.4.1.3 Yarn Mode

To operate the Spark interpreter in yarn mode, you must provide custom Spark libraries. To provide your own Spark libraries and (or) Hadoop client-configuration files for the Spark interpreter to connect to a Spark Standalone or Yarn cluster, follow these steps:

- a. Download the Spark libraries from the [Spark's Official Release Archive](#).
- b. Prepare Spark libraries by following these steps:
- c. 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'

- i. To use separate Hadoop libraries, download them as well and place the Hadoop libraries directory in the same directory.
- ii. 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/`
- d. Change the image of the Spark interpreter Init Container in the `<Studio_Installation_Path>/deployments/spark.yml` file to `{{imageRepository}}/fcc-studio/3rdparty:init`.
- e. (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 user interface if you are using a Cloudera cluster.

3.7.4.2 Configuration

The 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 are available 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 from the Interpreters page of the FCC Studio application user interface.

3.7.5 Configure the PySpark Interpreter

FCC Studio uses PySpark 2.4.0. Before you begin the configurations, check the prerequisites depending on your operation mode.

- [Prerequisites](#)
- [Configuration](#)
- [Use the Python Virtual Environments with PySpark](#)

3.7.5.1 Prerequisites

The PySpark interpreter has the same prerequisites as that as the Spark Interpreter. For more information, see [Configure the Spark Interpreter](#). Also, all Spark components must be configured to use the same Python version.

3.7.5.2 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, that 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.

3.7.5.3 Use the Python Virtual Environments with PySpark

To ensure that the two Python versions match, in case your components run on different machines, you must use the Python virtual environments with PySpark.

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

1. [Create a Virtual Environment with Conda](#)
2. [Include Virtual Environment in the Init Container](#)
3. [Update the Interpreter Properties](#)

3.7.5.3.1 Create 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 has to be substituted in further commands.

3. Activate your virtual environment using the following command:

```
conda activate <environment-name>
```
4. Execute the following 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
```

3.7.5.3.2 Include 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. To provide your own Spark libraries and (or) Hadoop client-configuration files for the Spark interpreter to connect to a Spark Standalone or Yarn cluster, follow these steps:

- a. Download the Spark libraries from the [Spark's Official Release Archive](#).
- b. Prepare Spark libraries by following these steps:
 - 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/
```
- iv. Change the image of the Spark interpreter Init Container in the `<Studio_Installation_Path>/deployments/spark.yml` file to `{{imageRepository}}/fcc-studio/3rdparty:init`.
- v. (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 user interface if you are using a Cloudera cluster.

3.7.5.3.3 Update the Interpreter Properties

The interpreter 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`), change the following:
- Change the value of the `spark.yarn.dist.archives` property to `<environment-abs-path>/<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`), change the value of the `zeppelin.pyspark.python` parameter to `<environment-abs-path>/bin/python`.

3.8 Create the Studio Schema

To create studio schema, 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 (Studio schema) is created.

3.9 Assign Grants for the Database Schemas

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_ols 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_ols TO <BD or ECM Schema User>;`
- `GRANT ALL privileges TO <Studio Schema User>;`

4 Setup Password Stores with Oracle Wallet

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.

Topics:

- [Setup the Password Stores for Database User Accounts](#)
- [Verify the Connectivity of the Wallet](#)

4.1.1 Setup the 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 user name 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 create a wallet and the aliases for the database user accounts. For more information on configuring authentication and password stores, see [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 create a wallet, 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 <wallet_location> -create
```

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

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

Figure 1:Wallet Creation

```

-bash-4.1$ pwd
/
-bash-4.1$ mkstore -wrl                               -create
Oracle Secret Store Tool : Version 12.1.0.2
Copyright (c) 2004, 2014, Oracle and/or its affiliates. All rights reserved.

Enter password:
Enter password again:
-bash-4.1$ mkstore -wrl                               -createCredential Studio_808
Oracle Secret Store Tool : Version 12.1.0.2
Copyright (c) 2004, 2014, Oracle and/or its affiliates. All rights reserved.

Your secret/Password is missing in the command line
Enter your secret/Password:
Re-enter your secret/Password:
Enter wallet password:
Create credential oracle.security.client.connect_string1
-bash-4.1$ mkstore -wrl                               -createCredential aif_
Oracle Secret Store Tool : Version 12.1.0.2
Copyright (c) 2004, 2014, Oracle and/or its affiliates. All rights reserved.

Your secret/Password is missing in the command line
Enter your secret/Password:
Re-enter your secret/Password:
Enter wallet password:
Create credential oracle.security.client.connect_string2
-bash-4.1$ mkstore -wrl                               -createCredential aif_
Oracle Secret Store Tool : Version 12.1.0.2
Copyright (c) 2004, 2014, Oracle and/or its affiliates. All rights reserved.

Your secret/Password is missing in the command line
Enter your secret/Password:
Re-enter your secret/Password:
Enter wallet password:

```

3. Create the database connection credentials for the studio schema alias using the following command:

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

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. You are prompted for the wallet password used in Step 1.

4. Create the database connection credentials for the atomic schema alias using the following command:

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

NOTE Creating an atomic schema is not required when you install FCC Studio without OFSAA.

5. Create the database connection credentials for the config schema alias using the following command:

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

NOTE Creating a config schema is not required when you install FCC Studio without OFSAA.

In this manner, create a wallet and associated database connection credentials for all the database user accounts.

The wallet is created in the `configmaps/<wallet_location>` directory 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 [Oracle Database Security Guide](#).


After the wallet is created, go to the `<wallet_location>` directory and click **Refresh**  to view the created wallet folder.

Figure 2: Location of the Created Wallet Folder

| Name | Size | Changed | Rights | Owner |
|-------------|------|---------------------|----------|-------|
| wallet_808_ | | 12-08-2020 14:52:49 | rwx----- | |

The wallet folder contains two files: `ewallet.p12` and `cwallet.sso`.

6. Move the wallet folder to the `<Studio_Installed_Path>/<alias-name>` directory.
7. In the `<wallet_location>` directory, configure the `tnsnames.ora` file to include the entry for each alias name to be set up.

Figure 3: Location of the Created Wallet Folder

```
Studio_808_ =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST = ) (PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = )
    )
  )
)
aif_ =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST = ) (PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = )
    )
  )
)
aif_ =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST = ) (PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = )
    )
  )
)
```

NOTE

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

4.1.2 Verify the Connectivity of the Wallet

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

1. Create a `sqlnet.ora` file 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.

```
$ export WALLET_LOCATION=<wallet_location>
$ export TNS_ADMIN=<tnsnames.ora_location>. If you have created a new
tnsnames.ora file, provide the location of the new file.
$ 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>"
```

4.2 Create the Credential Keystore

Credential keystore must be created for the Behavior Detection (BD) or Enterprise Case Management (ECM) Atomic schema and FCC Studio Schema. To create a credential keystore, follow these steps:

1. Login as 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 credential keystore file path and the alias are correctly mentioned in the `config.sh` file.

4.2.1 Copying and Adding Files

To copy the jar files, follow these steps:

1. Create the folder in the `GRAPH_FILES_PATH` parameter in a node of the big data cluster.
2. Create a folder called `jars`.
3. Navigate to the `<STUDIO_INSTALLED_PATH>/ficdb/etlJars` directory and copy all the jars present in this directory into the big data node inside the folder `jars`.

NOTE

To use the ES-Hadoop connector, download the `commons-httpclient-3.0.1.jar` and `elasticsearch-spark-20_2.11-7.6.2.jar` files and place them in the `jars` folder.

4.3 Download the Installer Kit

To download the software as a `.zip` folder, download the following mandatory one-off patch **31761460** from [My Oracle Support \(MOS\)](#).

5 Installation

After you download the installer kit, you can begin the installation, or deployment of the images and containers in the kubernetes environment. Perform the following steps to complete the installation:

- [Extract the Installer Kit](#)
- [Place Files in the Installation Directories](#)
- [Generate an Encrypted Password](#)
- [Generate the Public and Private Keys](#)
- [Generate the Key Store File for Secure Batch Service](#)
- [Configure the Preferred Services](#)
- [Configure the studio-env.yml File](#)
- [Configure the Extract Transfer and Load ETL Process](#)
- [Verifying the Resource Allocation for FCC Studio Services](#)
- [Deploying FCC Studio on the K8s Cluster](#)

5.1 Extract the Installer Kit

After downloading the **.zip** folder, follow these steps to extract the folder contents:

1. Extract the contents of the installer archive file in the download directory using the following commands:

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

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

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

WARNING 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
```

5.2 Place Files in the Installation Directories

To place the required jars and Kerberos files in the required locations, follow these steps:

1. To place the additional jar files, follow these steps:
 - a. Navigate to the `<Studio_Installation_Path>/batchservice/user/lib` directory.
 - b. Place the following additional jar files:

- `hive-exec-*.jar`. For example, `hive-exec-1.1.0.jar`.
- `HiveJDBC4.jar`
- `hive-metastore-*.jar`. For example, `hive-metastore-1.1.0.jar`.
- `hive-service-*.jar`. For example, `hive-service-1.1.0.jar`.

NOTE

- The version of the jars is client or user-specific. These jars can be obtained from the existing jars of the Cloudera installation.
- The `HiveJDBC4.jar` file is not available in the Cloudera setup. You must download the same from the [Cloudera](#) website.

