IBM Surveillance Insight for Financial Services
Version 2.0.3

IBM Surveillance Insight for Financial Services Installation Guide

IBM

Note

Before using this information and the product it supports, read the information in “Notices” on page 79.

Product Information

This document applies to Version 2.0.3 and may also apply to subsequent releases.

Copyright

Contents

Introduction

Use IBM® Surveillance Insight® for Financial Services to proactively detect, profile, and prioritize non-compliant behavior in financial organizations. The solution ingests structured and unstructured data, such as trade, electronic communication (emails, chats), and voice data, to flag risky behavior. Surveillance Insights helps you investigate sophisticated misconduct faster, by prioritizing alerts and reducing false positives, and reduce the cost of misconduct.

Some of the key problems that financial firms face in terms of compliance misconduct include:

- Fraudsters using sophisticated techniques making it hard to detect misconduct.
- Monitoring and profiling are hard to do proactively and efficiently with constantly changing regulatory compliance norms.
- A high rate of false positives increases the operational costs of alert management and investigations.
- Siloed solutions make fraud identification difficult and delayed.

IBM Surveillance Insight for Financial Services addresses these problems by:

- Leveraging key innovative technologies, such as behavior analysis and machine learning, to proactively identify abnormalities and potential misconduct without pre-defined rules.
- Using evidence-based reasoning that aids streamlined investigations.
- Using risk-based alerting that reduces false positives and negatives and improves the efficiency of investigations.
- Combining structured and unstructured data from different siloed systems into a single platform to perform analytics.

IBM Surveillance Insight for Financial Services takes a holistic approach to risk detection and reporting. Surveillance Insight combines structured data such as stock market data (trade data) with unstructured data such as emails and voice data, and it uses this data to perform behavior analysis and anomaly detection by using machine learning and natural language processing.

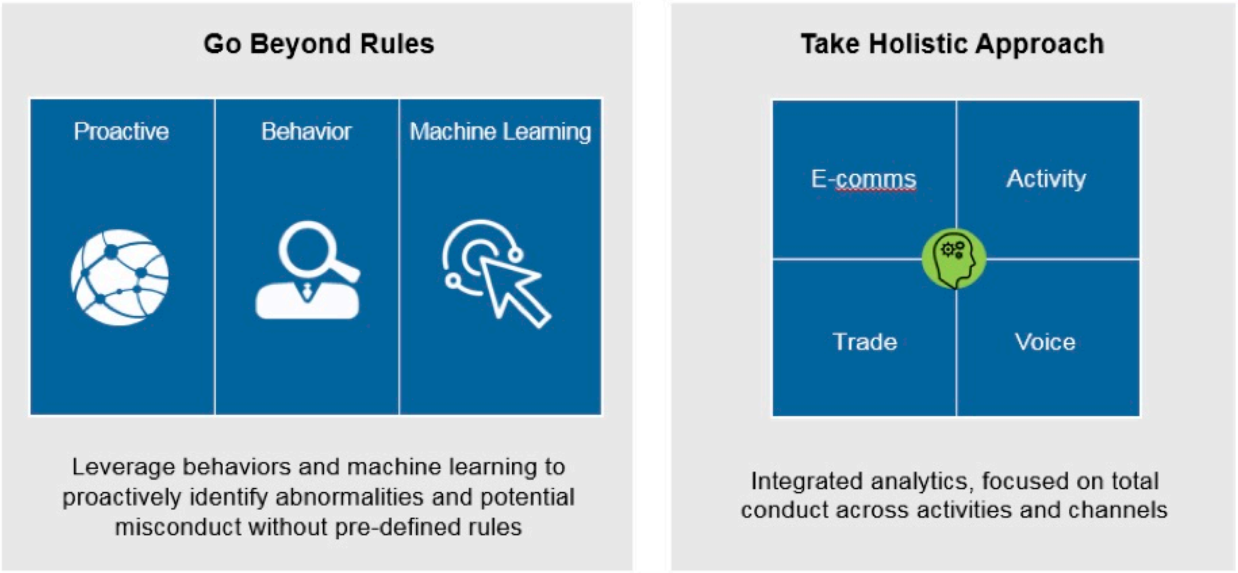
Go Beyond Rules
Proactive
Behavior
Machine Learning
Leverage behaviors and machine learning to proactively identify abnormalities and potential misconduct without pre-defined rules
Take Holistic Approach
E-comms
Activity
Trade
Voice
Integrated analytics, focused on total conduct across activities and channels

Figure 1: Surveillance Insight overview

Audience

This guide is intended for administrators and users of the IBM Surveillance Insight for Financial Services solution. It provides information on installation and configuration of the solution, and information about using the solution.

Finding information and getting help

To find product documentation on the web, access IBM Knowledge Center (www.ibm.com/support/knowledgecenter/SSWTQQ).

Accessibility features

Accessibility features help users who have a physical disability, such as restricted mobility or limited vision, to use information technology products. Some of the components included in the IBM Surveillance Insight for Financial Services have accessibility features. For more information, see Accessibility features.

The HTML documentation has accessibility features. PDF documents are supplemental and as such, include no added accessibility features.

Forward-looking statements

This documentation describes the current functionality of the product. References to items that are not currently available may be included. No implication of any future availability should be inferred. Any such references are not a commitment, promise, or legal obligation to deliver any material, code, or functionality. The development, release, and timing of features or functionality remain at the sole discretion of IBM.

Samples disclaimer

Sample files may contain fictional data manually or machine generated, factual data that is compiled from academic or public sources, or data that is used with permission of the copyright holder, for use as sample data to develop sample applications. Product names that are referenced may be the trademarks of their respective owners. Unauthorized duplication is prohibited.

Chapter 1. IBM Surveillance Insight for Financial Services

IBM Surveillance Insight for Financial Services provides you with the capabilities to meet regulatory obligations by proactively monitoring vast volumes of data for incriminating evidence of rogue trading or other wrong-doing through a cognitive and holistic solution for monitoring all trading-related activities. The solution improves current surveillance process results and delivers greater efficiency and accuracy to bring the power of cognitive analysis to the financial services industry.

The following diagram shows the high-level IBM Surveillance Insight for Financial Services process.

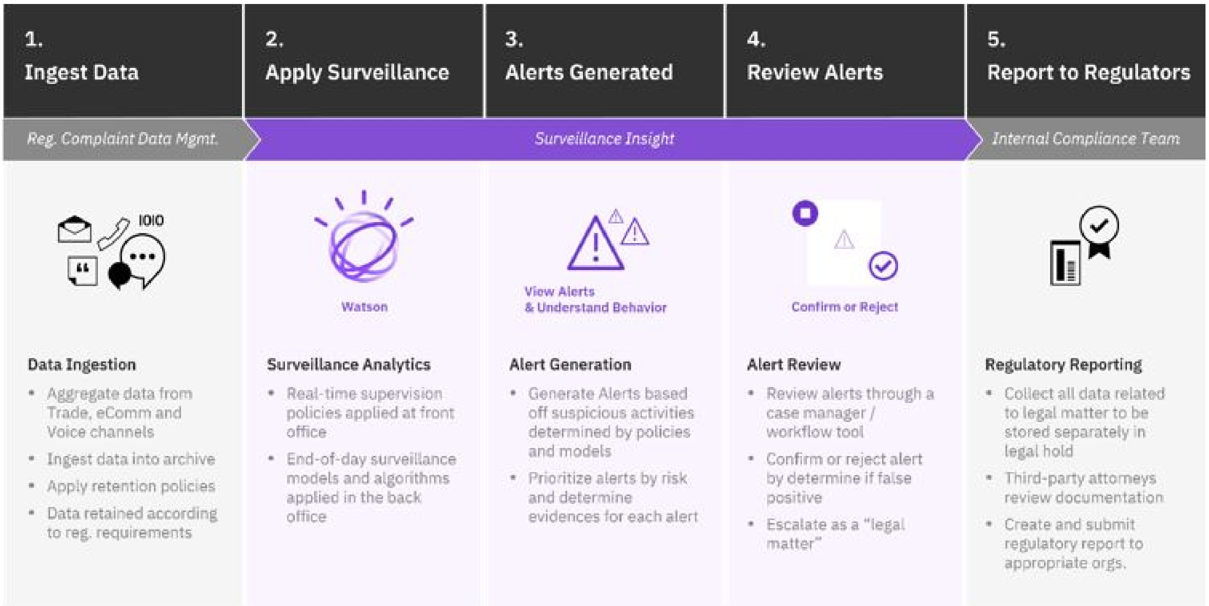
1.
Ingest Data
2.
Apply Surveillance
3.
Alerts Generated
4.
Review Alerts
5.
Report to Regulators
Reg. Complaint Data Mgmt.
Surveillance Insight
Internal Compliance Team
1010
Watson
View Alerts
& Understand Behavior
Confirm or Reject
Data Ingestion
Aggregate data from Trade, eComm and Voice channels
Ingest data into archive
Apply retention policies
Data retained according to reg. requirements
Surveillance Analytics
Real-time supervision policies applied at front office
End-of-day surveillance models and algorithms applied in the back office
Alert Generation
Generate Alerts based off suspicious activities determined by policies and models
Prioritize alerts by risk and determine evidences for each alert
Alert Review
Review alerts through a case manager / workflow tool
Confirm or reject alert by determine if false positive
Escalate as a "legal matter"
Regulatory Reporting
Collect all data related to legal matter to be stored separately in legal hold
Third-party attorneys review documentation
Create and submit regulatory report to appropriate orgs.

Figure 2: High-level process

1. As a first step in the process, data from electronic communications (such as email and chat), voice data, and structured stock market data are ingested into IBM Surveillance Insight for Financial Services.
2. The data is analyzed.
3. The results of the analysis are risk indicators with specific scores.
4. The evidences and their scores are used by the inference engine to generate a consolidated score. This score indicates whether an alert needs to be created for the current set of risk evidences. If needed, an alert is generated and associated with the related parties and stock market tickers.
5. The alerts and the related evidences that are collected as part of the analysis can be viewed in the IBM Surveillance Insight for Financial Services Workbench.

After the alerts are created and the evidences are collected, the remaining steps in the process are completed outside of IBM Surveillance Insight for Financial Services. For example, case investigators must work on the alerts and confirm or reject them, and then investigation reports must be sent out to the regulatory bodies as is required by compliance norms.

The solution architecture

IBM Surveillance Insight for Financial Services is a layered architecture that is made up of several components.

The following diagram shows the different layers that make up the product:

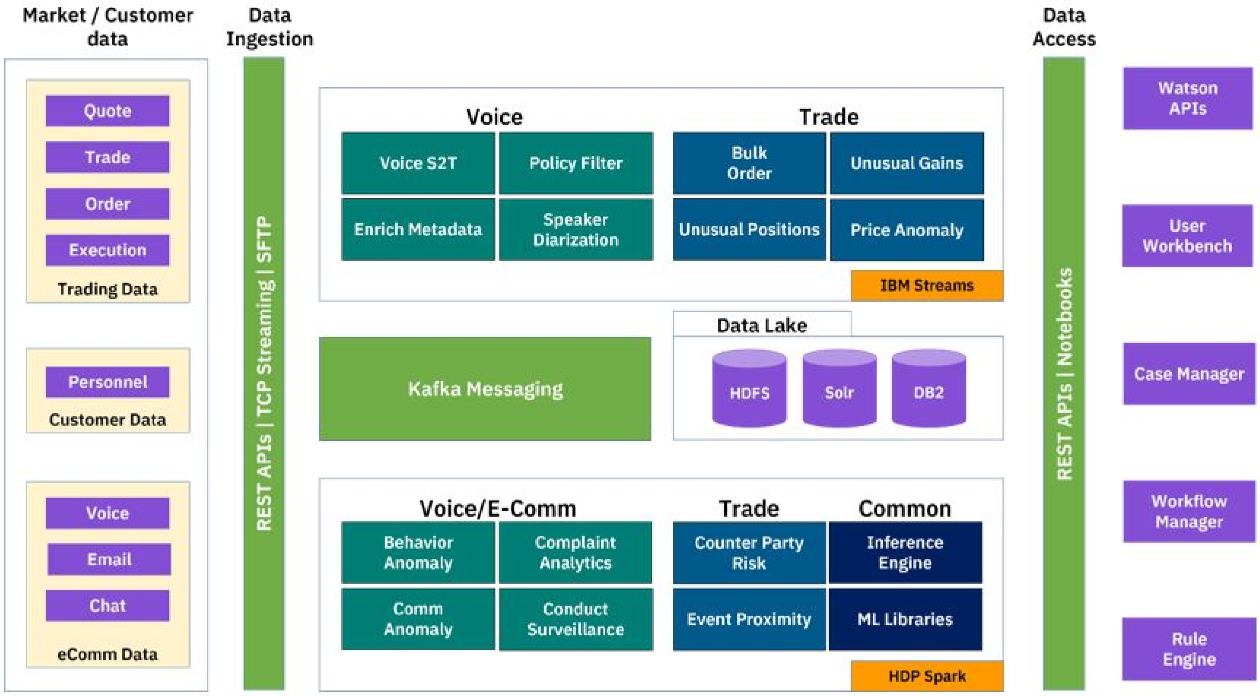

Figure 3: Product layers

- The data layer shows the various types of structured and unstructured data that is consumed by the product.
- The data ingestion layer contains the FTP/TCP-based adaptor that is used to load data into Hadoop. The Kafka messaging system and REST APIs are used for loading e-communications into the system.

  Note: IBM Surveillance Insight for Financial Services does not provide the adaptors with the product.
- The analytics layer contains the following components:
  - Specific use case implementations that leverage the base toolkit operators.
  - Speech 2 Text and Speaker diarization APIs are used in Voice surveillance.
  - Unusual gain, bulk order, and other operators are used for Trade surveillance.
  - NLP APIs are used for e-comm surveillance.
  - The Spark Streaming API is used by Spark jobs as part of the use case implementations.
  - Watson NLC/NLU APIs are used to perform Complaint analytics.
  - A surveillance library that contains the common components that provide core platform capabilities such as alert management, reasoning, and the policy engine.
- Apache Kafka is used as an integration component in the use case implementations and to enable asynchronous communication between the Streams jobs and the Spark jobs.
  - The data layer primarily consists of data in Apache Hadoop, Apache Solr and IBM Db2®.
  - The day-to-day market data is stored in Hadoop. It is accessed by using the spark-sql APIs.
  - Solr is used to index content to enable search capabilities in the workbench.

- Data in Db2 is accessed by using traditional relational SQL. REST services are provided for data that needs to be accessed by the user interfaces and for certain operations, such as alert management.
- Surveillance Insight provides the capability to detect risk from different types of data (trade, e-comm, and voice) and report to a case manager system so that further investigation can be carried out.
- User workbenches like Surveillance Dashboard and Complaints Dashboard are provided to view the results of the analysis.