2. To place the Kerberos files, follow these steps:
 - a. Navigate to the `<Studio_Installation_Path>/batchservice/user/conf` directory.
 - b. Place the following Kerberos files:
 - `krb5.conf`
 - keytab file name as mentioned in the `config.sh` file.

5.3 Generate an Encrypted Password

To generate an encrypted password, follow these steps:

1. Set the `export FIC_DB_HOME` path in the `<Studio_Installed_Directory>/ficdb` directory.
2. Run the `echo $FIC_DB_HOME` command.
3. Go to the `<Studio_Installation_Path>/ficdb/bin` directory and run the `./FCCM_Studio_Base64Encoder.sh <password to be encrypted>` command.

5.4 Generate the Public and Private Keys

The Public and Private keys are JSON Web Tokens (JWT) that are generated for PGX Authentication from FCC Studio.

To generate the keys, follow these steps:

NOTE

The following steps are mandatory for the first time FCC Studio installation.

1. Navigate to the `<Studio_Installation_Path>/ficdb/bin` directory.
2. Run the Shell Script `FCCM_Studio_JWT_Keygen.sh` from the directory.

The Public and Private Keys are generated and available in the `<Studio_Installation_Path>/secrets/keys` directory.

3. Copy the `private.key` and `public.key` files to the `<Studio_Installation_Path>/ficdb/conf` directory.

After generating the key store file and adding the batch service to the PGX trust store, configure the user mapping for GDPR and Redaction changes in the database.

5.5 Apply GDPR and Redaction Changes for FCC Studio

The General Data Protection Regulation (GDPR) is a regulation in EU law on data protection and privacy in the European Union and the European Economic Area. You can apply the GDPR changes that is required for FCC Studio.

To apply GDPR and Redaction, you must configure the following:

- [Generate the Key Store File for Secure Batch Service](#)
- [Add the Batch Service \(SSL\) to PGX Trust Store](#)

5.5.1 Generate the Key Store File for Secure Batch Service

Generating the Key Store file for Secure Batch Service is a process of generating the key store parameters and changing the key store parameters from HTTP to HTTPS protocol.

To configure the Key Store file for Secure Batch Service, follow these steps:

1. Run the `keytool -genkey -alias batchservice -keyalg RSA -keysize 2048 -keystore <Studio_Installation_Path>/OFS_FCCM_STUDIO/batchservice/conf/<Keystore file name>.jks` command in the Studio Server.

When generating the keytool ensure to provide the hostname in first name. For example:

Question: What is your first and last name?

Answer: Provide the batch service name.

2. Specify the keystore password. The `<Keystore file name>.jks` file is created in the path `<Studio_Installation_Path>/OFS_FCCM_STUDIO/batchservice/conf` directory.
3. Specify the following parameters in the `config.sh` file.
 - `export KYESTORE_FILE_NAME=<Keystore file name>.jks`
 - `export KYESTORE_PASS=password`

5.5.2 Add the Batch Service (SSL) to PGX Trust Store

Adding the Batch Service (SSL) to PGX Trust Store facilitates you to apply redaction on the graph batch service and connect with PGX.

To add the Batch Service to PGX Trust Store, follow these steps:

1. Copy the `<Keystore file name>.jks` file to the `<PGX Server path>/server/conf` directory.

2. Navigate to the <PGX Server path>/server/bin directory.
3. Open the start-server file in <PGX Server path>/server/bin directory and add the following lines in `export JAVA_OPTS`:
 - `Djavax.net.ssl.trustStore=<PGX Server path>/conf/<Keystore file name>.jks`
 - `Djavax.net.ssl.trustStorePassword=<Keystore file password>`

The code snippet shows an example of the file when the code is added:

```
#!/bin/bash

export HADOOP_EXTRA_CLASSPATH="$APP_HOME/hdfs-
libs/*:$APP_HOME/conf/hadoop_cluster"

export CLASSPATH="$APP_HOME/shared-lib/common/*:$APP_HOME/shared-
lib/server/*:$APP_HOME/shared-lib/embedded/*:$APP_HOME/shared-
lib/third-party/*:$APP_HOME/conf:$APP_HOME/shared-
memory/server/*:$APP_HOME/shared-memory/common/*:$APP_HOME/shared-
memory/third-party/*:$HADOOP_EXTRA_CLASSPATH"

export JAVA_OPTS="-Dpgx.max_off_heap_size=$PGX_SERVER_OFF_HEAP_MB -
Xmx${PGX_SERVER_ON_HEAP_MB}m -Xms${PGX_SERVER_ON_HEAP_MB}m -
XX:MaxNewSize=${PGX_SERVER_YOUNG_SPACE_MB}m -
XX:NewSize=${PGX_SERVER_YOUNG_SPACE_MB}m -
Dsun.security.krb5.debug=false -
Djavax.security.auth.useSubjectCredsOnly=false -
Djava.security.krb5.conf=$APP_HOME/conf/kerberos/krb5.conf -
Dpgx_conf=$APP_HOME/conf/pgx.conf -
Djavax.net.ssl.trustStore=/scratch/fccstudio/OFS_FCCM_STUDIO/pgx/server
/conf/keystore.jks -Djavax.net.ssl.trustStorePassword=password"

java -cp "$CLASSPATH" -Dfile.encoding=UTF-8 $JAVA_OPTS
oracle.pgx.server.Main $APP_HOME/shared-memory/server/pgx-webapp-*.war
$APP_HOME/conf/server.conf
```

After generating the key store file and adding the batch service to PGX trust store, in the database you must configure the user mapping for the changes made. For more information about how to configure user mapping, see the *FCC Studio Administration Guide*.

5.6 Configure 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. Set the `deployment` parameter depending on the services you want to deploy in the `serviceMapping.sh` file. Set the `deployment` parameter to `All` if you want to deploy all services, or set the `deployment` parameter to `Custom` if you want to choose specific services to deploy.

If the `deployment` parameter is set to `Custom`, then set the values of the desired services to `true` and the values of undesired services to `false`.

A sample `serviceMapping.sh` file is as follows.

Figure 4: Sample serviceMapping.sh File

```
#!/usr/bin/env bash
## This shell-script is used as configuration for starting only desired services
## Set the value of parameter, deployment, as :
## "All" for starting all services
## "Custom" for starting selected services and then set "true"/"false", where "true" means service is desired to be installed.
deployment=All
## For custom:
## FCC Studio independent of OFSAA: set "false" for service(s): authservice, entity-resolution, and matching-service
## FCC Studio lite: set "false" for service(s): fcc-pgql, fcc-pgx-algorithm, fcc-pgx-java and pgx-server.
##Services:
server=true
```

NOTE Do not set `false` for the `server` service.

5.7 Configure 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>/secrets/` directory.
3. Configure the `studio-env.yml` file as shown in the following table.

A sample `studio-env.yml` file is as follows.

Figure 4: Sample studio-env.yml file

```
apiVersion: v1
kind: Secret
metadata:
  name: studio-env
stringData:
  NON_OFSAA: 'false'
  REALM: com.oracle.ofss.fccm.studio.saml.FccSamlRealm
  OFSAA_SERVICE_URL: http://[REDACTED]
  LIVY_HOST_URL: ##LIVY_HOST_URL##
  STUDIO_DB_HOSTNAME: [REDACTED]
  STUDIO_DB_PORT: '1521'
  STUDIO_DB_SERVICE_NAME: [REDACTED]
  STUDIO_DB_SID: [REDACTED]
  STUDIO_DB_USERNAME: [REDACTED]
  STUDIO_DB_PASSWORD: password
  STUDIO_DB_ENCRYPTED_PASSWORD: cGFzc3dvcmQ=
  STUDIO_ALIAS_NAME: [REDACTED]
  STUDIO_WALLET_LOCATION: [REDACTED]
```

NOTE

- Do not alter the parameter values that are already set in the `studio-env.yml` file
- Retain the existing placeholder values for the parameters which are not shown as mandatory in the following table.
- 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.
- When 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 of FCC Studio.
- When 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 12: studio-env.yml Parameters

| Parameter | Significance | Installing with OFSAA (Mandatory) | Upgrading with OFSAA (Mandatory) | Installing without OFSAA (Mandatory) |
|------------|---|-----------------------------------|----------------------------------|--------------------------------------|
| apiVersion | Indicates the current API version.
For example: v1 | Yes | Yes | Yes |
| kind | Indicates the object in which the file information is stored.
For example: Secret | Yes | Yes | Yes |
| metadata | | | | |
| name | Indicates the file name.
For example: studio-env | Yes | Yes | Yes |
| stringData | | | | |
| NON_OFSAA | Indicates the type of installation. <ul style="list-style-type: none">To install FCC Studio with OFSAA on the Kubernetes cluster, enter <i>false</i>.To install FCC Studio without OFSAA on the Kubernetes cluster, enter <i>true</i>. | Enter false | Enter false | Enter true |

| | | | | |
|-------|---|-----|-----|-----|
| REALM | <p>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.</p> <p>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.</p> <p>The FCC Studio application can be accessed using the following realms:</p> <p>FCCMRealm</p> <p>Value=com.oracle.ofss.fccm.studio.datastudio.auth.FCCMRealm</p> <p>IdcsRealm</p> <p>Value=oracle.datastudio.realm.idcs.IdcsRealm</p> <p>DemoRealm</p> <p>Value=com.oracle.ofss.fccm.studio.datastudio.auth.DemoRealm</p> | Yes | Yes | Yes |
|-------|---|-----|-----|-----|