The deployment architecture

IBM Surveillance Insight for Financial Services is a multi-node architecture where different nodes host different parts of the solution.

The following diagram shows a high-level overview of the deployment architecture.

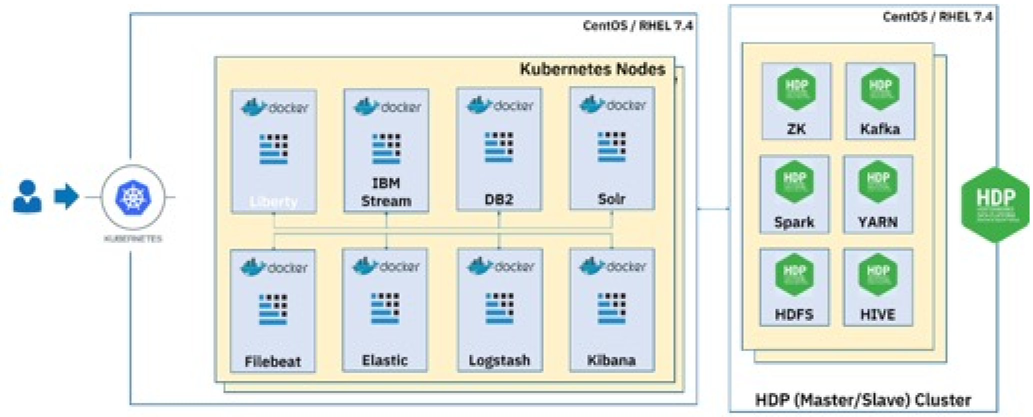

Figure 4: Deployment architecture

IBM Surveillance Insight for Financial Services supports either CentOS v7.4 or RHEL v7.4 operating systems.

IBM Surveillance Insight for Financial Services supports a small (3-node) or medium (6-node) topology for the Hortonworks Data Platform (HDP) installation. HDP provides big data technologies, such as HDFS, Spark, and Kafka.

Apart from HDP, all of the other components are dockerized and are installed and managed by using Kubernetes. The Kubernetes environment can be installed on one node and any configuration up to five nodes.

The security architecture

IBM Surveillance Insight for Financial Services secures data in-motion or at-rest.

The following diagram shows a high-level overview of the security architecture.

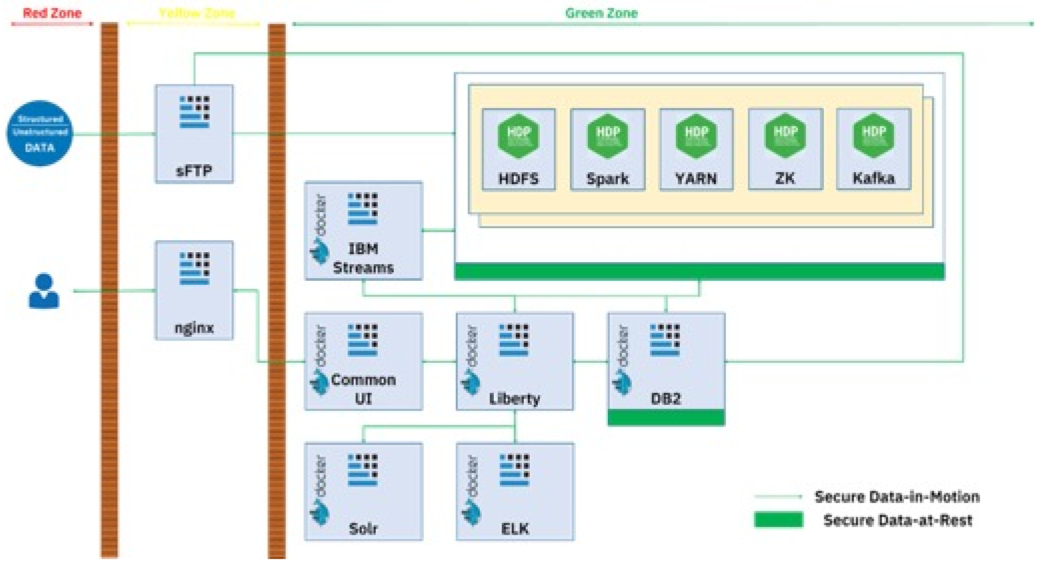

Figure 5: Security architecture

Data is transferred on a TLSv1.2 secured channel within the different components or outside the solution. An AES algorithm is used for encryption when data is stored in Db2 and HDFS

The components

IBM Surveillance Insight for Financial Services includes a base layer of components, and then four components cater to different data channels.

The components are:

- IBM Surveillance Insight for Financial Services (base)
- IBM Electronic Communication Surveillance Analytics
- IBM Trade Surveillance Analytics
- IBM Voice Surveillance Analytics
- IBM Complaints Analytics

The IBM Surveillance Insight for Financial Services (base) component is a prerequisite for any of the other channel-related components. The IBM Electronic Communication Surveillance Analytics component is a prerequisite for IBM Voice Surveillance Analytics or IBM Trade Surveillance Analytics.

Chapter 2. Supported operating systems and hardware requirements

Review the minimum hardware and operating system requirements before you install IBM Surveillance Insight for Financial Services.

For an up-to-date list of environments that are supported by IBM Surveillance Insight for Financial Services, see the IBM Software Product Compatibility Reports (www.ibm.com/support/docview.wss?uid=swg27047153).

The computer on which you run the solution installer and the computer on which you install IBM Surveillance Insight for Financial Services must be running 64-bit CentOS operating systems. The computers on which you install the IBM Open Platform components must be running 64-bit Red Hat Enterprise Linux Server Edition 7.3 operating systems.

Hardware requirements

The computer on which you install IBM Surveillance Insight for Financial Services must have the following hardware requirements:

Processors

2 sockets with 8 cores per socket

RAM

64 GB

Disk space

A flat partition with 500 GB of disk space

A `/var/lib/docker` directory with 100 GB of disk space that uses a B-tree file system (btrfs)

A `/docker-repo` directory with 100 GB of disk space that uses btrfs

Verifying host name configuration

Before installing IBM Surveillance Insight for Financial Services, you must verify the short host name, fully qualified host name, and domain name for your servers.

Perform these steps for each of your servers.

Procedure

1. Update the DNS settings or the `/etc/hosts` file on each server.

   Each server must have an entry in DNS or have an entry in the `/etc/hosts` file to allow for the resolution of both the short host name and long host name. To implement the name resolution by using the `/etc/hosts` file, edit the `/etc/hosts` file on the server to add the IP address of the server and the fully qualified host name. For example, add the following line to the file:

   ```
   nnn.nnn.nnn.nnn sifsserver.example.com sifsserver
   ```

   Where,

   - `nnn.nnn.nnn.nnn` is the IP address of the server
   - `sifsserver.example.com` is the fully qualified domain and host name for the server
   - `sifsserver` is the short name of the server

   Note: If you are using an `/etc/hosts` file for name resolution, ensure that `files` is listed first on the `hosts:` entry in the Name Service Switch configuration file (`/etc/nsswitch.conf` file).

For example,

```
# cat /etc/nsswitch.conf | grep "hosts"
hosts: files dns myhostname
```

2. Log on to the server as `root` user and open a command prompt.
3. Verify and record the defined short host name for the server by typing the following command:

   ```
   hostname -s
   ```

   The command returns the defined short host name for the server, such as `sifsserver`.
4. Verify and record the fully qualified domain and host name for the server by typing the following command:

   ```
   hostname -f
   ```

   The command returns the fully qualified domain and host name for the server, such as `sifsserver.example.com`.
5. Verify and record the domain name for the server by typing the following command:

   ```
   hostname -d
   ```

   The command returns the domain name for the server, such as `example.com`.

Chapter 3. Install the HDP cluster

Installing Apache Ant libraries on all nodes

You must install Apache Ant and Ant Contrib on all of the computers on which you will install a IBM Surveillance Insight for Financial Services component.

Procedure

1. Download Apache Ant 1.9.4 from the Apache Ant website (ant.apache.org/srcdownload.cgi).
2. Save the downloaded file to the /opt directory.
3. Go to the /opt directory, and decompress the file.

   ```
   cd /opt
   ```

   ```
   tar -xvzf apache-ant-1.9.4-bin.tar.gz
   ```

4. Edit the $HOME/.bash_profile file to include the following:

   ```
   export ANT_HOME=/opt/apache-and-1.9.4
   ```

5. Download the ant-contrib-1.0b3.jar file.
   Go to https://sourceforge.net/projects/ant-contrib/files/ant-contrib/1.0b3/, and download ant-contrib-1.0b3-bin.zip.
6. Copy ant-contrib-1.0b3.jar to the $ANT_HOME/lib directory.

Installing Python

You must install Python on each Hadoop node.

Installing Python packages (Online mode)

Use the following steps to install Python in an online mode environment.

Procedure

1. Verify if Python and Pip are installed.

   Use the which python command to identify the installed versions of Python. The command returns the location of the installed instances. If Python 3.5 is not displayed in the results, go to the Python location and verify the version. For example:

   ```
   python -V
   ```

   ```
   /usr/bin/python -V
   ```

   ```
   /usr/local/bin/python3.5 -V
   ```

   If Python 3.5.2 is not displayed, then you must install it.

   Also, the Python packages must be loaded in a specific order to avoid problems with conflicting dependencies. The preliminary packages are downloaded to pre_pythoninstall, and all the rest are downloaded to pythoninstall.

   If Python 3.5 and Pip are installed, you can skip steps 2 - 3.

2. Install Python.

    a) Log in as the root user, and run the following commands:

    ```
    mkdir -p /home/sifsuser/pythoninstall
    ```

    ```
    yum -y install dos2unix
    ```

    ```
    yum -y install yum-utils
    ```

    ```
    yum -y install openssl openssl-devel
    ```

    ```
    yum -y install gcc sqlite-devel bzip2 bzip2-devel gcc-c++
    ```

    b) Download and extract Python 3.5.2.

    ```
    cd /home/sifsuser/pythoninstall
    ```

    ```
    wget https://www.python.org/ftp/python/3.5.2/Python-3.5.2.tgz
    ```

    ```
    tar -xvzf /home/sifsuser/pythoninstall/Python-3.5.2.tgz
    ```

    c) Install Python.

    ```
    cd /home/sifsuser/pythoninstall/Python-3.5.2
    ```

    ```
    ./configure
    ```

    ```
    make
    ```

    ```
    make altinstall
    ```

    ```
    /usr/local/bin/python3.5 -V
    ```

    The results should show Python 3.5.2.

3. Install the Python package installer.

    a) Download and install the Python package installer as the root user.

    ```
    cd /home/sifsuser/pythoninstall
    ```

    ```
    wget https://files.pythonhosted.org/packages/b6/ac/
    7015eb97dc749283ffdec1c3a88ddb8ae03b8fad0f0e611408f196358da3/pip-9.0.1-py2.py3-none-
    any.whl -O /home/sifsuser/pythoninstall/pip-9.0.1-py2.py3-none-any.whl
    ```

    ```
    /usr/local/bin/python3.5 /home/sifsuser/pythoninstall/pip-9.0.1-py2.py3-none-any.whl/pip
    install --no-index --find-links=/home/sifsuser/pythoninstall/ pip-9.0.1-py2.py3-none-
    any.whl
    ```

    The results should show "Successfully installed pip-9.0.1".

    b) Enter the following command to verify the version:

    ```
    pip3.5 --version
    ```

    The result should display a Pip version that is equal to or greater than 9.0.1. You can ignore any update messages.

4. Install the ODBC driver.

Download the database driver as the root user by using the following command:

```
wget https://public.dhe.ibm.com/ibmdl/export/pub/software/data/db2/drivers/odbc_cli/
linuxx64_odbc_cli.tar.gz -P /home/sifsuser/pythoninstall
```

5. Install the Python packages that are used by IBM Surveillance Insight for Financial Services.

   a) As the root user, create a text file that is named `install.sh`.

   b) Add the following text to the file.

   ```
   wget https://github.com/pgmpy/pgmpy/archive/dev.zip -O /home/sifsuser/pythoninstall/
   dev.zip
   wget https://github.com/explosion/spacy-models/releases/download/en_core_web_sm-2.0.0/
   en_core_web_sm-2.0.0.tar.gz -O     /home/sifsuser/pythoninstall/
   en_core_web_sm-2.0.0.tar.gz
   /usr/local/bin/pip3.5 install  numpy==1.12.1
   /usr/local/bin/pip3.5 install  pandas==0.18.1
   /usr/local/bin/pip3.5 install  flask==0.12.2
   /usr/local/bin/pip3.5 install  flask_cors==3.0.3
   /usr/local/bin/pip3.5 install  flask_restful==0.3.6
   /usr/local/bin/pip3.5 install  spacy==2.0.5
   /usr/local/bin/pip3.5 install  nltk==3.2.4
   /usr/local/bin/pip3.5 install  ibm_db==2.0.8a
   /usr/local/bin/pip3.5 install  beautifulsoup4==4.5.1
   /usr/local/bin/pip3.5 install  H5py==2.7.0
   /usr/local/bin/pip3.5 install  gnip_trend_detection==0.5
   /usr/local/bin/pip3.5 install  Hdfs==2.1.0
   /usr/local/bin/pip3.5 install  Requests-kerberos==0.12.0
   /usr/local/bin/pip3.5 install  Scikit-learn==0.18.1
   /usr/local/bin/pip3.5 install  statsmodels==0.8.0
   /usr/local/bin/pip3.5 install  email_reply_parser==0.5.9
   /usr/local/bin/pip3.5 install  textacy==0.4.1
   /usr/local/bin/pip3.5 install  tensorflow==1.8
   /usr/local/bin/pip3.5 install  keras==2.0.2
   /usr/local/bin/pip3.5 install  scipy==1.0.1
   /usr/local/bin/pip3.5 install  wrapt==1.10.11
   /usr/local/bin/pip3.5 install  requests==2.19.1
   /usr/local/bin/pip3.5 install  pyspark==2.3.1
   /usr/local/bin/pip3.5 install  segtok==1.5.6
   ```

   c) Give execution permissions to the file, and then run the script as the root user:

   ```
   chmod 755 install.sh
   ```

   ```
   ./install.sh
   ```

6. Install the pgympy Bayesian network library:

   ```
   cd /home/sifsuser/pythoninstall/
   ```

   ```
   unzip /home/sifsuser/pythoninstall/dev.zip
   ```

   ```
   cd /home/sifsuser/pythoninstall/pgmpy-dev/
   ```

   ```
   /usr/local/bin/python3.5 setup.py install
   ```

7. Install spaCy.

   ```
   cd /home/sifsuser/pythoninstall
   ```

   ```
   /usr/local/bin/pip3.5 install --no-index --find-links=/home/sifsuser/pythoninstall/
   en_core_web_sm-2.0.0.tar.gz
   ```

   For more information about spaCy, see the Models & Language quick start (https://spacy.io/docs/usage/models).

8. Create a symbolic link for this model to en:

   ```
   /usr/local/bin/python3.5 -m spacy link en_core_web_sm en
   ```

Installing Python packages (Offline mode)

Use the following steps to install Python in an offline mode environment.

You must download the packages by using an internet-enabled computer, and then transfer the files to the offline computer. The internet-enabled computer must have Python 3.5.2 and the Python package manager Pip installed.

Procedure

1. On the online computer, verify if Python and Pip are installed.

   Use the `which python` command to identify the installed versions of Python. The command returns the location of the installed instances. If Python 3.5 is not displayed in the results, go to the Python location and verify the version. For example:

   ```
   python -V
   ```

   ```
   /usr/bin/python -V
   ```

   ```
   /usr/local/bin/python3.5 -V
   ```

   If Python 3.5.2 is not displayed, then you must install it.

   Also, the Python packages must be loaded in a specific order to avoid problems with conflicting dependencies. The preliminary packages are downloaded to `pre_pythoninstall`, and all the rest are downloaded to `pythoninstall`.

   If Python 3.5 and Pip are installed, you can skip steps 2 - 3.
2. Install Python on the online computer.
   a) Log in as the root user, and run the following commands:

   ```
   mkdir -p /home/sifsuser/pythoninstall
   ```

   ```
   yum -y install dos2unix
   ```

   ```
   yum -y install yum-utils
   ```

   ```
   yum -y install openssl openssl-devel
   ```

   ```
   yum -y install gcc sqlite-devel bzip2 bzip2-devel gcc-c++
   ```

   b) Download and extract Python 3.5.2.

   ```
   cd /home/sifsuser/pythoninstall
   ```

   ```
   wget https://www.python.org/ftp/python/3.5.2/Python-3.5.2.tgz
   ```

   ```
   tar -xvzf /home/sifsuser/pythoninstall/Python-3.5.2.tgz
   ```

c) Install Python.

```
cd /home/sifsuser/pythoninstall/Python-3.5.2
```

```
./configure
```

```
make
```

```
make altinstall
```

```
/usr/local/bin/python3.5 -V
```

The results should show Python 3.5.2.

3. Install the Python package installer on the online computer.

a) Download and install the Python package installer as the root user.

```
cd /home/sifsuser/pythoninstall
```

```
wget https://files.pythonhosted.org/packages/b6/ac/
7015eb97dc749283ffdec1c3a88ddb8ae03b8fad0f0e611408f196358da3/pip-9.0.1-py2.py3-none-
any.whl -O /home/sifsuser/pythoninstall/pip-9.0.1-py2.py3-none-any.whl
```

```
/usr/local/bin/python3.5 /home/sifsuser/pythoninstall/pip-9.0.1-py2.py3-none-any.whl/pip
install --no-index --find-links=/home/sifsuser/pythoninstall/ pip-9.0.1-py2.py3-none-
any.whl
```

The results should show "Successfully installed pip-9.0.1".

b) Enter the following command to verify the version:

```
pip3.5 --version
```

The result should display a Pip version that is equal to or greater than 9.0.1. You can ignore any update messages.

4. Download the prerequisite packages on the online computer.

Log in as the root user, and run the following commands:

```
cd /home/sifsuser/pre_pythoninstall
```

```
/usr/local/bin/pip3.5 download -d /home/sifsuser/pre_pythoninstall/ numpy==1.12.1
```

```
/usr/local/bin/pip3.5 download -d /home/sifsuser/pre_pythoninstall/ h5py==2.7.0
```

```
/usr/local/bin/pip3.5 download -d /home/sifsuser/pre_pythoninstall/ keras==2.0.2
```

```
/usr/local/bin/pip3.5 download -d /home/sifsuser/pre_pythoninstall/ scipy==1.0.1
```

```
/usr/local/bin/pip3.5 download -d /home/sifsuser/pre_pythoninstall/ pyyaml==3.12
```

```
/usr/local/bin/pip3.5 download -d /home/sifsuser/pre_pythoninstall/ six==1.11.0
```

```
cd /home/sifsuser
```

```
tar -czvf pre_pythoninstall.tar.gz pre_pythoninstall
```

5. Download the main packages on the online computer.

Log in as the root user, and run the following commands:

```
cd /home/sifsuser/pythoninstall
```

```
yumdownloader dos2unix yum-utils openssl openssl-devel gcc sqlite-devel bzip2 bzip2-devel
gcc-c++ --destdir /home/sifsuser/pythoninstall/ --resolve
```

```
wget https://public.dhe.ibm.com/ibmdl/export/pub/software/data/db2/drivers/odbc_cli/
linuxx64_odbc_cli.tar.gz -P /home/sifsuser/pythoninstall
```

```
wget https://github.com/pgmpy/pgmpy/archive/dev.zip -O /home/sifsuser/pythoninstall/dev.zip
```

```
wget https://github.com/explosion/spacy-models/releases/download/en_core_web_sm-2.0.0/
en_core_web_sm-2.0.0.tar.gz -O /home/sifsuser/pythoninstall/en_core_web_sm-2.0.0.tar.gz
```

```
/usr/local/bin/pip3.5 download -d /home/sifsuser/pythoninstall/  beautifulsoup4==4.5.1
```

```
/usr/local/bin/pip3.5 download -d /home/sifsuser/pythoninstall/  email_reply_parser==0.5.9
```

```
/usr/local/bin/pip3.5 download -d /home/sifsuser/pythoninstall/  flask_cors==3.0.3
```

```
/usr/local/bin/pip3.5 download -d /home/sifsuser/pythoninstall/  flask_restful==0.3.6
```

```
/usr/local/bin/pip3.5 download -d /home/sifsuser/pythoninstall/  flask==0.12.2
```

```
/usr/local/bin/pip3.5 download -d /home/sifsuser/pythoninstall/  gnip_trend_detection==0.5
```

```
/usr/local/bin/pip3.5 download -d /home/sifsuser/pythoninstall/ hdfs==2.1.0
```

```
/usr/local/bin/pip3.5 download -d /home/sifsuser/pythoninstall/ ibm_db==2.0.8a
```

```
/usr/local/bin/pip3.5 download -d /home/sifsuser/pythoninstall/ nltk==3.2.4
```

```
/usr/local/bin/pip3.5 download -d /home/sifsuser/pythoninstall/ pandas==0.18.1
```

```
/usr/local/bin/pip3.5 download -d /home/sifsuser/pythoninstall/ pypandoc==1.4
```

```
/usr/local/bin/pip3.5 download -d /home/sifsuser/pythoninstall/ pyspark==2.3.1
```

```
/usr/local/bin/pip3.5 download -d /home/sifsuser/pythoninstall/ requests==2.19.1
```

```
/usr/local/bin/pip3.5 download -d /home/sifsuser/pythoninstall/ requests-kerberos==0.12.0
```

```
/usr/local/bin/pip3.5 download -d /home/sifsuser/pythoninstall/ scikit-learn==0.18.1
```

```
/usr/local/bin/pip3.5 download -d /home/sifsuser/pythoninstall/ segtok==1.5.6
```

```
/usr/local/bin/pip3.5 download -d /home/sifsuser/pythoninstall/ spacy==2.0.5
```

```
/usr/local/bin/pip3.5 download -d /home/sifsuser/pythoninstall/ statsmodels==0.8.0
```

```
/usr/local/bin/pip3.5 download -d /home/sifsuser/pythoninstall/ tensorflow==1.8.0
```

```
/usr/local/bin/pip3.5 download -d /home/sifsuser/pythoninstall/ textacy==0.4.1
```

```
/usr/local/bin/pip3.5 download -d /home/sifsuser/pythoninstall/ wrapt==1.10.11
```

```
cd /home/sifsuser/
```

```
tar -czvf pythoninstall.tar.gz pythoninstall
```

6. Transfer the package files from the online computer to the offline computer.

   ```
   scp pre_pythoninstall.tar.gz root@<offline_host>:/home/sifsuser
   ```

   ```
   scp pythoninstall.tar.gz root@<offline_host>:/home/sifsuser
   ```

7. On the offline computer, decompress the transferred files.

   ```
   cd /home/sifsuser/
   ```

   ```
   tar -xzf /home/sifsuser/pre_pythoninstall.tar.gz
   ```

   ```
   tar -xzf /home/sifsuser/pythoninstall.tar.gz
   ```

8. Install the required RPMs on the offline computer.

   ```
   cd /home/sifsuser/pythoninstall
   ```

   ```
   yum localinstall *.rpm
   ```

   Note: If you see the message "This system is not registered with an entitlement server. You can use subscription-manager to register.", as the root user, set `enabled=0` in the `/etc/yum/pluginconf.d/subscription-manager.conf`, and run the command again.

9. Install Python on the offline computer.

   ```
   cd /home/sifsuser/pythoninstall/Python-3.5.2
   ```

   ```
   ./configure
   ```

   ```
   make
   ```

   ```
   make altinstall
   ```

10. Install the Python package installer on the offline computer.

    ```
    cd /home/sifsuser/pythoninstall
    ```

    ```
    /usr/local/bin/python3.5 /home/sifsuser/pythoninstall/pip-9.0.1-py2.py3-none-any.whl/pip
    install --no-index --find-links=/home/sifsuser/pythoninstall/ pip-9.0.1-py2.py3-none-any.whl
    ```

11. On the offline computer, install the odbc cli database driver.

    ```
    tar -xzf /home/sifsuser/pythoninstall/linuxx64_odbc_cli.tar.gz -C /usr/local/lib/python3.5/
    site-packages/
    ```

12. On the offline computer, export the database home parameter.

    ```
    export IBM_DB_HOME=/usr/local/lib/python3.5/site-packages/clidriver
    ```

13. On the offline computer, as the root user, create a text file that is named `pre_reqs.txt`in the `/home/sifsuser`, and add the following lines to the file:

    ```
    h5py==2.7.0
    keras==2.0.2
    numpy==1.12.1
    pyyaml==3.12
    scipy==1.0.1
    six==1.11.0
    ```

14. Install the prerequisite Python libraries on the offline computer.

```
cd /home/sifsuser
```

```
/usr/local/bin/pip3.5 install --no-deps --no-index --find-links=/home/sifsuser/
pre_pythoninstall -r pre_reqs.txt
```

15. On the offline computer, create a file that is named `requirements.txt` in the `/home/sifsuser` directory, and include the following content in the file:

```
beautifulsoup4==4.5.1
email_reply_parser==0.5.9
flask_cors==3.0.3
flask_restful==0.3.6
flask==0.12.2
gnip_trend_detection==0.5
hdfs==2.1.0
ibm_db==2.0.8a
nltk==3.2.4
pandas==0.18.1
pypandoc==1.4
pyspark==2.3.1
requests==2.19.1
requests-kerberos==0.12.0
scikit-learn==0.18.1
segtok==1.5.6
spacy==2.0.5
statsmodels==0.8.0
tensorflow==1.8.0
textacy==0.4.1
wrapt==1.10.11
```

16. On the offline computer, install the other libraries by running the following commands:

```
/usr/local/bin/pip3.5 install --no-index --find-links=/home/sifsuser/pythoninstall/ pypandoc
```

```
/usr/local/bin/pip3.5 install --no-index --find-links=/home/sifsuser/pythoninstall/ -r
requirements.txt
```

17. On the offline computer, install the pgympy Bayesian network library:

```
cd /home/sifsuser/pythoninstall/
```

```
unzip /home/sifsuser/pythoninstall/dev.zip
```

```
cd /home/sifsuser/pythoninstall/pgmpy-dev/
```

```
/usr/local/bin/python3.5 setup.py install
```

18. On the offline computer, install spaCy.

```
cd /home/sifsuser/pythoninstall
```

```
/usr/local/bin/pip3.5 install --no-index --find-links=/home/sifsuser/pythoninstall/
en_core_web_sm-2.0.0.tar.gz
```

    For more information about spaCy, see the Models & Language quick start (https://spacy.io/docs/usage/models).

19. On the offline computer, create a symbolic link for this model to en:

```
/usr/local/bin/python3.5 -m spacy link en_core_web_sm en
```

Installing Filebeat on the Hadoop nodes

You must install Filebeat on all of the Hadoop node computers.

Procedure

1. Install Filebeat.

   For an offline installation, you must download the RPM on an online computer. Use the following command to download it.

   ```
   curl -L -O https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-6.2.4-x86_64.rpm
   ```

2. Copy the RPM file to each Hadoop node computer.
3. On the Hadoop node computer, go to the directory where you copied the RPM file, and run the following command:

   ```
   sudo rpm -vi filebeat-6.2.4-x86_64.rpm
   ```

4. As the root user, open the `/etc/filebeat/filebeat.yml` file in a text editor.
   a) Update the following paths:

   ```
   # Paths that should be crawled and fetched. Glob based paths.
    paths:
    - /home/sifsuser/logs/sifsspark*.log
   ```

   b) Update with following by uncommenting the section, and modifying the `kubernetes_master_ip` value with the appropriate value.

   ```
   #----------------------------- Logstash output --------------------------
   output.logstash:
     # The Logstash hosts
     hosts: ["<kubernetes-master-ip>:5045"]
   ```

   c) Save and close the file.
5. Start Filebeat by running the following command:

   ```
   sudo service filebeat start
   ```

   Filebeat will start sending the updates by sending the logs from the path that you entered above to the Logstash service that is running on the Kubernetes cluster.
6. Repeat steps 2 - 5 on each Hadoop node computer.

Install Jupyter Notebook

You can install Jupyter Notebook in either an online or offline installation.

Before you can install the Jupyter Notebook, you must have Python and Pip installed.

You must ensure that the IBM Surveillance Insight for Financial Services certificate and key files are available in the `/home/sifsuser/security` directory on each Hortonworks Data Platform (HDP) cluster nodes. The files are:

- `fci_universal_ks.pvtk.pem`
- `fci_universal_ks.crt.pem`

Installing Jupyter Notebook (Online mode)

Use the following steps to install Jupyter in an online environment.

Procedure

1. Log in as the root user.

2. Install the Python virtual environment.

   ```
   cd /home/sifsuser
   ```

   ```
   /usr/local/bin/pip3.5 install virtualenv
   ```

3. Create and activate a virtual environment.

   ```
   virtualenv jup_env
   ```

   ```
   . ./jup_env/bin/activate
   ```

   Note: Ensure that you include the 2 periods in the command.

   The cursor prompt changes to show that all further changes will affect the virtual environment.