| | | | | |
|-------------------|---|-----|-----|----|
| | <p>NOTE:
The DemoRealm is used only for demo purpose and is not recommended for usage.</p> | | | |
| OFSAA_SERVICE_URL | <p>Indicates the URL of the main database being used for installation.
For example:
<code>https://<HostName>:<PortNo>/<ContextName>/rest-api</code></p> | Yes | Yes | No |
| LIVY_HOST_URL | <p>Indicates the URL of the Livy application.
Example:
<code>http://<HostName>:<PortNo></code>
NOTE:
This parameter is applicable only if fcc-spark-sql, fcc-spark-scala, and (or) fcc-pyspark interpreters are to be used.</p> | 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. This value must not be blank. | Yes | Yes | Yes |

| | | | | |
|---|--|-----|-----|-----|
| STUDIO_DB_ENCRYPTED_PASSWORD | Indicates the encrypted password that is provided for the Studio schema.
For example, cGFzc3dvcmQ. | Yes | Yes | Yes |
| 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 <code>tnsnames.ora</code> file where an entry for the <code>STUDIO_ALIAS_NAME</code> is present. | Yes | Yes | Yes |
| Hadoop Connection Details | | | | |

| | | | | |
|---------------------------------|--|-----|-----|----------|
| STUDIO_HADOOP_CREDENTIAL_ALIAS | Indicated the alias password saved in Hadoop.
For example, studio.password.alias | Yes | Yes | Yes |
| STUDIO_HADOOP_CREDENTIAL_PATH | Indicates the credentials path.
For example, <Studio Installed Path>oracle.password.jceks | 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. This value must not be blank. | Yes | Yes | Enter NA |
| 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 <code>tnsnames.ora</code> file where an entry for the <code>BD_CONFIG_ALIAS_NAME</code> 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. This value must not be blank. | Yes | Yes | Enter NA |

| | | | | |
|--|--|------------|------------|-----------------|
| <p>BD Atomic Wallet Details.
For more information on creating wallet, see Appendix - Setting Up Password Stores with Oracle Wallet .</p> | | | | |
| <p>BD_ATOMIC_ALIAS_NAME</p> | <p>Indicates the BD or ECM atomic alias name.
NOTE:
Enter the alias name that was created during wallet creation.</p> | <p>Yes</p> | <p>Yes</p> | <p>Enter NA</p> |
| <p>BD_ATOMIC_WALLET_LOCATION</p> | <p>Indicates the BD or ECM atomic wallet location.</p> | <p>Yes</p> | <p>Yes</p> | <p>Enter NA</p> |
| <p>BD_ATOMIC_TNS_ADMIN_PATH</p> | <p>Indicates the path of the tnsnames.ora file where an entry for the BD_ATOMIC_ALIAS_NAME is present.</p> | <p>Yes</p> | <p>Yes</p> | <p>Enter NA</p> |
| <p>SQL Scripts</p> | | | | |
| <p>FSINFODOM</p> | <p>Indicates the name of the BD or ECM Infodom.</p> | <p>Yes</p> | <p>Yes</p> | <p>Enter NA</p> |
| <p>FSSEGMENT</p> | <p>Indicates the name of the BD or ECM segment.</p> | <p>Yes</p> | <p>Yes</p> | <p>Enter NA</p> |

| | | | | |
|---|---|------------|------------|-----------------|
| <p>DATAMOVEMENT_LINK_NAME</p> | <ul style="list-style-type: none"> • 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. | <p>Yes</p> | <p>Yes</p> | <p>Enter NA</p> |
| <p>DATAMOVEMENT_LINK_TYPE</p> | <p>If the DB link is used, enter DBLINK in this field. If the DB link is not used, enter SCHEMA in this field.</p> | <p>Yes</p> | <p>Yes</p> | <p>Enter NA</p> |
| <p>Cloudera Setup Details
Contact System Administrator to obtain the required details for these parameters.</p> | | | | |
| <p>HADOOP_CREDENTIAL_PROVIDER_PATH</p> | <p>Indicates the path where Hadoop credential is stored.</p> | <p>Yes</p> | <p>Yes</p> | <p>Enter NA</p> |

| | | | | |
|-----------------------|---|-----|-----|----------|
| HADOOP_PASSWORD_ALIAS | <p>Indicates the Hadoop alias given when creating the Hadoop credentials.</p> <p>NOTE:
Enter the alias name that was created during wallet creation.</p> <p>For information on how to create a credential keystore, see Creating the Credential Keystore</p> | Yes | Yes | Enter NA |
| Hive_Host_Name | Indicates the Hive hostname. | Yes | Yes | Enter NA |
| Hive_Port_number | <p>Indicates the Hive port number.</p> <p>Contact your System Administrator to obtain the port number.</p> | Yes | Yes | Enter NA |
| HIVE_PRINCIPAL | <p>Indicates the Hive Principal.</p> <p>Contact your System Administrator to obtain this value.</p> | 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 |
| 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_krb5_conf_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 | <ol style="list-style-type: none"> 1. Create a file with the name <code>sqoop.properties</code> in the Big Data server and add the following entry to the same: 2. <code>oracle.jdbc.mapDateToTimestamp=false</code> 3. Enter the location of the <code>sqoop.properties</code> file as the value for this parameter. 4. Example: <code>/scratch/ofsa/</code> <p>NOTE:
Ensure that the location name ends with a <code>'/'</code>.</p> | Yes | Yes | Enter NA |
| SQOOP_PARTITION_COL | Indicates the column in which the HIVE table is partitioned.
The value must be <code>SNAPSHOT_DT</code> | Yes | Yes | Enter NA |

| | | | | |
|--------------------|--|-----|-----|----------|
| SQOOP_TRG_HOSTNAME | Indicates the hostname of the Big Data server where SQOOP will run. | Yes | Yes | Enter NA |
| SQOOP_TRG_PASSWORD | Indicates the password of the user of the Big Data server where SQOOP will run. This value must not be blank. | Yes | Yes | Enter NA |
| SQOOP_WORKDIR_HDFS | Indicates the SQOOP working directory in HDFS.
Example: /user/ofsa | Yes | Yes | Enter NA |
| Internal Services | | | | |
| AUTH_SERVICE_URL | Indicates the AUTH service URL that gets activated after the <code>fcc-studio.sh</code> file runs.
Example:
<code>http://<HostName>:7041/authservice</code> | Yes | Yes | No |
| BATCH_SERVICE_URL | Indicates the Batch service URL that gets activated after the <code>fcc-studio.sh</code> file runs.
Example:
<code>http://<HostName>:7043/batchservice</code> | Yes | Yes | Yes |

| | | | | |
|--|---|-----|-----|-----|
| META_SERVICE_URL | Indicates the META service URL that gets activated after the <code>fcc-studio.sh</code> file runs.
Example:
<code>http://<HostName>:7045/met
aservice</code> | Yes | Yes | Yes |
| SESSION_SERVICE_URL | Indicates the Session service URL that gets activated after the <code>fcc-studio.sh</code> file runs.
Example:
<code>http://<HostName>:7047/ses
sionservice</code> | Yes | Yes | Yes |
| PGX_SERVER_URL | Indicates the URL of the PGX server.
Example:
<code>http://<HostName>:<PortNo></code>
The value for PortNo must be 7007. | Yes | Yes | Yes |
| ORE Interpreter Settings
NOTE:
This section is applicable only if ORE interpreter is used. | | | | |
| RSERVE_USERNAME | Indicates the RServe username.
Contact your System Administrator for the username. | No | No | No |
| RSERVE_PASSWORD | Indicates the RServe password.
Contact your System Administrator for the username. | 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:
<code>/usr/lib64/R/library</code> | No | No | No |
| RSERVE_CONF_PATH | Indicates the path where the <code>Rserve.conf</code> file is present.
Default value: <code>/var/ore-
interpreter/rserve</code> | 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 |

| | | | | |
|----------------------------------|---|-----|-----|-----|
| 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. | 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 file path in the HDFS where the <code>graph.json</code> 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 |
| ETL | | | | |
| HDFS_GRAPH_FILES_PATH | Indicates the filepath in the HDFS where the graph.json is formed. | Yes | Yes | No |
| GRAPH_FILES_PATH | Indicates the directory in the Big Data server for graph files. | Yes | Yes | No |
| GRAPH_NAME | Indicates the name you want to assign to the global graph at the end of ETL. | Yes | Yes | No |
| ETL_PROCESSING_RANGE | Indicates the duration for which the data would be moved from Oracle to Hive.