   `(jup_env) [root@ ~]#`

   You can exit the virtual environment by entering the following command:

   ```
   deactivate
   ```

4. Create a text file that is named `constraints.txt` and includes the following entries:

   ```
   backcall==0.1.0
   bleach==1.5
   html5lib==0.9999999
   parso==0.3.1
   ```

5. Install Jupyter.

   ```
   cd /home/sifsuser
   ```

   ```
   pip3.5 install jupyter==1.0.0 -c constraints.txt
   ```

   This command uses the virtual environment pip3.5. Do not change to `/usr/local/bin/pip3.5`.

6. Exit the virtual environment.

   ```
   deactivate
   ```

Installing Jupyter Notebook (Offline mode)

Use the following steps to install Jupyter Notebooks in an offline environment.

Procedure

1. Do the following steps using an online server:

   a) As the root user, run the following command to install the Python virtual environment.

   ```
   /usr/local/bin/pip3.5 install virtualenv
   ```

   b) Run the following command to create and activate a virtual environment.

   ```
   virtualenv jup_env
   ```

   ```
   . ./jup_env/bin/activate
   ```

   Note: Ensure that you include the 2 periods in the command.

   The cursor prompt changes to show that all further changes will affect the virtual environment.

   `(jup_env) [root@ ~]#`

You can exit the virtual environment by entering the following command:

```
deactivate
```

c) Create the offline installation directory.

```
mkdir ./jup_env/jupyterinstall
```

d) Create a text file that is named `constraints.txt` and includes the following entries:

```
backcall==0.1.0
bleach==1.5
html5lib==0.9999999
parso==0.3.1
```

e) Download the required packages to the offline folder.

```
pip3.5 download -d ./jup_env/jupyterinstall jupyter==1.0.0 -c constraints.txt
```

Note: This step uses the virtual environment instance of pip3.5. Do not change to `/usr/local/bin/pip3.5`.

f) Compress the offline installation directory.

```
tar -czvf jupyterinstall.tar.gz ./jup_env/jupyterinstall
```

g) Copy the compressed file to the offline server.

```
scp jupyterinstall.tar.gz root@<offline_host>:/home/sifsuser/
```

h) Exit the virtual environment.

```
deactivate
```

2. Do the following steps using the offline server:

a) Log in to the offline server as the root user.

b) Decompress the offline installation folder.

```
cd /home/sifsuser
```

```
tar -xzvf /home/sifsuser/jupyterinstall.tar.gz
```

c) Install Jupyter.

```
/usr/local/bin/pip3.5 install --no-index --find-links=/home/sifsuser/jup_env/
jupyterinstall/ jupyter
```

Note: The command is all that is required to install Jupyter and its dependencies. In addition, there is a `requirements.txt` file in the `/home/sifsuser/jupyterinstall` directory in case an itemized list or a version controlled list of Python packages is required.

Running Jupyter Notebook

After you have installed Jupyter, use the following steps to run it.

Procedure

1. Log in to the Hadoop master node as the sifsuser user.

```
su sifsuser
```

2. Go to the `/home/sifuser` directory. Jupyter Notebooks will not start from an unauthorized directory.

```
cd /home/sifsuser
```

```
mkdir /home/sifsuser/notebooks
```

3. Update the `/home/sifuser/.bashrc` file to include the following:

```
alias jupyter=’/usr/local/bin/jupyter’
alias python3='/usr/local/bin/python3.5'
unset XDG_RUNTIME_DIR
```

4. Save and close the file.
5. Source the `.bashrc` file.

```
source ~/.bashrc
```

6. Create a Jupyter Notebook configuration file.

```
/usr/local/bin/jupyter notebook --generate-config
```

7. Open `/home/sifsuser/.jupyter/jupyter_notebook_config.py` and update the Jupyter Notebook default settings.

```
## The directory to use for notebooks and kernels.
c.NotebookApp.notebook_dir = '/home/sifsuser/notebooks'

# Set options for certfile, ip, password, and toggle off
# browser auto-opening
c.NotebookApp.certfile = '/home/sifsuser/security/fci_universal_ks.crt.pem'

## The full path to a private key file for usage with SSL/TLS.
c.NotebookApp.keyfile = '/home/sifsuser/security/fci_universal_ks.pvtk.pem'

# Set ip to '*' to bind on all interfaces (ips) for the public server
c.NotebookApp.ip = '*'

# set a known, fixed port for server access
c.NotebookApp.port = 8889

## The number of additional ports to try if the specified port is not available.
c.NotebookApp.port_retries = 5

## Shut down the server after N seconds with no kernels or terminals running …
c.NotebookApp.shutdown_no_activity_timeout = 1200

## Supply overrides for the tornado.web.Application that the Jupyter notebook
#  uses.
c.NotebookApp.tornado_settings = {
 "headers": {
     "Content-Security-Policy": "frame-ancestors * 'self'; report-uri /api/security/csp-
report"
    }
  }
```

The notebook will be embedded in the Design Studio as an iframe.

8. To keep the notebook in one tab in Design Studio, create a new custom directory and JavaScript file.

```
cd /home/sifsuser
```

```
mkdir ~/.jupyter/custom
```

```
vi ~/.jupyter/custom/custom.js
```

9. Add the following line to the file:

```
define(['base/js/namespace'], function(Jupyter){Jupyter._target = '_self';});
```

10. Save and close the file.

11. Copy /opt/IBM/IBM_Surveillance_Analytics_2.0.3_Multiplatform_English/analytics.jupyter to /home/sifsuser/analytics.jupyter.

    ```
    cp -r /opt/IBM/IBM_Surveillance_Analytics_2.0.3_Multiplatform_English/analytics.jupyter /
    home/sifsuser/
    ```

12. Run the following command to change the ownership.

    ```
    sudo chown -R sifsuser:sifsuser /home/sifsuser/analytics.jupyter
    ```

13. Optional: If required, update the public IP address that the Jupyter Notebook is available on.

    ```
    /home/sifsuser/analytics.jupyter/config/properties/nlp.properties
    ```

    Update the PUBLIC_IP parameter in the jupyter_ip_service section.
14. As the sifsuser, go to /home/sifsuser/analytics.jupyter.

    ```
    cd /home/sifsuser/analytics.jupyter mkdir logs
    ```

15. Run the following command:

    ```
    python3.5 jupyterREST.py &
    ```

    This command starts the Jupyter service.

Configure security on the HDP nodes

Enabling Kerberos for HDP

You must enable Kerberos for your Hortonworks Data Platform (HDP) environment.

Procedure

1. Install a KDC Server.
   a) On the Ambari server, run the following command:

      ```
      yum install krb5-server krb5-libs krb5-workstation
      ```

   b) Open the KDC server configuration file: /etc/krb5.conf.
      1) In the [libdefaults] section, uncomment default_realm = EXAMPLE.COM.
      2) Change the default_realm value to IBM.COM.
      3) Uncomment the [realms] section, and change the domain to IBM.COM.
      4) Update the kdc and admin_server values with the fully qualified domain name of the Ambari server node.
2. Create the KDC database.
   a) Log in to the Ambari server as the root user.
   b) Use the kdb5_util tool to create the KDC database.

      ```
      kdb5_util create –r IBM.COM –s
      ```

   c) Enter a password when you are prompted.

      Ensure that you record the password that you entered to create the database.

3. Run the following commands to start the KDC and admin servers:

```
systemctl enable krb5kdc
```

```
systemctl enable kadmin
```

4. Add administrators to the Kerberos database.

   a) Run the following command to create a KDC admin user:

   ```
   kadmin.local -q "addprinc admin/admin"
   ```

   b) When prompted, enter a password for the user.

   c) Open the `/var/kerberos/krb5kdc/kadm5.acl` file, and add permissions for the admin user to administer the domain.

   Change `EXAMPLE.COM` to `IBM.COM`.

   ```
   */admin@IBM.COM *
   ```

   d) Restart the KDC and admin servers:

   ```
   systemctl restart krb5kdc
   ```

   ```
   systemctl restart kadmin
   ```

Enabling Kerberos in the Ambari console

Procedure

1. Log in to the Ambari console.

   `https://ambari.server:8081`

2. Click **Admin** > **Kerberos**.
3. Click **Enable Kerberos**.

   You can ignore the Yarn warning.

4. Select **Existing MIT KDC**, select all of the prerequisites, and click **Next**.
5. Enter the information for the KDC and admin account.

   In the **KDC** section, enter the following:

   - In the **KDC Host** field, enter the IP address or the fully qualified domain name for the KDC host.
   - In the **Realm name** field, enter `IBM.COM`.

   In the **Kadmin** section, enter the following:

   - In the **Kadmin Host** field, enter the IP address or the fully qualified domain name for the KDC administrative host.
   - Enter `admin/admin` in the **Admin principle** field, and provide a password.

6. Confirm your configuration, and click **Next**.
7. Click **Install**.

   The installation installs the Kerberos clients on the hosts and tests access to the KDC server.

8. Exit the wizard when the installation is complete.

   After Kerberos is enabled, ensure that all of the services are up and running. If there are errors in any of the services, you can manually restart them by using the Ambari console.

9. Create non-root users and groups that are named sifsuser and solruser on all of the HDP cluster node computers.

   ```
   groupadd sifsuser
   ```

   ```
   useradd sifsuser -g sifsuser
   ```

   ```
   groupadd solruser
   ```

   ```
   useradd solruser -g solruser
   ```

10. Log in to any HDFS node as the hdfs user, and create home directories for the sifsuser and solruser users.

    ```
    su – hdfs
    ```

    ```
    hdfs dfs -mkdir /user/sifsuser
    ```

    ```
    hdfs dfs -chown sifsuser:hadoop /user/sifsuser
    ```

    ```
    hdfs dfs -mkdir /user/solruser
    ```

    ```
    hdfs dfs -chown solruser:hadoop /user/solruser
    ```

    Note: You can find the HDFS node by logging in to the Ambari console and selecting the **HDFS** service. On the **Configs** tab, click **Advanced**. The HDFS host name is shown in **Namenode**, **Datanode**.

11. Log in to the Ambari server as the root user.
12. Add a principal for sifsuser and solruser. In a terminal, run the following commands:

    ```
    kadmin.local
    addprinc -randkey sifsuser@IBM.COM
    ktadd -norandkey -k /etc/security/keytabs/sifsuser.keytab sifsuser@IBM.COM

    addprinc solruser/<hdp_ambari>
    ktadd -k /etc/security/keytabs/solr.keytab solruser/<hdp_ambari>
    quit
    ```

13. Enter a password when prompted.
14. Copy `/etc/security/keytabs/sifsuser.keytab` and `/etc/security/keytabs/solr.keytab` from the Ambari server to each HDP cluster node. Ensure that you copy the files to the same path on each node.
15. Set the ownership and permissions for the keytab files.

    ```
    chown sifsuser:hadoop /etc/security/keytabs/sifsuser.keytab
    ```

    ```
    chmod a+r /etc/security/keytabs/sifsuser.keytab
    ```

    ```
    chown solruser:hadoop /etc/security/keytabs/solr.keytab
    ```

    ```
    chmod a+r /etc/security/keytabs/solr.keytab
    ```

16. To verify Kerberos authentication, log in as sifsuser on any one of the HDFS nodes and run the following command:

    ```
    su - sifsuser
    ```

    ```
    kinit –kt /etc/security/keytabs/sifsuser.keytab sifsuser@IBM.COM
    ```

    The command should complete without errors.

Enabling HDFS and Solr encryption

Procedure

1. Log in to the Ranger KMS console:

   `https://<ranger_server>:6182/index.html`

   The default credentials are `keyadmin/keyadmin`.

   Tip: You can find the Ranger server details by logging into the Ambari console, and selecting **Ranger** > **Configs** > **Advanced**. Use the **Enternal URL** value under **Ranger Setting**.
2. Create an encryption key that is named sifshdfskey.
   a) In the Ranger console, click the **Encryption** tab.
   b) Select the KMS Service.
   c) Click **Add New Key**.
   d) Enter `sifshdfskey` as the key name.
   e) Select the cipher name. Ranger supports AES/CTR/NoPadding as the cipher suite.
   f) Specify the key length as 128 bits.
   g) Add other attributes as needed, and save the key.
3. Create an encryption key that is named solrhdfskey.
   a) In the Ranger console, click the **Encryption** tab.
   b) Select the KMS Service.
   c) Click **Add New Key**.
   d) Enter `solrhdfskey` as the key name.
   e) Select the cipher name. Ranger supports AES/CTR/NoPadding as the cipher suite.
   f) Specify the key length as 128 bits.
   g) Add other attributes as needed, and save the key.
4. Create an encryption zone on one of the HDFS nodes.
   a) Log in to an HDFS node computer as the hdfs user.
   b) Run the following commands:

   ```
   hdfs crypto -createZone -keyName sifshdfskey -path /user/sifsuser
   ```

   ```
   hdfs dfs -chown sifsuser:hadoop /user/sifsuser
   ```

   ```
   hdfs crypto -createZone -keyName solrhdfskey -path /user/solruser
   ```

   ```
   hdfs dfs -chown solruser:hadoop /user/solruser
   ```

   c) To verify the zone, run the following command:

   ```
   hdfs crypto -listZones
   ```

   The new zone and its associated key should be listed.

   ```
   $ hdfs crypto -listZones
   /user/sifsuser  sifshdfskey
   /user/solruser  solrhdfskey
   ```
5. Create access policies for the encryption keys.
   a) Log in to the Ranger console as the keyadmin user.
   b) In **Service Manager**, click **fcicluster_kms**, and click **Add New Policy**.

c) Create a policy that is named sifspolicy, and set the following values:

- In **Key Name**, enter `sifshdfskey`.
- In **Select User**, enter `sifsuser, hdfs`.
- In **Permissions**, click **Add**, and select **Decrypt_EEK**, and click the check mark to add the permission.
- In **Delegate Admin**, select `yes`.

d) Click **Add**.

e) Create a policy that is named solrpolicy, and set the following values:

- In **Key Name**, enter `solrhdfskey`.
- In **Select User**, enter `solruser`.
- In **Permissions**, click **Add**, and select **Decrypt_EEK**, and click the check mark to add the permission.
- In **Delegate Admin**, select `yes`.

f) Click **Add**.

6. Verify that the contents are encrypted.

a) Log in to one of the HDFS nodes as the sifsuser user.

b) Enter the following commands to write a test file that is named `testdata.txt` onto HDFS:

```
echo "Verification of encryption" > testdata.txt
```

```
hdfs dfs -put testdata.txt /user/sifsuser/
```

c) Enter the following command:

```
hdfs dfs -cat /user/sifsuser/testdata.txt
```

This should show clear text data.

d) Log into the HDFS master node as the hdfs user, and run the following command:

```
hdfs dfs -cat /.reserved/raw/user/sifsuser/testdata.txt
```

This should show encrypted data.

Note: If the Kerberos session has expired, run the following command as sifsuser:

```
kinit –kt /etc/security/keytabs/sifsuser.keytab sifsuser@IBM.COM
```

Enabling wire encryption

You enable wire encryption from the Ambari console.

Procedure

1. Log in to the Ambari console.

   `https://ambari.server:8081`

   The default credentials are `admin/admin`.
2. Click **HDFS**, and then click the **Configs** tab.
3. Click **Advanced**.
4. Expand **Custom core-site**, and click **Add Property**.

a) In the **Properties** box, enter the following:

```
hadoop.rpc.protection=privacy
```

   b) Click **Add**.

5. Expand **Custom hdfs-site**, and click **Add Property**.

   a) In the **Properties** box, enter the following:

   ```
   dfs.encrypt.data.transfer=true
   ```

   b) Click **Add**.

6. Click **Save**.
7. In the Ambari console, restart all affected components.

Updating the Kerberos configuration for Kafka

Procedure

1. Log in to the Ambari console.

   `https://ambari.server:8081`

   The default credentials are `admin/admin`.

2. Click **Kafka**, and then click the **Configs** tab.
3. Expand **Advanced kafka-broker**.
4. In **security.inter.broker.protocol**, enter SASL_SSL
5. Expand **Kafka Broker**.
6. Change the **listeners** value to `SASL_SSL://`*`<KafkaBrokerHost>`*`:6667`
7. Click **Save**.
8. In the Ambari console, restart all affected components.

Verifying Kafka

To ensure that your environment is ready to install IBM Surveillance Insight for Financial Services, you can verify your Kafka settings.

Procedure

1. Create a file that is named `client.ssl.properties` in the `/usr/hdp/2.6.4.0-91/kafka/conf` directory.
2. Add the following contents to `client.ssl.properties`:

   ```
   security.protocol=SASL_SSL
   ssl.keystore.location=<ssl.keystore.location>
   ssl.keystore.password= <ssl.keystore.password>
   ssl.truststore.location=<ssl.truststore.location>
   ssl.truststore.password=<ssl.truststore.password>
   ssl.key.password=<ssl.key.password>
   ```

   For the keystore and truststore properties, see the **Kafka** > **Configs** > **Custom kafka-broker** settings in the Ambari console.

3. Create a file that is named `kafka_client_jaas_sifs.conf` in the `/usr/hdp/2.6.4.0-91/kafka/conf` directory.
4. Add the following contents to `kafka_client_jaas_sifs.conf`:

   ```
   KafkaClient {
   com.sun.security.auth.module.Krb5LoginModule required
   useKeyTab=true
   keyTab="/etc/security/keytabs/kafka.service.keytab"
   principal="kafka/<Kafka_Broker_Host_name>"
   useTicketCache=true
   renewTicket=true
   ```

```
serviceName="kafka";
};
```

Change the `<Kafka_Broker_Host_name>` value to the name of the Kafka broker computer.

5. In a terminal window, enter the following command to set the JVM parameters:

```
export KAFKA_OPTS="-Djava.security.auth.login.config=/usr/hdp/2.6.4.0-91/kafka/conf/
kafka_client_jaas_sifs.conf"
```

6. Enter the following command to create a Kafka topic:

```
/usr/hdp/2.6.4.0-91/kafka/bin/kafka-topics.sh --create --zookeeper <ZooKeeper_Host>:2181 --
replication-factor 1 --partitions 1 --topic sifs.ecomm.in
```

Replace *`<ZooKeeper_Host>`* with the appropriate Zookeeper hostname. The hostname is shown in the Ambari console. Click **kafka** > **Configs** > **KafkaBroker** > **zookeeper.connect**.

7. In the current terminal, start the producer by running the following commands:

```
export KAFKA_OPTS="-Djava.security.auth.login.config=/usr/hdp/2.6.4.0-91/kafka/conf/
kafka_client_jaas_sifs.conf"
```

```
cd /usr/hdp/2.6.4.0-91/kafka
```

```
bin/kafka-console-producer.sh --broker-list <KafkaBroker>:6667 --topic sifs.ecomm.in --
producer.config conf/client.ssl.properties --security-protocol SASL_SSL
```

8. In another terminal, start the consumer by running following commands:

```
export KAFKA_OPTS="-Djava.security.auth.login.config=/usr/hdp/2.6.4.0-91/kafka/conf/
kafka_client_jaas_sifs.conf"
```

```
cd /usr/hdp/2.6.4.0-91/kafka
```

```
bin/kafka-console-consumer.sh --bootstrap-server <Kafka_Broker_Host>:6667 --topic
sifs.ecomm.in --new-consumer --consumer.config conf/client.ssl.properties --security-
protocol SASL_SSL
```

9. Go to the `/usr/hdp/2.6.4.0-91/kafka/bin` directory, and run the following commands to run the consumer and producer and validate the message exchange:

```
./kafka-console-producer.sh --broker-list /<Kafka_Broker_Host>:6667 --topic sifs.ecomm.in --
producer.config ../conf/client-ssl.properties --security-protocol SASL_SSL
```

```
./kafka-console-consumer.sh --bootstrap-server /<Kafka_Broker_Host>:6667 --topic
sifs.ecomm.in --new-consumer --consumer.config ../conf/client-ssl.properties --security-
protocol SASL_SSL
```

Messages that are placed on the producer console should be visible in the consumer session.

For example, enter `Hello World!` in the producer terminal. This message should appear in the consumer terminal.

10. On each of the HDFS slave nodes, create a directory that is named `kafka/conf` in `/usr/hdp/2.6.4.0-91`, if it does not exist.

For example,

```
mkdir -p /usr/hdp/2.6.4.0-91/kafka/conf
```

a) Copy `kafka_client_jaas_sifs.conf` to all of the slave nodes in `/usr/hdp/2.6.4.0-91/kafka/conf`.

```
scp /usr/hdp/2.6.4.0-91/kafka/conf/kafka_client_jaas.conf root@<hdp_node>:/usr/hdp/
2.6.4.0-91/kafka/conf
```

b) Copy `kafka.service.keytab` from the Kafka node to the `/etc/security/keytabs` directory on all of the HDFS nodes.

```
scp /etc/security/keytabs/kafka.service.keytab root@<hdp_node>:/etc/security/keytabs
```

c) Run the following command to grant read permissions for all users:

```
chmod a+r /etc/security/keytabs/kafka.service.keytab
```

Configure YARN and Spark

You must add configuration parameters for YARN and Spark by using the Ambari console.

Procedure

1. Log in to the Ambari console.

   `https://ambari.server:8081`
2. Click **Spark2** > **Configs** > **Advanced**.
3. Expand **spark2-log4j-properties**.
4. Add the following **log4j** properties:

   ```
   #SIFS logging
   log4j.logger.com.ibm.sifs=INFO,RollingAppenderSifs
   log4j.logger.com.ibm.si=INFO,RollingAppenderSifs
   #Rolling SIFS Log
   log4j.appender.RollingAppenderSifs=org.apache.log4j.DailyRollingFileAppender
   log4j.appender.RollingAppenderSifs.File=/home/sifsuser/logs/sifsspark.log
   log4j.appender.RollingAppenderSifs.DatePattern='.'yyyy-MM-dd
   log4j.appender.RollingAppenderSifs.layout=org.apache.log4j.PatternLayout
   log4j.appender.RollingAppenderSifs.layout.ConversionPattern=[%p] %d %c %M - %m%n
   ```
5. Click **Save**.

   The Spark job logs will be available in `/home/sifsuser/logs/sifsspark.log`.
6. Click **Spark2** > **Configs**.
7. Expand **Custom spark2-defaults**.
8. Click **Add Property**.

   a) In the **Properties** box, enter the following values:

   ```
   spark.executorEnv.FCI_KAFKA_MSG_KS_ALIASNAME=fci_kafka_msg_key_label
   spark.executorEnv.FCI_KAFKA_MSG_KS_PASSWORD=<fci_kafka_msg.p12 Password>
   spark.executorEnv.FCI_KAFKA_MSG_KS_PATH=/home/sifsuser/security/fci_kafka_msg.p12
   spark.yarn.appMasterEnv.FCI_KAFKA_MSG_KS_ALIASNAME=fci_kafka_msg_key_label
   spark.yarn.appMasterEnv.FCI_KAFKA_MSG_KS_PASSWORD=<fci_kafka_msg.p12 Password>
   spark.yarn.appMasterEnv.FCI_KAFKA_MSG_KS_PATH=/home/sifsuser/security/fci_kafka_msg.p12
   ```

   b) Click **Add**.

   c) Restart all affected components.
9. Click **Ranger KMS** > **Configs**.
10. Click the **Advanced** tab.
11. Expand **Advanced dbks-site**.
12. In **hadoop.kms.blacklist.DECRYPT_EEK**, enter a space.

    A blank value is not allowed by Ambari. However, you can enter a space.
13. Click **Save**.
14. To enable Spark to spawn multiple executors, click **YARN** > **Configs** > **Settings**.
15. Change the **Number of virtual cores** setting to 3.

    The value that you enter in this section might need to be changed based upon the configuration of HDP environment.
16. Click **Save**.

17. Restart all affected components.

Chapter 4. Installing IBM Base Surveillance Insight for Financial Services

You must install the IBM Surveillance Insight for Financial Services base Kubernetes cluster before you can install the other components.

Procedure

1. Log on to the Kubernetes master node computer as the root user.
2. Create a directory for the installation files.

   ```
   mkdir /fcimedia/sifs-base
   ```

3. Save the downloaded IBM Surveillance Insight for Financial Services eImages to the `/fcimedia/sifs-base` directory.

   You must download:

   - `CNT5LEN.tar`
   - `CNT5MEN.tar`
   - `CNT5NEN.tar`
   - `CNT5PEN.tar`
   - `CNT5QEN.tar`
   - `CNT5REN.tar`
   - `CNT5SEN.tar`
   - `CNT5TEN.tar`
   - `CNT5UEN.tar`
   - `CNT5VEN.tar`
   - `CNT5WEN.tar`
   - `CNT5XEN.tar`
   - `CNT5YEN.tar`
   - `CNT5ZEN.tar`
   - `CNT60EN.tar`
   - `CNT61EN.tar`
   - `CNT62EN.tar`
   - `CNT63EN.tar`
   - `CNT64EN.tar`
   - `CNT65EN.tar`
   - `SHA1SUM.tar`
4. Go to the `/fcimedia/sifs-base` directory.
5. Run the following commands to extract the installation scripts from the `surv_insights_1_2.0.3_mp_en.tar` file.

   ```
   tar xf CNT5LEN.tar
   tar xzf sifs-install-
   kit.tar.gz


   rm -rf sifs-install-kit.tar.gz
   ```

6. Run the following command to generate a checksum file for the installation:

```
cd /fcimedia/sifs-base && sha1sum *.tar > ~/SIFS_V2.0.3_Checksum.sha1 && mv ~/
SIFS_V2.0.3_Checksum.sha1 .
```

7. Copy the `CNT2KEN.tar` file from the `/fcimedia/fcco` directory to the `/fcimedia/sifs-base` directory.
8. Go to the `helm` directory where you extracted the installation files:

```
cd /fcimedia/sifs-base/sifs-install-kit/helm/
```

9. Copy the `install.hosts.properties` from the IBM Financial Crimes Insight installation file directory to the `sifs-install-kit/helm` directory.

```
cp $HOME/fci-install-kit/helm/install.hosts.properties .
```

10. Open the `sifs-values.yaml` file in a text editor.
11. Edit the following values:
    - Change the hdp_host value to the IP address for your Hadoop master node.
    - Change the HEALTH_CHECK_URL value to the IP address for your Kubernetes master node.
12. Save and close the file.
13. Open the `install.properties` file in a text editor.
14. Edit the following values:

```
external.docker.registry.url = <your docker registry>:5000
external.nfsserver = <your NFS server>
```

15. Uncomment the following lines:

```
external.docker.registry.isSecure =
false
external.docker.registry.isPreloaded = false
```

16. Add the following line:

```
docker.registry.file = sifs-docker-registry.tar
```

17. In the case of Custom-NFS mounting, do the following. Otherwise, you can skip this step.

    a) Comment out the following lines:

```
configure.nfsserver =
true

nfsserver.master.mount= /fci-
exports
mount_point.1 = --path /sifs-db2-
instance
mount_point.2 = --path /sifs-solr-
instance
mount_point.3 = --path /sifs-liberty-
instance
mount_point.4 = --path /sifs-elasticsearch-
instance
mount_point.5 = --path /sifs-logstash-
instance
mount_point.6 = --path /sifs-filebeat-
instance
mount_point.7 = --path /sifs-liberty-streams-instance
```

    b) Create new file that is named `CustomNFS-si1.yaml` in the `helm` directory, and add the following content:

```
datastorePvNfsPath: /fcisi/sifs-db2-
instance
libertyPvNfsPath: /fcisi/sifs-liberty-
instance
libertystreamsPvNfsPath: /fcisi/sifs-liberty-streams-
instance
```

```
elasticsearchPvNfsPath: /fcisi/sifs-elasticsearch-
instance
logstashPvNfsPath: /fcisi/sifs-logstash-
instance
solrPvNfsPath: /fcisi/sifs-solr-
instance
filebeatPvNfsPath: /fcisi/sifs-filebeat-instance
```

c) Open the `charts.args` file in a text editor, and change the following:

```
chart.args = -f sifs-values.yaml --set global.coreReleaseName=fcco -f CustomNFS-si1.yaml
```

d) Save and close the file.

18. Run the following command:

```
./install.sh
```

Updating the HDP mount point configuration

Procedure

1. Log in to the NFS node as the root user.
2. Update the Liberty Docker.

   a) Open `/fcisi/sifs-liberty-instance/config/kafka.properties` in a text editor, and replace the `bootstrap.servers` value with the Kafka host.

   b) Open `/fcisi/sifs-liberty-instance/krb5.conf` in a text editor, and replace the `admin_server` and kdc values with the Ambari host.

   c) Copy `/etc/security/keytabs/sifsuser.keytab` from the Ambari node to the `/fcisi/sifs-liberty-instance/keytabs` directory on the NFS node.

   d) Copy the `kafka.service.keytab` file from HDP Kafka broker host to the directory that is referenced in the useKeytab parameter in the `/fcisi/sifs-liberty-instance/config/sifs-jass.conf` file.

   e) Update the principal values in the `/fcisi/sifs-liberty-instance/config/sifs-jass.conf` file.

   f) From the Hadoop master node, copy the `/usr/hdp/2.6.4.0-91/hadoop/` directory to the `/fcisi/sifs-liberty-instance/` directory.