For example: If the ETL_PROCESSING_RANGE = 2Y, 3M, 10D, that is, 2 years, 3 months, and 10 days, and the present date is 20200814, then the data movement occurs for the range 20180504 to 20200814. | Yes | Yes | No |

| | | | | |
|--|---|-----|-----|-----|
| OLD_GRAPH_SESSION_DURATION | Indicates the session older than this specified duration will be removed from the PGX server. If unsure, you can set this value for a week (7D). | Yes | Yes | No |
| REMOVE_TRNXS_EDGE_AFTER_DURATION | Indicates the date range for which transaction edges will be maintained in graph. For example: 6Y, 3M, 10D, which means 6 years, 3 months and 15 days. | Yes | Yes | No |
| CONNECTOR_CHANGESET_SIZE | Indicates the number of nodes or edges you want to process during an update of graph. If unsure, you can set it to 10000. | Yes | Yes | No |
| PGX_SERVER_URLS | Indicates the comma ',' separated values of PGX URLs. If you have only one PGX URL, then the value is <code>http://<k8s master hostname FQDN>:7007</code> . | Yes | Yes | No |
| Quantifind Details
For Quantifind, the generated Quantifind token must be encoded. Use the <code><Fic_DB_path>/FCCM_Studio_Base64Encoder.sh</code> file for encoding Quantifind token. | | | | |
| QUANTIFIND_URL | Indicates the URL of the Quantifind. For example, <code>https://api-test.quantifind.com</code> | Yes | Yes | Yes |

| | | | | |
|----------------------------|--|-----|-----|-----|
| ENCRYPTED_QUANTIFIND_TOKEN | Indicates the token that is generated when integrating with Quantifind.
For example,
c2FtcGxIX2VuY3J5cHRIZF9xdWFudGhmaW5kX3Rva2Vu. | Yes | Yes | Yes |
| QUANTIFIND_APPNAME | Indicates the Quantifind App Name.
For example, OracleIntegrationTest | Yes | Yes | Yes |
| QUANTIFIND_ENABLED | Indicates that Quantifind is enabled.
Options are True or False. | Yes | Yes | Yes |
| HTTPS_PROXY_HOST | Indicates the proxy host that is used.
For example, www-proxy-idc.in.oracle.com | Yes | Yes | Yes |
| HTTPS_PROXY_PORT | Indicates the proxy port that is used.
For example, 80 | Yes | Yes | Yes |
| HTTPS_PROXY_USERNAME | Indicates the proxy username used if there is any.
For example,
##HTTP_PROXY_USERNAME## | Yes | Yes | Yes |
| HTTPS_PROXY_PASSWORD | Indicates the proxy password used if there is any.
For example,
##HTTP_PROXY_PASSWORD## | Yes | Yes | Yes |

| | | | | |
|--|---|-----|-----|-----|
| <p>SAML</p> <p>The SAML related parameters are applicable only if SAMLRealm is used in the Realm parameter.</p> | <ol style="list-style-type: none"> 1. For SAML Realm, the certificate from IDP (key.cert file) is required. 2. The certificate that is obtained from the IDP must be renamed to <code>key.cert</code> and placed in the <code><Studio_Installation_Path>/OFS_FCCM_STUDIO/databastudio/server/conf</code> directory. 3. This certificate is used to identify the trust of the SAML response from the Identity Provider. 4. Specify the Role Attribute name from IDP, in which the User Roles are present in the SAML response. | | | |
| SAML_ISSUER | Indicates the SAML entity ID (Studio URL) configured in the IDP. | Yes | Yes | Yes |
| SAML_DESTINATION | Indicates the SAML IDP URL that is provided by the Identity Provider after creating the SAML Application. | Yes | Yes | Yes |
| SAML_ASSERTION | Indicates the SAML consume URL (Studio/URL/saml/consume) that is configured in IDP. | Yes | Yes | Yes |
| SAML_ROLE_ATTRIBUTE | Indicates the SAML client identifier provided by the SAML Administrator for the Role and Attributes information, while creating the SAML application for FCC Studio. | Yes | Yes | Yes |

| | | | | |
|---|---|-----|-----|-----|
| SAML_LOGOUT_URL | Indicates the SAML client identifier provided by the SAML Administrator for the Logout URL information when creating the SAML application for FCC Studio. | Yes | Yes | Yes |
| SAML_COOKIE_DOMAIN | Indicates the SAML client identifier provided by the SAML Administrator for the Logout URL information when creating the SAML application for FCC Studio. | Yes | Yes | Yes |
| API_USER | Indicates the API users. | 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 |
| 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 | <p>Indicates the source database for FCC Studio.</p> <p>The available options are ECM and BD.</p> <p>NOTE:</p> <ul style="list-style-type: none">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. <p>Here, ECM schema is used as the source of the FCDM data to load the case information into the graph.</p> | Enter BD or ECM | Enter BD or ECM | Enter NA |

| | | | | |
|---|---|---------------------|---------------------|----------|
| Graph Settings | | | | |
| CB_CONFIGURED | Indicates the setting of the graph edges. When the corresponding edges of the graph is required, set the value to <code>true</code> . | Enter true or false | Enter true or false | Enter NA |
| Keystore file and pass details for batch service | | | | |
| KEYSTORE_FILE_NAME | Indicates the keystore file name used for secure batch service.
For example: keystore.jks | Yes | Yes | Yes |
| KEYSTORE_PASS | Indicates the keystore password used for the secure batch service. | Yes | Yes | Yes |
| KEYSTORE_ALIAS | Indicates the keystore alias name used for the secure batch service.
For example: batchservice | Yes | Yes | Yes |

5.8 Configure the Extract Transfer and Load (ETL) Process

Extract Transfer and Load (ETL) is the procedure of copying data from one or more sources into a destination system which represents the data differently from the source or in a different context than the source. Data movement and graph loading is performed using ETL.

NOTE In case you have 8.0.7.4.0 installed and the spark cluster has both `batchservice-8.0.7.*.0.jar` and `elasticsearch-spark-20_2.11-7.*.jar` files installed, you must remove them from the spark class path.

To configure the Data Movement and Graph Load, copy the `FCCM_Studio_SqoopJob.sh`, `FCCM_Studio_ETL_Graph.sh`, `FCCM_Studio_ETL_Connector.sh`, and `FCCM_Studio_ETL_BulkSimilarityEdgeGeneration.sh` files from the `<Studio_Installed_Path>/out/ficdb/bin` directory and add in 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*.

NOTE Before you run the sqoop job, ensure that the `serverconfig.properties` file from the `<Studio_Installed_Path>/batchservice/conf` directory has the correct values.

5.8.1 Configure the Extract Transfer Load (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.jar`

- HiveJDBC4.jar
- hive-metastore-1.1.0.jar
- hive-service-1.1.0.jar

| | |
|-------------|---|
| NOTE | <ul style="list-style-type: none"> • 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 in the applicable files in the configmaps directory as described in the following table:

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

Table 13:Configuring config.sh File

| Parameter | Description |
|---------------------------------|--|
| Deployment Configuration | |
| NAMESPACE | Enter a value to create a namespace with the specified value.
For example: fccs |
| PGX Service | |
| 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 |
| 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: <code>hdfs:///user/fccstudio/graph.json</code> |
| HDFS_GRAPH_FILES_PATH | Indicates the filepath in the HDFS where the <code>graph.json</code> is formed. |
| Quantifind Details | |
| QUANTIFIND_URL | Indicates the URL of the Quantifind.
For example, <code>https://api-test.quantifind.com</code> |
| ENCRYPTED_QUANTIFIND_TOKEN | Indicates the token that is generated when integrating with Quantifind.
For example,
<code>c2FtcGxIX2VuY3J5cHRlZF9xdWFudGlmaW5kX3Rva2Vu.</code> |
| QUANTIFIND_APPNAME | Indicates the Quantifind App Name.
For example, <code>OracleIntegrationTest</code> . |
| QUANTIFIND_ENABLED | Indicates that Quantifind is enabled. Options are True or False. |

- Grant Execute permission to the `<Studio_Installation_Path>/bin` directory using the following command:

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

- 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>"
  }
]
-->
```

5.9 Verify the Resource Allocation for FCC Studio Services

The required resources must be allocated to the FCC Studio services as per the architecture.

Topics:

- [Resource Limits](#)
- [Resource Types](#)
- [Resource Parameters in FCC Studio](#)

5.10 Resource Limits

For FCC Studio to run reliably, the available resources of the Kubernetes cluster must be allocated accordingly. The components are memory-intensive and therefore it is recommended to set memory constraints for each component.

5.11 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.

5.12 Resource Parameters in FCC Studio

After extracting the FCC Studio application installer software, the resource limits must 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 14: 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 |
| 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, see 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, see the PGX 20.0.2 Memory Consumption documentation . |
| fcc-pgx.yml /
pgx-interpretter | k8 | spec.containers[].resources.requests.memory | Memory request size for the PGX interpretter. | 4Gi |
| | k8 | spec.containers[].resources.limits.memory | Memory limit size for the PGX interpretter. | 16Gi

Sizing should depend on the number and behavior (memory requirements of sessions) of concurrent users |

| | | | | |
|--|-----------|--|---|----------------|
| <p>authservice.yml
/ authservice</p> | <p>k8</p> | <p>spec.containers[].resources.requests.memory</p> | <p>Memory request size for the authservice (used for getting roles of a user from DB) component.</p> | <p>default</p> |
| | <p>k8</p> | <p>spec.containers[].resources.limits.memory</p> | <p>Memory limit size for the authservice (used for getting roles of a user from DB) component.</p> | <p>default</p> |
| <p>metaservice.yml
/ metaservice</p> | <p>k8</p> | <p>spec.containers[].resources.requests.memory</p> | <p>Memory request size for the metaservice (used for custom interpreter api's like loaddataset, listdataset in scala interpreter etc.) component.</p> | <p>default</p> |

| | | | | |
|-------------------------------------|----|---|---|---------|
| | k8 | spec.containers[].resources.limits.memory | Memory limit size for the metaservice (used for custom interpreter APIs such as loaddataset, listdataset in Scala interpreter and so on) 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 and so on) component. | default
Depends on volume of data processed in ETL. |
| | k8 | spec.containers[].resources.limits.memory | Memory limit size for the batchservice (used for managing batches like sqoopjob, graph load, notebook execution and so on) component. | default
Depends on volume of data processed in ETL. |
| 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 |
| 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 |

5.13 Deploy 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

Execute the `./fcc-studio.sh -h` command for usage instructions.

Congratulations! Your installation is complete.

After successful completion, the script displays a URL that can be used to access the FCC Studio Application. For more information, see [Access the FCC Studio Application](#).

6 5 Post-installation Steps when OFSAA is Installed

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

Topics:

- [Verify the Installation](#)
- [Checking Logs of FCC Studio Services](#)
- [Configure the SSH Connection](#)
- [Configure Schema Creation](#)
- [Configure the ICIJ Data](#)
- [Restart the PGX Service](#)
- [Access the FCC Studio Application](#)
- [Perform the OFSAA Configuration for Batch Execution](#)
- [Configure and Run Published Notebooks](#)

6.1 Verify the Installation

1. To verify the FCC Studio deployment, wait for a minimum of 10 minutes after completing the execution of the `./fcc-studio.sh` command, and run the `kubectl get pods -n <Namespace>` command.

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.

NOTE

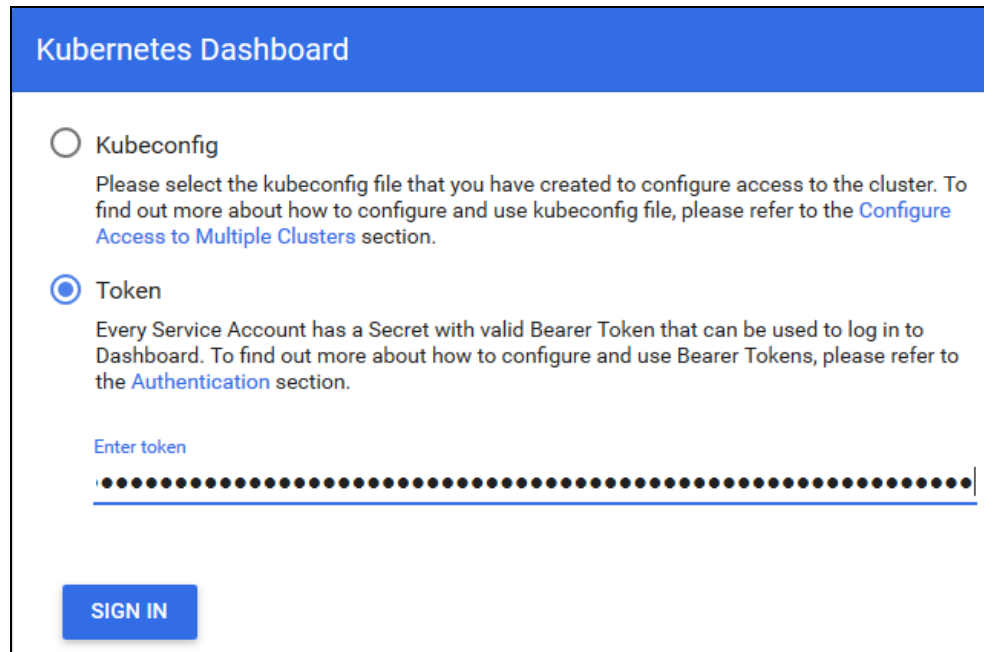
- For FCC Studio deployed with OFSAA, ensure all the pods are ready before accessing the FCC Studio application.
- For FCC Studio deployed with non-OFSAA, ensure the metaservice is up and running before accessing the FCC Studio application.

6.1.1 Check 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:

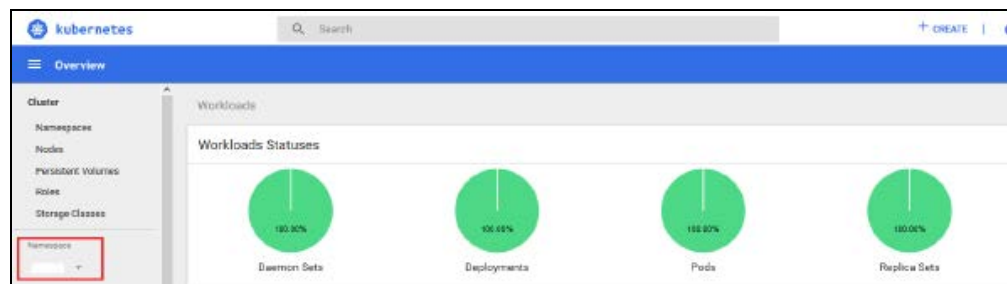
1. Access the Kubernetes Dashboard. The Kubernetes Dashboard login page is displayed.

Figure 5:Kubernetes Dashboard



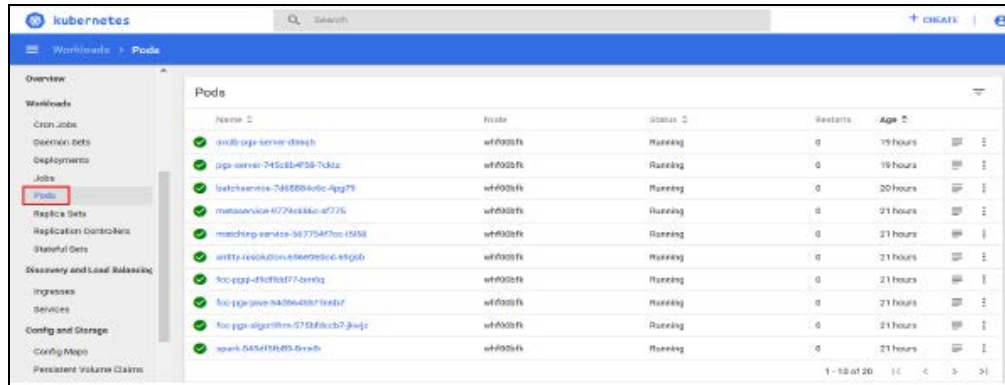
2. Select **Token** and enter the Admin user secret token in the Enter Token field.
3. Click **SIGN IN**. The Kubernetes Dashboard page is displayed.
4. Select <Namespace> from the Namespace drop-down list on the menu items displayed on the LHS.

Figure 6:Kubernetes Dashboard Overview



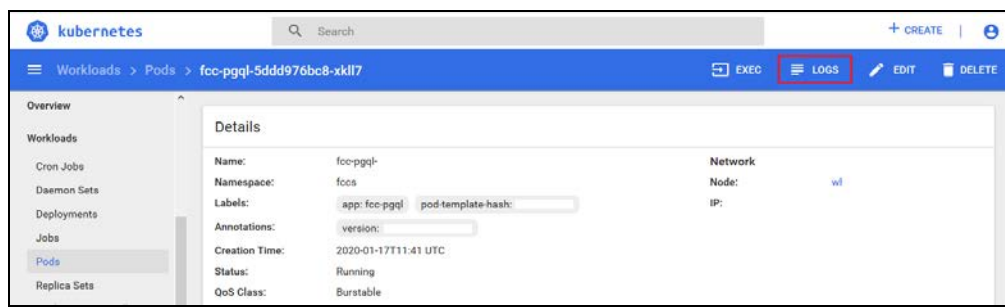
5. Navigate to Workloads > Pods from the menu items displayed on the LHS. The Pods page is displayed with the details of all the services installed as part of the FCC Studio installation.

Figure 7:Kubernetes Dashboard Pods



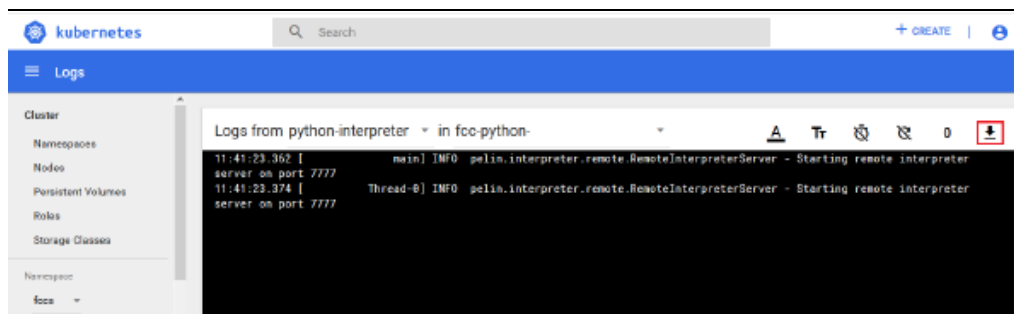
6. Click the service name from the Name column. The service details are displayed.
7. Click the Logs tab.

Figure 8:Kubernetes Dashboard Logs



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

Figure 9:Sample Log



WARNING

If you notice any errors in the log files, do not proceed further. Contact [My Oracle Support \(MOS\)](#) and provide the applicable error code and log files.

If the installation of FCC Studio is unsuccessful, you must reinstall the application after performing the cleanup tasks. For more information, see [Reinstall FCC Studio](#).

6.2 Configure the Schema Creation

Schema creation is a one-time activity that replicates the table structure from the Financial Crime Data Model (FCDM) Atomic schema to the Hive Atomic schema.

Topics:

- [Configure the Schema Creation from the FCC Studio Server](#)
- [Configure the Schema Creation from the OFSAA Server](#)

6.2.1 Configure the Schema Creation from the FCC Studio Server

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

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

NOTE The `$FIC_DB_HOME` path can be set from the `.profile` file.

2. Create a Hive Schema with the name mentioned in the `HIVE_SCHEMA` parameter in the `config.sh` file.

For information on the `config.sh` file, see [Configure the config.sh File](#).

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