   ```
   scp -r /usr/hdp/2.6.4.0-91/hadoop/ root@<nfs server>:/fcisi/sifs-liberty-instance/
   ```

3. Update the Solr Docker.

   a) Copy the `/etc/security/keytabs/solr.keytab` file from the Ambari node to the `/fcisi/sifs-solr-instance/keytabs/` directory on the NFS node.

   b) Create a folder that is named hadoop in the `/fcisi/sifs-solr-instance` directory on the NFS server.

   c) Copy the `conf` directory from the `/usr/hdp/2.6.4.0-91/hadoop` directory (available at the HDFS namenode) to the `/fcisi/sifs-solr-instance/hadoop` directory on the NFS server.

   d) Open `/fcisi/sifs-solr-instance/krb5.conf` in a text editor, and replace the values for the NFS server.

   e) Open `/fcisi/sifs-solr-instance/solrconfig.xm` in a text editor, and replace the `<hdfs.namenode>` values with the your HDFS node.

f) Run the following command on the Solr container:

```
kubectl exec -it <solr> bash
cd /opt/ibm/sifs
./create_solr_cores.sh
```

Note: If you see errors while creating the core, from the Ambari console, click **HDFS** and hover your mouse hover over the **Active NameNode**. Ensure that you use the displayed hostname as the value for `<hdfs.namenode>` in the `/fcisi/sifs-solr-instance/solrconfig.xml` file. Then, rerun `create_solr_cores.sh`.

After the cores are successfully created, you can see the `data` and `index` directories get created in the `/user/solruser` HDFS directory as follows:

```
[root@servername hadoop]# hadoop fs -ls /user/solruser/sifs/data
  Found 3 items
  drwxrwxr-x   - solruser hadoop          0 2018-07-10 04:12 /user/solruser/sifs/data/
index
  drwxrwxr-x   - solruser hadoop          0 2018-07-10 04:02 /user/solruser/sifs/data/
snapshot_metadata
  drwxrwxr-x   - solruser hadoop          0 2018-07-10 04:12 /user/solruser/sifs/data/tlog
```

4. Establish a mount point for the Kibana pod.

a) Run the following command:

```
kubectl edit deployment sifs-base-kibana
```

b) Modify the paths in env and `volumeMounts`:

`/usr/share/kibana/config/kibana.crt` to `/usr/share/certificates/kibana.crt`

`/usr/share/kibana/config/kibana.key` to `/usr/share/certificates/kibana.key`

Integrating Surveillance Insight with the Financial Crimes Insight interface

You can integrate the Surveillance Workbench, Design Studio, and Complaints Workbench applications with the IBM Financial Crimes Insight Investigative UI.

Procedure

1. Log in to the NFS server as the root user.
2. Open the `/fcicore/fci-cui-web/nginx.conf` file in text editor.
3. Uncomment the following section, and save the file:

```
upstream fci-si {
server sifs-base-liberty.default.svc.cluster.local:9443;
}
location /surveillance {
    proxy_pass https://fci-si;
    client_max_body_size 0;
 }
 location /SIFSServices {
    proxy_pass https://fci-si;
    proxy_read_timeout 10d;
    client_max_body_size 0;
 }
 location /ui.complaintsdashboard {
    proxy_pass https://fci-si;
    client_max_body_size 0;
 }
 location /complaintsservices {
    proxy_pass https://fci-si;
    proxy_read_timeout 10d;
    client_max_body_size 0;
 }
 location /surveillancetool {
    proxy_pass https://fci-si;
    client_max_body_size 0;
 }
 location /SIFSModelServices {
    proxy_pass https://fci-si;
    proxy_read_timeout 10d;
    client_max_body_size 0;
 }
 location /CommServices {
    proxy_pass https://fci-si;
    proxy_read_timeout 10d;
    client_max_body_size 0;
 }
 location /SIFSVoiceDataService {
    proxy_pass https://fci-si;
    proxy_read_timeout 10d;
    client_max_body_size 0;
 }
 location /SIFSVoiceIngestionService {
    proxy_pass https://fci-si;
    proxy_read_timeout 10d;
    client_max_body_size 0;
 }
 location /righttoforget {
    proxy_pass https://fci-si;
    client_max_body_size 0;
 }
```

4. Log in to the Kubernetes master node as the root user.
5. Open the `fcco-common-ui` config map by using following command:

   ```
   kubectl edit cm fcco-common-ui
   ```

6. Add the following values to the existing keys and save the file:

   ```
   APP_ROLE_ANALYST: case-manager,surveillance/search/index.html
   APP_ROLE_DATA_SCIENTIST: case-manager,surveillancetool/
   APP_ROLE_INVESTIGATOR: case-manager,surveillance/search/index.html
   APP_ROLE_SUPERVISOR: case-manager,surveillance/dashboard/index.html,ui.complaintsdashboard/
   ```

   Note: Do not add spaces between multiple values.
7. On Kubernetes master node, run the following commands to delete the existing Common-UI pods. After the pods are deleted, Kubernetes will automatically start fresh pods with your changes.

   ```
   kubectl delete pod $(kubectl get pods -l release=fcco,app=common-ui-nodejs -o
   jsonpath='{.items[*].metadata.name}')
   ```

   ```
   kubectl delete pod $(kubectl get pods -l release=fcco,app=common-ui-nginx -o
   jsonpath='{.items[*].metadata.name}')
   ```

8. After the pods are up and running, log in to the application at `https://<web-server-host>:443/` and access the applications by clicking on the appropriate card.

   Design Studio is accessible only to users with a data_scientist role.

   Surveillance Dashboard is accessible only to users with a supervisor, investigator, or analyst role.

   Complaints Dashboard is accessible only to user with a supervisor role.

Integrating Case Manager

Surveillance Insight detects suspicious activities and registers a case with the Case Manager system. You must configure Surveillance Insight to integrate with the Case Manager.

Procedure

1. Log in to the Kubernetes master node as the root user.
2. Log in to the Liberty container.

   ```
   kubectl exec -it sifs-base-liberty-<id> /bin/bash
   ```

You can get the name of the pod by running the following command: `kubectl get pods`

3. After you are logged in to the pod, edit the `fci_sifs.properties` file.

   ```
   vi /home/sifsuser/casemanager/fci_sifs.properties
   ```

4. Edit the Case Manager host, port, user, and password information:

   ```
   fci_host={fci_host}
   fci_port={fci_port}
   fci_user={fci_user}
   fci_passsword={fci_passsword}
   ```

5. Replace `localhost` with the hostname of the Security Auth container.

   ```
   fci_security_auth_url=https://localhost:4010/security-auth/api/v1.0/login/ldap
   ```

6. Update the following values for the Case Manager database host, port, user, and password information. The password should be the xor-encoded values.

   ```
   fci_db_url={fci_db_url}
   fci_db_user={fci_db_user}
   fci_db_password={fci_db_password}
   fci_db_secured={fci_db_secured}
   ```

7. Save and close the file.
8. Restart the Liberty server inside the container.

   a) Go to the `/home/sifsuser/casemanager` directory.

   b) Run the following command to register Surveillance Insight with the Case Manager:

   ```
   java -cp "/home/sifsuser/casemanager/*" com.ibm.sifs.services.RegisterSIFS /home/sifsuser/
   casemanager/fci_sifs.properties
   ```

   c) Exit from the container.
9. If the Liberty container is stopped, repeat the above steps.

Install the big data content on HDP

IBM Surveillance Insight for Financial Services uses different big data components that you must install on the Hortonworks Data Platform (HDP) environment.

There are separate packages for each of the following big data components:

- Surveillance Insight Bigdata Content for Base
- Surveillance Insight Bigdata Content for EComm
- Surveillance Insight Bigdata Content for Trade
- Surveillance Insight Bigdata Content for Complaints

Installing Surveillance Insight big data content for the base components

Procedure

1. Create an `/opt/IBM` directory on each Hadoop node.

   For example,

   ```
   mkdir -p /opt/IBM
   ```

2. Log in to the Hadoop master node as the root user.
3. Download the big data base package (`CNT62EN.tar`) to the HDFS master node in the `/root` directory.

4. Extract `CNT62EN.tar`.

   This generates `IBM_Surveillance_Analytics_2.0.3_Multiplatform_English.zip`
5. Extract `IBM_Surveillance_Analytics_2.0.3_Multiplatform_English.zip` to the `/opt/IBM` directory.

   ```
   cd /root
   tar -xvf CNT62EN.tar
   unzip IBM_Surveillance_Analytics_2.0.3_Multiplatform_English.zip -d /opt/IBM/
   IBM_Surveillance_Analytics_2.0.3_Multiplatform_English
   ```
6. Go to the `/opt/IBM/IBM_Surveillance_Analytics_2.0.3_Multiplatform_English/bin` directory, and edit the `build.properties` file to contain:

   ```
   ANT_HOME=/opt/apache-ant-1.9.4
    BigData_Artifacts_Location=/opt/IBM/IBM_Surveillance_Analytics_2.0.3_Multiplatform_English
    script_Location=/opt/IBM/IBM_Surveillance_Analytics_2.0.3_Multiplatform_English/bin
    JAVA_HOME=/etc/alternatives/java_sdk_1.8.0_openjdk
   ```

   If you are using a different Java, set the JAVA_HOME value to the appropriate location. To get the path, you can run the `which java` command.
7. Edit the following values in the `Install_BigData.sh` file.

   ```
   export JAVA_HOME=/etc/alternatives/java_sdk_1.8.0_openjdk
   export ANT_HOME=/opt/apache-ant-1.9.4
   ```
8. Save and close the file.
9. Run the following command to change the script properties.

   ```
   cd /opt/IBM/IBM_Surveillance_Analytics_2.0.3_Multiplatform_English/bin
   ```

   ```
   dos2unix *.sh
   ```
10. Run the following command:

    ```
    sh Install_BigData.sh
    ```
11. As the root user, create a `security` directory the `/home/sifsuser` directory.

    ```
    mkdir -p /home/sifsuser/security
    ```
12. From the Kubernetes master node, copy the keystores in the `/fci-exports/sifs-liberty-instance/usr/servers/SIFSServer/resources/security/` to the HDFS master node in the `/home/sifsuser/security` directory.

    If the JKS files are different than the default ones, skip this step, and see "Update keystores" on page 69.
13. On the Kubernetes master node, run the following commands:

    ```
    cd /fci-exports/sifs-liberty-instance/usr/servers/SIFSServer/resources/security/
    ```

    ```
    scp * root@<hds master>:/home/sifsuser/security/
    ```
14. Copy the `db2jcc4-11.1.3.jar` file to the `/home/sifsuser/lib` on the HDFS master node.

    ```
    cp /opt/IBM/IBM_Surveillance_Analytics_2.0.3_Multiplatform_English/SIFSSLMTag/
    db2jcc4-11.1.3.jar /home/sifsuser/lib/
    ```
15. Delete the `spark-streaming_2.11-2.1.1.jar` file from the `/home/sifsuser/lib` directory.

    ```
    cd /home/sifsuser/lib
    ```

    ```
    rm spark-streaming_2.11-2.1.1.jar
    ```

16. Change the ownership to the sifsuser:

    ```
    chown -R sifsuser:sifsuser /home/sifsuser
    ```

Configuring the cron job

Hadoop is configured to access by using the sifsuser user account. The sifsuser ticket will expire every day, which will restrict access to HDFS unless the ticket is renewed. You can automate the ticket renewal by using a cron job.

Procedure

1. Log on to the Hadoop master node as the root user.
2. Copy `/opt/IBM/IBM_Surveillance_Analytics_2.0.3_Multiplatform_English/bin/renewTicket.sh` to the `/home/sifsuser` directory.

    ```
    cp /opt/IBM/IBM_Surveillance_Analytics_2.0.3_Multiplatform_English/bin/renewTicket.sh /home/
    sifsuser
    ```

3. Create a new file that is named `renewSolrTicket.sh` in the `/home/solruser` directory, and add the following content:

    ```
    kinit -kt /etc/security/keytabs/solr.keytab solruser/<ambari_server>
    ```

    Replace *<ambari_server>* with the Ambari server host name.
4. As the root user, run the following commands:

    ```
    chown sifsuser:sifsuser /home/sifsuser/renewTicket.sh
    ```

    ```
    chown solruser:solruser /home/solruser/renewSolrTicket.sh
    ```

    ```
    chmod 755 /home/sifsuser/renewTicket.sh /home/solruser/renewSolrTicket.sh
    ```

5. Copy the `/home/sifsuser/renewTicket.sh` and `/home/solruser/renewSolrTicket.sh` files to each Hadoop node.

    ```
    scp /home/sifsuser/renewTicket.sh root@<hadoop_cluster>:/home/sifsuser
    ```

    ```
    scp /home/sifsuser/renewSolrTicket.sh root@<hadoop_cluster>:/home/solruser
    ```

    Replace *<hadoop_cluster>* with the host name of the Hadoop cluster node.
6. Run the following command on each Hadoop node:

    ```
    crontab -u sifsuser -e
    ```

7. In the blank screen, enter the following content and save the file.

    ```
    00 09-21 * * * /home/sifsuser/renewTicket.sh
    ```

8. Run the following command on each Hadoop node:

    ```
    crontab -u solruser -e
    ```

9. In the blank screen, enter the following content and save the file.

    ```
    00 09-21 * * * /home/solruser/renewSolrTicket.sh
    ```

Installing the base machine learning artifacts

Procedure

1. Log on to the Hadoop master node as the root user.
2. Install the job service:
   a) Copy the analytics.jobapi file to the /home/sifsuser directory.

   ```
   cp -R /opt/IBM/IBM_Surveillance_Analytics_2.0.3_Multiplatform_English/analytics.jobapi /
   home/sifsuser
   ```

   b) Run the following command to change the ownership:

   ```
   chown -R sifsuser:sifsuser /home/sifsuser/analytics.jobapi
   ```

   c) Change to the sifsuser user.

   d) Go to the /home/sifsuser/analytics.jobapi directory.

   e) Run the following command:

   ```
   nohup /usr/local/bin/python3.5 submitJob.py > nohup &
   ```

   See the README.md file for information about the service.

   The jobs that you can submit from the Surveillance Dashboard are configured in the job.properties file. If any new jobs need to be submitted, the new job details must be configured in the job.properties file.
3. Install the email parsing service:
   a) As the root user, copy ml.emailparse to the /home/sifsuser directory.

   ```
   cp -R /opt/IBM/IBM_Surveillance_Analytics_2.0.3_Multiplatform_English/ml.emailparse /home/
   sifsuser
   ```

   b) Run the following command to change the ownership:

   ```
   chown -R sifsuser:sifsuser /home/sifsuser/ml.emailparse
   ```

   c) Change to the sifsuser user.

   d) Go to the /home/sifsuser/ml.emailparse directory.

   e) Run the following command:

   ```
   nohup /usr/local/bin/python3.5 mailparseRESTAPI.py > nohup &
   ```

   See the mailparse_RESTAPI_README.md file for information about the service.
4. Install the keywords service:
   a) Copy ml.keywords to the /home/sifsuser directory.