```
FCCM_Studio_SchemaCreation.sh HIVE
```

4. Check the `<Studio_Installation_Path>/logs/batchservice.logs` for more information.

6.2.2 Configure the Schema Creation from the OFSAA Server

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

1. Copy all the jar files from the `<Studio_Installation_Path>/ficdb/lib` directory and paste into the `<OFSAA_FIC_HOME_PATH>/ficdb/lib` directory.
2. Copy all the `.sh` files from the `<Studio_Installation_Path>/ficdb/bin` directory and paste into the `<OFSAA_FIC_HOME_PATH>/ficdb/bin` directory.
3. Create a Hive Schema with the name mentioned in the `HIVE_SCHEMA` parameter in the `config.sh` file.

For information on the `config.sh` file, see [Configure the config.sh File](#).

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

```
FCCM_Studio_SchemaCreation.sh HIVE
```

- Check the <Studio_Installaton_Path>/logs/batchservice.logs for more information.

If the atomic schema creation fails, login to the atomic schema of BD/ECM and run the following query:

```
select * from fcc_orahive_datatypemapping;
```

The `fcc_orahive_datatypemapping` table must have only 5 rows. If there are more than 5 rows, run the following query to delete the additional rows:

```
select * from fcc_orahive_datatypemapping for update
```

If the studio schema creation fails, login as a studio user and run the following query:

```
select * from fcc_datastudio_schemaobjects
```

Run the following query to replace all `Y` values with `"`:

```
update fcc_datastudio_schemaobjects set SCHEMA_OBJ_GENERATED=''
```

After the schema creation is successful, the value of the `SCHEMA_OBJ_GENERATED` attribute changes to `Y`.

6.3 Configure the ICIJ Data

To configure the International Consortium of Investigative Journalists (ICIJ) data, you must configure the file path. To configure the file path, follow these steps:

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

- Place the watchlist files in HDFS, that is accessible by the user.
- Update the `FILEPATH` of the watch list files in the `fcc_studio_etl_files` table. The `FILEPATH` refers to the complete hdfs filepath of the csv file. For mapping between `DF_NAME` and `FILEPATH`, see the following image:

Figure 10: 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 |

The following table provides the file path for the csv files:

Table 15: Path of CSV Files

| DF_NAME | File Path |
|------------------------------|---------------------------------------|
| Panama_External_Address | panama_papers.nodes.address.csv |
| Panama_External_Entity | panama_papers.nodes.entity.csv |
| Panama_External_Entity | panama_papers.nodes.intermediary.csv |
| Panama_External_Entity | panama_papers.nodes.officer.csv |
| Panama_edges_address_of | panama_papers.edges.csv |
| Panama_edges_is_related_to | panama_papers.edges.csv |
| Offshore_External_Address | offshore_leaks.nodes.address.csv |
| Offshore_External_Entity | offshore_leaks.nodes.entity.csv |
| Offshore_External_Entity | offshore_leaks.nodes.intermediary.csv |
| Offshore_External_Entity | offshore_leaks.nodes.officer.csv |
| Offshore_edges_address_of | offshore_leaks.edges.csv |
| Offshore_edges_is_related_to | offshore_leaks.edges.csv |
| Bahama_External_Address | bahamas_leaks.nodes.address.csv |
| Bahama_External_Entity | bahamas_leaks.nodes.entity.csv |
| Bahama_External_Entity | bahamas_leaks.nodes.intermediary.csv |
| Bahama_External_Entity | bahamas_leaks.nodes.officer.csv |
| Bahama_edges_address_of | bahamas_leaks.edges.csv |
| Bahama_edges_is_related_to | bahamas_leaks.edges.csv |
| Paradise_External_Address | paradise_papers.nodes.address.csv |

| | |
|------------------------------|--|
| Paradise_External_Entity | paradise_papers.nodes.entity.csv |
| Paradise_External_Entity | paradise_papers.nodes.intermediary.csv |
| Paradise_External_Entity | paradise_papers.nodes.officer.csv |
| Paradise_External_Entity | paradise_papers.nodes.other.csv |
| Paradise_edges_is_related_to | paradise_papers.edges.csv |
| Paradise_edges_address_of | paradise_papers.edges.csv |
| Paradise_edges_is_linked_to | paradise_papers.edges.csv |

6.4 Restart the PGX Service

You must restart the PGX service to ensure that graphs are created. To restart the PGX service, follow these steps:

NOTE Restart the PGX service only after the
`./FCCM_Studio_ETL_Graph.sh sqoop` job is run.

1. Navigate to the `<Studio_Installed_Path>/out` directory.
2. Start a new putty session and run the following commands:


```
kubectl -n fccs delete -f deployments/pgx-server.yml
kubectl -n fccs apply -f deployments/pgx-server.yml
```

6.5 Access the FCC Studio Application

To access FCC Studio, 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 installed on-premise.

The FCC Studio application login page is displayed.

Figure 11:FCC Studio Application Login Page



2. Enter the **Username** and **Password**.
3. Click **Login**.

After you access the application, you can view the ready-to-use notebooks. To check if you have been assigned any roles, create a notebook. If you cannot create a notebook, contact [My Oracle Support \(MOS\)](#).

6.6 Perform the OFSAA Configuration for Batch Execution

NOTE This configuration is not applicable for FCC Studio installed without OFSAA.

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

1. Copy the files in the `<Studio_Installation_Path>/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 from the `<Studio_Installation_Path>/ficdb/lib` directory into the `$FIC_DB_HOME/lib` directory.

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

6.7 Configure and Run Published Notebooks

NOTE This configuration is not applicable for FCC Studio installed without OFSAA.

To perform the configuration required to run published notebooks, copy the required `FCCM_Studio_NotebookExecution.sh` file from the `<Studio_Installation_Path>/out/ficdb/bin` directory into the `<FIC_HOME of OFSAA_Installed_Path>/ficdb/bin` directory.

For information on running published notebooks, see the *Executing Published Notebook* section in the *OFS Crime and Compliance Studio Administration Guide*.

7 Post-installation Steps when OFSAA is Not Installed

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

Topics:

- [Verify the Installation](#)
- [Checking Logs of FCC Studio Services](#)
- [Restart the PGX Service](#)
- [Access the FCC Studio Application](#)

NOTE

Before running the post-installation steps, an SSH connection to the Big Data server must be configured

7.1 Verify the Installation

1. To verify the FCC Studio deployment, wait for a minimum of 10 minutes after completing the execution of the `./fcc-studio.sh` command, and run the `kubectl get pods -n <Namespace>` command.

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.

NOTE

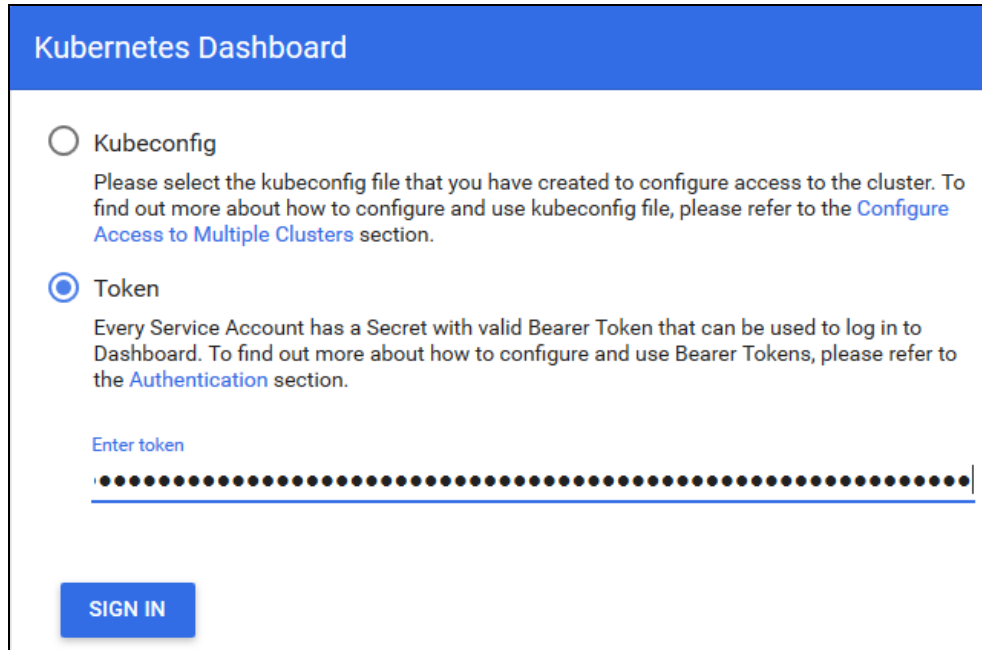
- For FCC Studio deployed with OFSAA, ensure all the pods are ready before accessing the FCC Studio application.
- For FCC Studio deployed with non-OFSAA, ensure the metaservice is up and running before accessing the FCC Studio application.

7.1.1 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:

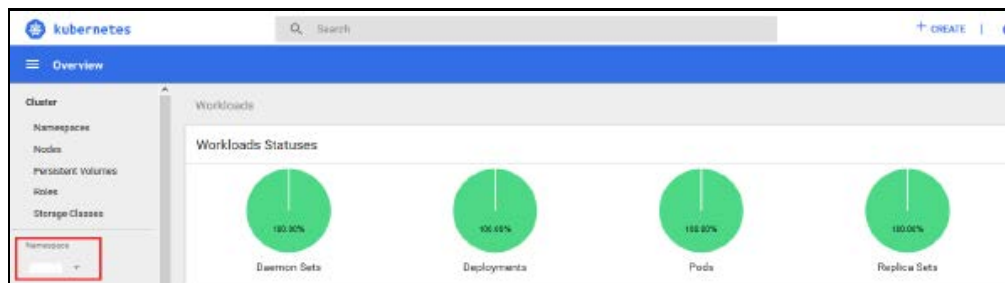
1. Access the Kubernetes Dashboard. The Kubernetes Dashboard login page is displayed.

Figure 12:Kubernetes Dashboard



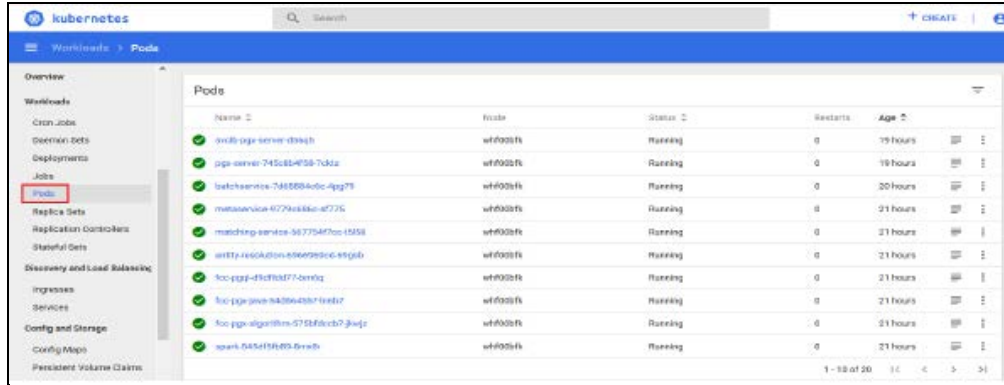
2. Select **Token** and enter the Admin user secret token in the Enter Token field.
3. Click **SIGN IN**. The *Kubernetes Dashboard* page is displayed.
4. Select <Namespace> from the Namespace drop-down list on the menu items displayed on the LHS.

Figure 13:Kubernetes Dashboard Overview



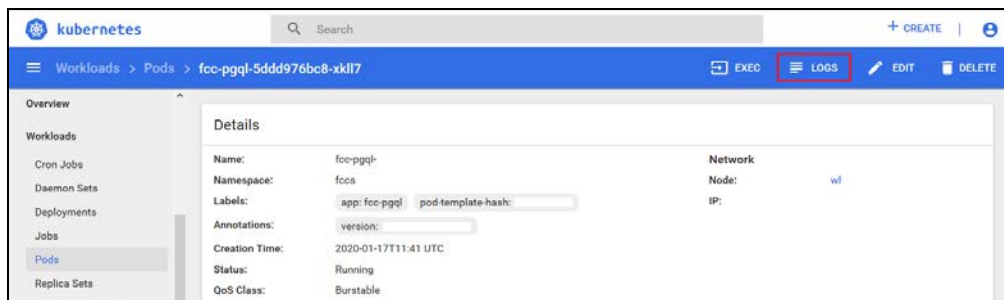
5. Navigate to Workloads > Pods from the menu items displayed on the LHS. The *Pods* page is displayed with the details of all the services installed as part of the FCC Studio installation.

Figure 14:Kubernetes Dashboard Pods



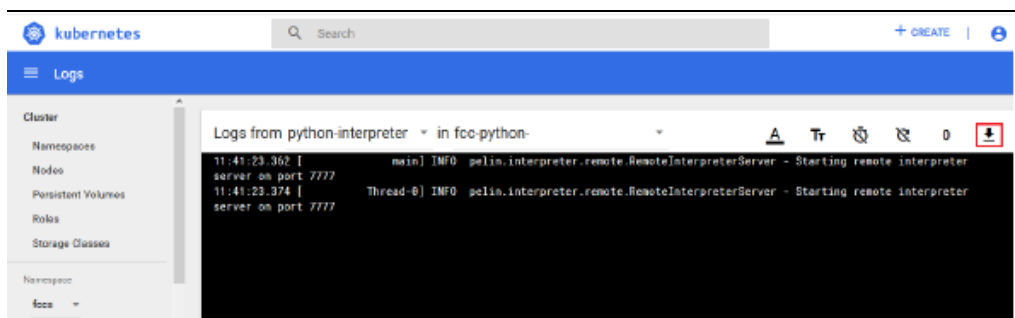
6. Click the service name from the Name column. The service details are displayed.
7. Click the **Logs** tab.

Figure 15:Kubernetes Dashboard Logs



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

Figure 16:Sample Log



WARNING If you notice any errors in the log files, do not proceed further. Contact [My Oracle Support \(MOS\)](#) and provide the applicable error code and log files.

If the installation of FCC Studio is unsuccessful, you must reinstall the application after performing the cleanup tasks. For more information, see [Reinstall FCC Studio](#).

7.2 Restart the PGX Service

You must restart the PGX service to ensure that graphs are created. To restart the PGX service, follow these steps:

NOTE Restart the PGX service only after the `./FCCM_Studio_ETL_Graph.sh sqoop job` is run.

1. Navigate to the `<Studio_Installed_Path>/out` directory.
2. Start a new putty session and run the following commands:

```
kubectl -n fccs delete -f deployments/pgx-server.yml  
kubectl -n fccs apply -f deployments/pgx-server.yml
```

7.3 Access the FCC Studio Application

To access FCC Studio, 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 installed on-premise.

The FCC Studio application login page is displayed.

Figure 17:FCC Studio Application Login Page



2. Enter the Username and Password.
3. Click Login.

After you access the application, you can view the ready-to-use notebooks. To check if you have been assigned any roles, create a notebook. If you cannot create a notebook, contact [My Oracle Support \(MOS\)](#).

8 Upgrade

Follow these steps to upgrade an existing instance of FCC Studio:

1. [Cleanup for Upgrade](#)
2. [Upgrade Steps with OFSAA](#)
3. [Upgrade Steps without OFSAA](#)
4. [Configure the studio-env.yml File for Upgrade](#)

You can upgrade an existing instance of FCC Studio as follows:

- Upgrade FCC Studio from v8.0.7.1 onwards to v8.0.8.0.0 with OFSAA.

NOTE

Here, 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.

- Upgrade FCC Studio from v8.0.7.1 onwards to v8.0.8.0.0 without OFSAA.

NOTE

Here, ensure to provide the same Studio schema and wallet related information that you used during the installation of the existing instance of FCC Studio.

8.1 Cleanup for Upgrade

Before you begin the upgrade process, perform the following cleanup 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
```

8.1.1 Perform Extract Transfer and Load (ETL) Cleanup

To perform the ETL cleanup, follow these steps:

- Extract the contents of the installer archive file in the download directory using the `unzip -a <FCC_Studio_Installer_Archive_File>.zip`. The FCC Studio installer file is extracted in the `<Studio_Installation_Path>` directory.
- Configure the applicable parameters in the `config.sh` file. For more information, see [Configure the config.sh File](#).
- Generate the keystore file. For more information, see [Generate the Key Store File for Secure Batch Service](#).
- Generate an encrypted password. For more information, see [Generate an Encrypted Password](#).
- Remove the existing jar files from the spark cluster.
- Create the folder in the `GRAPH_FILES_PATH` parameter in a node of the big data cluster.
- Create a folder called jars.
- Navigate to the `<STUDIO>/ficdb/etlJars` directory and copy all the jars present in this directory into the big data node inside the folder jars.

NOTE To use the ES-Hadoop connector, [download](#) the `commons-httpclient-3.0.1.jar` and `elasticsearch-spark-20_2.11-7.6.2.jar` files and place them in the jars folder.

- Run `FCCM_Studio_SchemaCreation.sh HIVE`.

8.2 Upgrade Steps with OFSAA

The following table provides the steps to upgrade FCC Studio with OFSAA.

Table 16: Upgrade Steps with OFSAA

| Sl. No. | Activity |
|-------------------------------|--|
| Pre-installation Steps | |
| 1 | Install all the prerequisite hardware and software |
| 2 | Setup the environmental settings |
| 3 | Download the files for Big Data |
| 4 | Configure the Elastic Search Component |
| 5 | Configure the Interpreter Settings |
| 6 | Create the Studio Schema |

| | |
|--------------------------------|--|
| 7 | Assign Grants for the Database Schemas |
| 8 | Setup Password Stores with Oracle Wallet |
| 9 | Create the Credential Keystore |
| 10 | Download the installer kit. |
| Installation Steps | |
| 1 | Extract the installer kit. |
| 2 | Place files in the Installation Directories |
| 3 | Generate an Encrypted Password |
| 4 | Generate the Public Key and Private Key |
| 5 | Generate the Key Store File for Secure Batch Service |
| 6 | Add the Batch Service (SSL) to PGX Trust Store |
| 7 | Configure the Preferred Services |
| 8 | Configure the studio-env.yml File |
| 9 | Configure the Extract Transfer and Load (ETL) Process |
| 10 | Verify the Resource Allocation for FCC Studio Services |
| 11 | Deploy FCC Studio on the K8s Cluster |
| Post-Installation Steps | |
| 1 | Verify the Installation |
| 2 | Configure the Schema Creation |
| 3 | Configure the ICIJ Data |
| 4 | Access the FCC Studio Application |
| 5 | Perform the OFSAA Configuration for Batch Execution |
| 6 | Configure and Run Published Notebooks |

8.2.1 Start 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 [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`

8.3 Upgrade Steps without OFSAA

The following table provides the steps to upgrade FCC Studio without OFSAA.

Table 17: Upgrade Steps without OFSAA

| Sl. No. | Activity |
|-------------------------------|--|
| Pre-installation Steps | |
| 1 | Install all the prerequisite hardware and software |
| 2 | Setup the environmental settings |
| 3 | Download the files for Big Data |
| 4 | Configure the Elastic Search Component |
| 5 | Configure the Interpreter Settings |
| 6 | Create the Studio Schema |
| 7 | Assign Grants for the Database Schemas |
| 8 | Setup Password Stores with Oracle Wallet |
| 9 | Create the Credential Keystore |
| 10 | Download the installer kit. |
| Installation Steps | |
| 1 | Extract the installer kit. |
| 2 | Place files in the Installation Directories |
| 3 | Generate an Encrypted Password |
| 4 | Generate the Public Key and Private Key |
| 5 | Generate the Key Store File for Secure Batch Service |
| 6 | Add the Batch Service (SSL) to PGX Trust Store |
| 7 | Configure the Preferred Services |
| 8 | Configure the studio-env.yml File |
| 9 | Configure the Extract Transfer and Load (ETL) Process |
| 10 | Verify the Resource Allocation for FCC Studio Services |

| | |
|--------------------------------|--|
| 11 | Deploy FCC Studio on the K8s Cluster |
| Post-Installation Steps | |
| 1 | Verify the Installation |
| 2 | Access the FCC Studio Application |

8.4 Configure the studio-env.yml File for Upgrade

Configure the `studio-env.yml` file to upgrade FCC Studio with OFSAA. For more information, see [Configure the studio-env.yml File](#).

9 Reinstall FCC Studio

If the installation of FCC Studio is unsuccessful, you must reinstall the application after performing the required cleanup tasks.

To reinstall 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](#).

9.1.1 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_ols 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](#).

9.1.2 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;
```

9.1.3 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;
```

10 Appendix - Starting and Stopping Services

Following are the steps to start and stop the service and authservice files. Similar steps is applicable for other services.

10.1 Start 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.

10.2 Stop 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
```

11 Appendix A - Tables and Sequences

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

- [Studio Schema Tables](#)
- [Studio Schema Sequences](#)

11.1 Studio Schema Tables

The following table lists the Studio Schema tables that must be dropped during reinstallation of FCC Studio.

| | | |
|----------------------------|---------------------------|-----------------------|
| 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 | | |

11.2 Studio Schema Sequences

The following table lists the Studio Schema sequences that must be dropped during reinstallation of FCC Studio.

| | | |
|----------------------------|------------------------|-------------------------|
| SEQ_COMMENT | SEQ_ENTITY_PERMISSIONS | SEQ_GRAPH |
| SEQ_GROUP | SEQ_INTERPRETER_RESULT | SEQ_INTERPRETER_VARIANT |
| SEQ_JOB | SEQ_NOTEBOOK | SEQ_PARAGRAPH |
| SEQ_PERMISSION | SEQ_PERMISSION_MAPPING | SEQ_RESULT_MESSAGE |
| SEQ_ROLE | SEQ_TASK | SEQ_USER |
| SEQ_VISUALIZATION_TEMPLATE | | |

12 Appendix B - Using Python Virtual Environments with PySpark

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

1. [Create a Virtual Environment with Conda](#)
2. [Include a Virtual Environment in the Init Container](#)
3. [Update the Interpreter Properties](#)

12.1 Create 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
```

12.2 Include a 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 information on how to create a new Init container, see [Yarn Mode](#).

12.3 Update the 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`.

13 Frequently Asked Questions (FAQs)

The FCC Studio installer performs all the pre-requisite validation checks during installation. The questions below help you resolve some issues that you may face during installation.

NOTE This is not an official support document and just attempts to share the knowledge of problem resolution to a few of the known issues. If your issue is not resolved after performing the proposed solution, contact [My Oracle Support \(MOS\)](#).

1. Why does my console show an unsuccessful message during wallet creation?

Please check if you have run the following commands correctly. For more information on wallet creation, see [Setup Password Stores with Oracle Wallet](#).

- a. `mkstore -wrl <wallet_location> -create //creates a wallet in the specified location`
- b. `mkstore -wrl <wallet_location> -createCredential <alias-name> <database-user-name> //creates an alias in the studio schema`
- c. `mkstore -wrl <wallet_location> -createCredential <alias-name> <database-user-name> //creates an alias in the atomic schema`
- d. `mkstore -wrl <wallet_location> -createCredential <alias-name> <database-user-name> //creates an alias in the config schema`

2. Where can I find my created wallet?

Your wallet will be in the directory you have set as your wallet location.

3. Can I run `./install.sh` again if I make updates to the `studio-env.yml` file?

Make updates to the `studio-env.yml` file and then trigger the installation by running the `./install.sh` command. If you have already triggered the installation, then you must do the following. For more information on the attributes you can configure, see [Configure the studio-env.yml File](#).

- Perform a backup of the `studio-env.yml` file.
- Update the applicable attributes in the `studio-env.yml` file.
- run `./install.sh`
- run `./fcc-studio.sh`.

4. When should I create a Database link, and if yes, how do I do it?

Create a Database link to connect the Atomic and Config database schemas to the Studio database schema if the databases are different. You must create the link in the Studio database.

In the following example, a link has been created from the config schema to the atomic schema by running the following script:

```
create public database link <studio database link>
```

```
connect to <Config Schema>
```

```
identified by password
```

```
using ' (DESCRIPTION = ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP) (HOST
=<host name> (PORT = <port number>)) (CONNECT_DATA = (SERVICE_NAME =
<service name>))) ';
```

```
Config schema : <Config Schema>/password
```

```
' (DESCRIPTION = ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP) (HOST
=<host name> (PORT = <port number>)) (CONNECT_DATA = (SERVICE_NAME =
<service name>))) ';
```

After running the script, run the FCDM connector and ICIJ connector jobs.

5. Why does my installed studio setup not have any notebooks?

There are some default notebooks that are ready-to-use when you install FCC Studio. If you do not see any notebooks when you log in to the application, you may not be assigned any roles. Check the <Studio_Installation_Path>/logs directory to see if you have been assigned any roles, and if not, contact your administrator.

6. What can I do if the schema creation fails?

If the Atomic schema creation fails, login to the BD and ECM Atomic schemas and run the following query:

```
select * from fcc_orahive_datatypemapping;
```

The `fcc_orahive_datatypemapping` table must not have duplicate data types.

If the Studio schema creation fails, login as a Studio user and run the following query:

```
select * from fcc_datastudio_schemaobjects
```

Run the following query to replace all Y values with "":

```
update fcc_datastudio_schemaobjects set SCHEMA_OBJ_GENERATED=''
```

After the schema creation is successful, the value of the `SCHEMA_OBJ_GENERATED` attribute changes to **Y**.

You can also check for any errors in the applicable log file in the <Studio_Installation_Path>/logs directory.

7. Why is the sqoop job not successful?

The Sqoop job may fail if some of the applicable values are null, or if the service name or SID value is not provided. Do one of the following:

- Check if there are any null values for the applicable configurations in the `config.sh` and `FCC_DATASTUDIO_CONFIG` tables. If there are any null values, add the required value.
- Check for any errors in the applicable log file in the <Studio_Installation_Path>/logs directory.

8. Why am I getting the following error when I run the sqoop job:


```
Error: Could not find or load main class
com.oracle.ofss.fccm.studio.batchclient
                                         .client.BatchEx
ecute
```

Set the `FIC_DB_HOME` path in the `<Studio_Installation_Path>/ficdb` directory.

You can also check for any errors in the applicable log file in the `<Studio_Installation_Path>/logs` directory.

9. Why is the PGX Server not starting?

The PGX server starts only after the FCDM tables are created, that is, after the FCDM connector job is run. Check if all FCDM tables are created and then start the PGX server. You can also check for any errors in the applicable log file in the `<Installed path>/logs` directory.

10. Why is the ICIJ connector job failing?

This can happen because of a missing `csv` file path in the `FCC_STUDIO_ETL_FILES` table. Add the `CSV` file path. You can also check for any errors in the applicable log file in the `<Installed path>/logs` directory.

OFSAA Support

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 that has all the revised/recently released documents.