   ```
   cp -R /opt/IBM/IBM_Surveillance_Analytics_2.0.3_Multiplatform_English/ml.keywords /home/
   sifsuser/
   ```

   b) Run the following command to change the ownership:

   ```
   chown -R sifsuser:sifsuser /home/sifsuser/ml.keywords
   ```

   c) Change to the sifsuser user.

   d) Go to the /home/sifsuser/ml.keywords directory.

   e) Run the following command:

   ```
   nohup /usr/local/bin/python3.5  keywordsRESTAPI.py >nohup &
   ```

   See the RESTAPI_keywords_README.md file for information about the service.
5. Install the discovery service:

a) Copy `analytics.discovery` to the `/home/sifsuser` directory.

```
cp -R /opt/IBM/IBM_Surveillance_Analytics_2.0.3_Multiplatform_English/
analytics.discovery /home/sifsuser/
```

b) Run the following command to change the ownership:

```
chown -R sifsuser:sifsuser /home/sifsuser/analytics.discovery
```

c) Change to the sifsuser user.

d) Go to the `/home/sifsuser/analytics.discovery` directory.

e) Open the `config/REST.cfg` file and update the HOSTNAME property to point to your Db2 instance, and update the Db2 password with the xor-encoded Db2 user password.

f) Open the `config/discovery_services.cfg` file and update the CLIENT property to point to your HDFS instance.

```
CLIENT = http://<IP>:<PORT>
```

g) Run the following command:

```
nohup /usr/local/bin/python3.5 discovery_services.py >nohup &
```

See the `README.md` file for information about the service.

6. Install the ML sampling service:

a) Copy `ml.sampling` to the `/home/sifsuser` directory.

```
cp -R /opt/IBM/IBM_Surveillance_Analytics_2.0.3_Multiplatform_English/ml.sampling /home/
sifsuser/
```

b) Run the following command to change the ownership:

```
chown -R sifsuser:sifsuser /home/sifsuser/ml.sampling
```

c) On the Hadoop master node, change to the sifsuser user.

d) Run the following command:

```
hadoop fs -mkdir /user/sifuser/comm_raw
```

e) Copy the Yarn logger file to a directory on the HDFS system that is accessible by all nodes.

As the sifsuser, run the following command on the Hadoop master node:

```
hdfs dfs -put /home/sifsuser/ml.sampling/logger.py /user/sifsuser/comm_raw/
```

f) Go to the `/home/sifsuser/ml.sampling` directory.

g) Run the following commands:

```
cd /home/sifsuser/ml.sampling
```

```
cp /home/sifsuser/security/fci_universal_ks.crt.pem ./config/fci_universal_ks.crt.pem
```

```
zip myzip ./config/config.cfg ./config/fci_universal_ks.crt.pem
```

If you change the configuration file or the certificate, then you must repeat these steps.

If zip is not found, run the following command: `yum install zip`

h) Run the following commands to compress the Python modules so that you can copy them:

```
cd /home/sifsuser/ml.sampling
```

```
tar -zcvf samplemails.tar.gz samplemails
```

```
tar -zcvf parsemails.tar.gz parsemails
```

i) Copy the compressed files to the Hadoop master node as the root user.

```
/usr/local/lib/python3.5/site-packages/ cd /usr/local/lib/python3.5/site-packages/
```

```
chown root:root parsemails.tar.gz
```

```
chown root:root samplemails.tar.gz
```

```
tar -zxvf samplemails.tar.gz
```

```
tar -zxvf parsemails.tar.gz
```

```
chown sifsuser:sifsuser ibm_db*
```

```
cd /usr/local/lib/python3.5/site-packages/clidriver
```

```
chmod o+w cfg
```

j) Copy the compressed files to the remaining Hadoop nodes as the root user.

```
scp samplemails.tar.gz parsemails.tar.gz root@<HDPCluster>:/usr/local/lib/python3.5/site-
packages/
```

Change the *<HDPCluster>* value to the host name of the HDP cluster nodes.

k) On each of the Hadoop nodes in the cluster, change the ownership and decompress the packages:

```
cd /usr/local/lib/python3.5/site-packages/
```

```
chown root:root parsemails.tar.gz
```

```
chown root:root samplemails.tar.gz
```

```
tar -zxvf samplemails.tar.gz
```

```
tar -zxvf parsemails.tar.gz
```

```
chown sifsuser:sifsuser ibm_db*
```

```
cd /usr/local/lib/python3.5/site-packages/clidriver
```

```
chmod o+w cfg
```

Updating the Spark properties for the base components

Procedure

1. Log on to the Hadoop master node as the sifsuser user.

2. Open the `/home/sifsuser/lib/sifs.spark.properties` file and edit the appropriate properties:

   - Update the following IP addresses and port numbers with the Kubernetes host name and port number for Liberty:

     ```
     InferenceREST=https://<IP>:<PORT>/SIFSModelServices/inference/model/predict
     CreateAlertREST=https://<IP>:<PORT>/SIFSServices/alertservice/v1/alert/createAlert
     CreateEvidenceREST=https://<IP>:<PORT>/SIFSServices/alertservice/v1/alert/createEvidence
     UpdateAlertREST=https://<IP>:<PORT>/SIFSServices/alertservice/v1/alert/updateAlert
     ```

   - Replace *<IP>* with the host name of Kubernetes master node and *<PORT>* with the appropriate port number.
   - Update the following IP address and port number with the Kubernetes host name and port number for Db2:

     ```
     db2jdbcurl=jdbc:db2://<IP>:<PORT>/SIFS:sslConnection=true;currentSchema=SIFS;
     ```

   - Replace *localhost* with the host name of Kubernetes master node:

     ```
     SolrProxyURL=https://localhost:<PORT>/SIFSServices/surveillanceui/v1/index/update
     ```

   - Replace *<IP>*`:8020` with the fcicluster:

     ```
     HDFSFilePath=hdfs://<IP>:8020/user/sifsuser/
     ```

   - Update the following IP address and port number with the Kafka host name and port number:

     ```
     metadata.broker.list=<IP>:6667
     bootstrap.servers=<IP>:6667
     ```

   - Update the rules and dictionary paths:

     ```
     dictPath=/home/sifsuser/config/dict
     rulesPath=/home/sifsuser/config/rules
     ```

   If the JKS and the truststore and keystore passwords have changed, see "Update keystores" on page 69.

3. Copy the contents of the `sifsuser` directory to all of the nodes in the cluster.

   a) Save the changes to `sifs.spark.properties` for all the components.

   b) As the sifsuser, copy the `/home/sifsuser` directory from the Hadoop master node to all nodes in the HDP cluster.

   ```
   scp -r /home/sifsuser/* sifsuser@<hdp_cluster>:/home/sifsuser/
   ```

   Replace *<hdp_cluster>* with the host names of the nodes in your HDP cluster.

4. Add PYSPARK_PYTHON to the `.bashrc` file for the sifsuser.

   a) As sifsuser, log in to each node in the Hadoop cluster, and edit the `.bashrc` file.

   ```
   vi ~/.bashrc
   ```

   b) Add the following:

   ```
   PYSPARK_PYTHON=/usr/local/bin/python3.5
   ```

   c) Save and close the file.

Chapter 5. Installing IBM Ecomm Surveillance Insight for Financial Services

Procedure

1. Save the ecomm big data package (CNT5EEN.tar) to the Hadoop master node in the /root directory.
2. Extract the file to the /opt/IBM directory. The extraction generates a file that is named FinancialMkts_SurveillanceInsight_BigDataContent_EComm.zip.

   ```
   cd /root
   ```

   ```
   tar -xvzf CNT5EEN.tar
   ```

   ```
   unzip FinancialMkts_SurveillanceInsight_BigDataContent_EComm.zip -d /opt/IBM/
   FinancialMkts_SurveillanceInsight_BigDataContent_EComm
   ```

3. Go to the /opt/IBM/FinancialMkts_SurveillanceInsight_BigDataContent_EComm/ecomm/bin directory, and edit the build.properties file to contain:

   ```
   JAVA_HOME=/etc/alternatives/java_sdk_1.8.0_openjdk
   ANT_HOME=/opt/apache-ant-1.9.4
   BigData_Artifacts_Location=/opt/IBM/FinancialMkts_SurveillanceInsight_BigDataContent_EComm/
   ecomm/
   script_Location=/opt/IBM/FinancialMkts_SurveillanceInsight_BigDataContent_EComm/ecomm/bin
   ```

   If you are using a different Java, set the JAVA_HOME value to the appropriate location. To get the path, you can run the which java command.
4. Edit the following values in the Install_BigData.sh file.

   ```
   export JAVA_HOME=/etc/alternatives/java_sdk_1.8.0_openjdk
   export ANT_HOME=/opt/apache-ant-1.9.4
   ```

5. Save and close the file.
6. Open Copy_files.sh and comment out the following line, and save the file.

   ```
   sudo cp -avr $BigData_Artifacts_Location/lib/ $SIFS_User_HomeDir/
   ```

7. Run the following command to change the script properties.

   ```
   dos2unix *.sh
   ```

8. Run the following command:

   ```
   sh Install_BigData.sh
   ```

9. As the root user, run the following commands to copy the directories to the /home/sifsuser directory.

   ```
   cd /opt/IBM/FinancialMkts_SurveillanceInsight_BigDataContent_EComm/ecomm
   ```

   ```
   cp -R config /home/sifsuser/
   ```

   ```
   cp -R config/lexicons /home/sifsuser/
   ```

10. As the root user, copy the Spark scripts to the /home/sifsuser directory.

    ```
    cd /opt/IBM/FinancialMkts_SurveillanceInsight_BigDataContent_EComm/ecomm/bin
    ```

    ```
    cp *.sh /home/sifsuser/lib
    ```

    ```
    chmod 755 /home/sifsuser/lib/*.sh
    ```

11. Change the ownership to the sifsuser:

    ```
    chown -R sifsuser:sifsuser /home/sifsuser
    ```

Installing the ecomm machine learning artifacts

Procedure

1. Log on to the Hadoop master node as the root user.
2. Install the entity extraction service:

   a) Remove the existing analytics.entityextraction.spaCy entity from the /home/sifsuser directory.

   ```
   rm -rf /home/sifsuser/analytics.entityextraction.spaCy
   ```

   b) Copy analytics.entityextraction.spaCy to the /home/sifsuser directory.

   ```
   cp -R /opt/IBM/FinancialMkts_SurveillanceInsight_BigDataContent_EComm/ecomm/
   analytics.entityextraction.spaCy /home/sifsuser
   ```

   c) Run the following command to change the ownership:

   ```
   chown -R sifsuser:sifsuser /home/sifsuser/analytics.entityextraction.spaCy
   ```

   d) Remove the config directory.

   ```
   rm -rf /home/sifsuser/analytics.entityextraction.spaCy/config
   ```

   e) Change to the sifsuser.

   f) Go to the /home/sifsuser/analytics.entityextraction.spaCy directory.

   g) Run the following command to start the service:

   ```
   nohup /usr/local/bin/python3.5 EntityExtraction_spaCy_RESTapi.py > nohup  &
   ```

3. Install the natural language service:

   a) Copy analytics.nlpframework to the /home/sifsuser directory.

   ```
   cp -R /opt/IBM/FinancialMkts_SurveillanceInsight_BigDataContent_EComm/ecomm/
   analytics.nlpframework /home/sifsuser
   ```

   b) Run the following command to change the ownership:

   ```
   chown -R sifsuser:sifsuser /home/sifsuser/analytics.nlpframework
   ```

   c) Change to the sifsuser user.

   d) Go to the /home/sifsuser/config/properties directory.

   e) Edit the nlp.properties file and update the hdfs_url value. Update *<host_name>* and *<port_no>* with the correct values for your HDFS server. The default port number is 50070.

f) Ensure that /home/sifsuser/ml.emailparse exists. If it does not, you must install the base machine learning artifacts ("Installing the base machine learning artifacts" on page 36).

g) Go to the /home/sifsuser/analytics.nlpframework directory.

h) Run the following commands:

```
nohup /usr/local/bin/python3.5 RestApi.py > nohup &
```

```
nohup /usr/local/bin/python3.5 NLUIntegrated.py > noup_nlu &
```

Updating the Spark properties for the ecomm components

Procedure

1. Log on to the Hadoop master node as the sifsuser user.
2. Create a spark-warehouse directory in the /usr/hdp/current/spark2-client/bin directory.

   ```
   cd /usr/hdp/current/spark2-client/bin/
   ```

   ```
   mkdir spark-warehouse
   ```

3. Open the /home/sifsuser/lib/sifs.spark.properties file and edit the appropriate properties:

   - Update the following IP addresses and port numbers with the Kubernetes host name and port number for Liberty:

     ```
     policyServiceUrl=https://<IP>:<PORT>/CommServices/ecomm/policy
     CreateCommREST=https://<IP>:<PORT>/SIFSServices/commservice/v1/createComm
     iterationServiceUrl=https://<IP>:<PORT>/SIFSModelServices/discoverymodel/iterations
     sampleServiceUrl=https://<IP>:<PORT>/SIFSServices/commservice/v1/communication/
     {communicationid}/createTag
     ```

   - Update the following IP address and port number with the Hadoop master node and the Python Entity Extraction Service port number. The default port number is 5003.

     ```
     entityServiceUrl=https://<IP>:<PORT>/analytics/models/v1/analyzetext/
     ```

   - Update the following IP address and port number with the Hadoop master node and the Python Discovery Service port number. The default port number is 5004.

     ```
     discoveryServiceUrl=https://<IP>:<PORT>/discovery/v1/analyse
     ```

   - Update the following IP address and port number with the Hadoop master node and the Python NLC Service port number. The default port number is 5001.

     ```
     nlcServiceUrl=https://<IP>:<PORT>/nlc/v1/models/
     ```

   - Update the path with the spark-warehouse directory path:

     ```
     sparkWarehousePath=file:////usr/hdp/current/spark2-client/bin/spark-warehouse
     ```

4. Save and close the file.
5. As the sifsuser, copy the /home/sifsuser directory from the Hadoop master node to all nodes in the HDP cluster.

   ```
   scp -r /home/sifsuser/* sifsuser@<hdp_cluster>:/home/sifsuser/
   ```

   Replace <*hdp_cluster*> with the host names of the nodes in your HDP cluster.

Create Kafka topics for the ecomm components

Procedure

Log in to the Kafka node as the root user, and run the following commands:

```
export KAFKA_OPTS="-Djava.security.auth.login.config=/usr/hdp/2.6.4.0-91/kafka/conf/
kafka_client_jaas_sifs.conf"
```

```
/usr/hdp/2.6.4.0-91/kafka/bin/kafka-topics.sh --create --zookeeper <zookeeper_host>:2181 --
replication-factor 1 --partitions 1  --topic sifs.alert.in
```

```
/usr/hdp/2.6.4.0-91/kafka/bin/kafka-topics.sh --create --zookeeper <zookeeper_host>:2181 --
replication-factor 1 --partitions 1  --topic sifs.chat.in
```

```
/usr/hdp/2.6.4.0-91/kafka/bin/kafka-topics.sh --create --zookeeper <zookeeper_host>:2181 --
replication-factor 1 --partitions 1  --topic sifs.ecomm.in
```

```
/usr/hdp/2.6.4.0-91/kafka/bin/kafka-topics.sh --create --zookeeper <zookeeper_host>:2181 --
replication-factor 1 --partitions 1  --topic sifs.email.in
```

```
/usr/hdp/2.6.4.0-91/kafka/bin/kafka-topics.sh --create --zookeeper <zookeeper_host>:2181 --
replication-factor 1 --partitions 1  --topic sifs.policy.in
```

Replace *<zookeeper_host>* with the appropriate host name.

If you are not using port 2181, ensure that you update the port number.

Run the Spark jobs for ecomm

The ecomm Spark job artifacts are located in the `/home/sifsuser/lib` directory. You must run the scripts as the sifsuser user.

- To read and process email data from Kafka:

  ```
  ./processEmail.sh
  ```

- To read and process chat data from Kafka

  ```
  ./processChat.sh
  ```

- To read and process voice data from Kafka

  ```
  ./processCommunication.sh
  ```

- To run inference for communication data for a given date (yyyy-mm-dd)

  ```
  ./analyzeComm.sh <date>
  ```

- To compute a profile for communication data for a given date (yyyy-mm-dd)

  ```
  ./computeProfile.sh <date>
  ```

- To compute party risk score

  ```
  ./partyRiskScoring.sh
  ```

Run the Spark jobs for the discover service

The following Spark jobs are for Discovery processing. The jobs are run from the /home/sifsuser/lib directory.

The jobs can also be triggered from the **Admin** tab of the Surveillance Dashboard.

1. To run discovery for batch of emails.

   ```
   ./processDiscovery.sh <emaildataset> <modelid> <iterationid>
   ```

   The emails must be in HDFS.
2. To run unsupervised learning for a batch of emails.

   ```
   ./processUnsupervised.sh <emaildataset> <emailfeaturesdataset> <rules file>
   ```

   The emails must be in HDFS.
3. To sample email features data.

   ```
   ./processSampling.sh <emailfeaturesdataset> <number of dataset> <rows per dataset> <ratio>
   <datasetlabel>
   ```

4. To generate NLC training file.

   ```
   ./processSampling_NLC.sh <emailfeaturesdataset> <outputdataset> <dataset to be used> <tags>
   <outputnumber>
   ```

5. To run the ecomm pipeline for a batch of emails.

   ```
   ./processEcommPipeline.sh <emaildataset>
   ```

   The emails must be in HDFS.

Loading ecomm sample data

Procedure

1. Load the policy data:
   a) Log in to the Hadoop master node as the root user.

   b) Go to the /opt/IBM/IBM_Surveillance_Analytics_2.0.3_Multiplatform_English/Sample_Data/EComm/Policy directory.

   c) Run the following commands:

   ```
   curl -k -H 'Content-Type: application/json' -H 'auth:digest' -H 'source:Actiance' -X POST
   --data-binary @policy.json -v --user ibmrest1:ibmrest@pwd1 --digest https://
   <localhost>:<port>/CommServices/ecomm/policy
   ```

   ```
   curl -k -H 'Content-Type: application/json' -H 'auth:digest' -H 'source:Actiance' -X POST
   --data-binary @policy2.json -v --user ibmrest1:ibmrest@pwd1 --digest https://
   <localhost>:<port>/CommServices/ecomm/policy
   ```

   Update *<localhost>*:*<port>* with the host name of the Kubernetes master and the Liberty port. By default, the port is 8981.
2. Load the email data:
   a) Log in to the Hadoop master node as the root user.

   b) Copy the snapshots folder from Db2 Docker to the Hadoop master node.

   The snapshots directory is in /opt/ibm/sifs/sql/sampledata in the Db2 Docker.

    snapshots must be copied to /opt/IBM/FinancialMkts_SurveillanceInsight_BigDataContent_Base-2.0.3/Sample_Data/EComm/sampledata on the Hadoop master node.

   c) Go to the /opt/IBM/IBM_Surveillance_Analytics_2.0.3_Multiplatform_English/Sample_Data/EComm/snapshots directory.
   d) Open load_data.sh and change the localhost value to the Kubernetes master node IP address or hostname, and change 9443 to the Kubernetes port number. By default, the port is 8981.
   e) Ensure that the processEmail.sh Spark job is running.
   f) Run the following command:

      ```
      sh load_data.sh <date>
      ```

      For example,

      ```
      sh load_data.sh 2017-04-01
      ```

      This loads the data for all of the dates starting from 2017-04-01 until 2018-01-23 in a sequential manner and load data for one day at a time. After the data for one day is processed, data for the next day is loaded.

3. Load the chat data:
   a) Log in to the Hadoop master node as the root user.
   b) Go to the /opt/IBM/IBM_Surveillance_Analytics_2.0.3_Multiplatform_English/Sample_Data/EComm/SnapshotChats directory.
   c) Open load_data.sh and change the localhost value to the Kubernetes master node IP address or hostname, and change 9443 to the Kubernetes port number. By default, the port is 8981.
   d) Ensure that the processChat.sh Spark job is running.
   e) Run the following command

      ```
      sh load_data.sh <date>
      ```

      For example,

      ```
      sh load_data.sh snapshot_chat_1.txt
      ```

      Load the data for all of the chat messages starting from snapshot_chat_1.txt, snapshot_chat_2.txt, and snapshot_chat_3.txt in order.

Loading the discovery sample data

Procedure

1. Log in to the Hadoop master node as the sifsuser.
2. Create a discovery directory for HDFS.

   ```
   hdfs dfs -mkdir /user/sifsuser/discovery-model
   ```

3. Load the data.

   Copy the snapshots_discovery directory from the Db2 Docker in /opt/ibm/sifs/sql/sampledata to the Hadoop master node in the /opt/IBM/

FinancialMkts_SurveillanceInsight_BigDataContent_Base-2.0.3/Sample_Data/EComm/sampledata directory.

```
hdfs dfs -put /opt/IBM/FinancialMkts_SurveillanceInsight_BigDataContent_Base-2.0.3/
Sample_Data/EComm/sampledata/snapshots_discovery /user/sifsuser/discovery-model
```

4. Create the discovery model.
   a) Log in to the Surveillance Insight URL.
   b) On the **Discovery** tab, click **Create Model**.
   c) Enter a name and a description, and click **Create**.
5. Create the discovery dataset and publish the lexicons.
   a) On the **Discovery** tab, click **Create Dataset**.
   b) Enter a name, description, and a path, and click **Create**.

      The dataset path is the HDFS directory where the emails are stored.
   c) On the **Discovery** tab, click **Models**, and select a model.
   d) When the iteration status is **Discovery Done**, click **View**.

      The context, action verb, object, and emotion for the dataset that you uploaded is displayed.
   e) From each category, select a few entities and associate these entities to new or existing lexicons.
   f) Click **Save**, and then **Finish**.

      For example, under **Context** select terms like **Wells**, **loan**, **payment** and more, and then click **New Lexicon** > **Bank**. On the similar lines create a few more lexicons, such as Customer and Loan, and include the required entities. Click **Finish**.
   g) From the **Lexicon** tab, click **Save** and publish the lexicons models.
   h) Copy the saved lexicons from the Liberty container on the Kubernetes master node to all of the HDP cluster nodes. Copy all lexicons to the /home/sifsuser/lexicons directory on each HDP node.
6. Create the rules.
   a) Log in to the HDFS master node as the sifsuser.
   b) Create a rule file named complaint.rules in the /home/sifsuser/config/rules/ directory on the HDP master cluster.

      The file should contain:

      ```
      tag=complaints,non-complaints
      rule1=(context == 'Customer' || context == 'Bank'|| context == 'Loan')
      ```

      The rules file can be edited as per your lexicons.
   c) Copy the file and path to each node in the HDP cluster.
   d) Ensure that the sifs.spark.properties file in the /home/sifsuser/lib directory has the rulesPath value set to /home/sifsuser/config/rules/.

Chapter 6. Installing IBM Voice Surveillance Insight for Financial Services

You install IBM Voice Surveillance Insight for Financial Services after you install the base components.

Procedure

1. Log in to the Kubernetes master node computer as the root user.
2. Create a directory for the installation files.

   ```
   mkdir /fcimedia/sifs-voice
   ```

3. Save the downloaded IBM Surveillance Insight for Financial Services eImages to the `/fcimedia/sifs-voice` directory.

   You must download `CNT5IEN.tar`, `CNT5JEN.tar`, , `CNT5KEN.tar`, and `SHA1SUM.tar` to the `/fcimedia/sifs-voice` directory.
4. Go to the `/fcimedia/sifs-voice` directory.
5. Run the following commands:

   ```
   tar xf CNT5IEN.tar
   ```

   ```
   tar xzf sifs-voice-install-kit.tar.gz
   ```

   ```
   rm -rf sifs-voice-install-kit.tar.gz
   ```

   ```
   cd /fcimedia/sifs-voice && sha1sum *.tar > ~/SIFS_V2.0.3_Checksum.sha1 && mv ~/
   SIFS_V2.0.3_Checksum.sha1 .
   ```

6. Run the following command to generate a checksum file for the installation:

   ```
   cd /fcimedia/sifs-voice && sha1sum *.tar > ~/SIFS_V2.0.3_Checksum.sha1 && mv ~/
   SIFS_V2.0.3_Checksum.sha1 .
   ```

7. Copy the `CNT2KEN.tar` file from the `/fcimedia/fcco` directory to the `/fcimedia/sifs-voice` directory.

   `CNT2KEN.tar` contains the Helm charts that are used for the installation.

   ```
   cp /fcimedia/fcco/CNT2KEN.tar /fcimedia/sifs-voice
   ```

8. Go to the `helm` directory where you extracted the installation files:

   ```
   cd /fcimedia/sifs-voice/sifs-voice-install-kit/helm/
   ```

9. Copy the `install.properties` file from the IBM Financial Crimes Insight installation file directory to the `helm` directory.

   ```
   cp /fcimedia/fcco/fci-install-kit/helm/install.hosts.properties .
   ```

10. Open the `install.properties` file in a text editor.

    a) Edit the following values for your environment:

       ```
       external.docker.registry.url = (your docker
       registry):5000
       external.nfsserver = (your NFS server)
       ```

b) Uncomment the following lines:

```
external.docker.registry.isSecure = false
external.docker.registry.isPreloaded = false
```

c) Add the following line:

```
docker.registry.file = sifs-docker-registry.tar
```

d) Optional: If you are using Custom-NFS mounting, add the following lines. If you are not, then you can skip to step 13.

```
configure.nfsserver = true
nfsserver.master.mount= /fci-exports
mount_point.1 = --path /sifs-voice-instance
```

e) Save and close the file.

11. Optional: If you are using Custom-NFS mounting, create a new file that is named `CustomNFS-si3.yaml`, and save it to the `helm` directory. Add the following contents to the file:

```
streamsPvNfsPath: /fcisi/sifs-voice-instance
libertystreamsPvNfsPath: /fcisi/sifs-liberty-streams-instance
```

12. Optional: If you are using Custom-NFS mounting, modify the `chart.args` files to include the following:

```
chart.args = -f sifs-voice-values.yaml --set global.coreReleaseName=fcco --set
global.sifsBaseReleaseName=sifs-base -f CustomNFS-si3.yaml
```

13. Run the following command from the `/fcimedia/sifs-voice/sifs-voice-install-kit/helm` directory:

```
./install.sh
```

Updating the HDP mount point configuration

Procedure

1. Log in to the computer where the NFS mounts are created.
2. Update Liberty Docker.

   a) Go to `/fci-exports/sifs-liberty-instance/config/kafka.properties`.

   b) Update the bootstrap.servers value.

   c) Copy the `krb5.conf` file from the hadoop.master computer to the Liberty Docker `/etc/` directory.

   d) Copy `/etc/security/keytabs` to `/fci-exports/sifs-liberty-instance/`.

   e) In `sifs-jaas.conf`, under `/fcisi/sifs-liberty-instance/config`, update the useKeytab and principal values.

   f) From the hadoop.master computer, copy the `/usr/hdp/2.6.4.0-91/hadoop/` directory to `/fci-exports/sifs-liberty-instance/`.

   For example,

   ```
   scp /usr/hdp/2.6.4.0-91/hadoop/ root@<nfs server>:/fci-exports/sifs-liberty-instance/
   ```

3. Update the trade Docker.

   a) Go to the `/fcisi/sifs-trade-instance/config/properties` on the mount point.

   b) Update bootstrap.servers value in `consumer.properties` and `producer.properties`.

   c) Copy the `krb5.conf` file from the hadoop.master computer to the trade Docker `/etc/` directory.

d) Copy /etc/security/keytabs to /fci-exports/sifs-liberty-instance/.

e) In sifs-jaas.conf, under /fcisi/sifs-trade-instance/config/properties, update the useKeytab and principal values.

f) From the hadoop.master computer, copy the /usr/hdp/2.6.4.0-91/hadoop/ directory to /fci-exports/sifs-liberty-instance/.

For example,

```
scp /usr/hdp/2.6.4.0-91/hadoop/ root@<nfs server>:/fci-exports/sifs-trade-instance/
```

4. Update the voice Docker.

a) Go to the /fcisi/sifs-voice-instance/config/properties on the mount point.

b) Update bootstrap.servers value in consumer.properties and producer.properties.

c) Copy the krb5.conf file from the hadoop.master computer to the voice Docker /etc/ directory.

d) Copy /etc/security/keytabs to /fci-exports/sifs-liberty-instance/.

e) In sifs-jaas.conf, under /fcisi/sifs-voice-instance/config/properties, update the useKeytab and principal values.

f) From the hadoop.master computer, copy the /usr/hdp/2.6.4.0-91/hadoop/ directory to /fci-exports/sifs-voice-instance/.

For example,

```
scp /usr/hdp/2.6.4.0-91/hadoop/ root@<nfs server>:/fci-exports/sifs-voice-instance/
```

Create Kafka topics for the voice components

The Kafka topics for the ecomm components must already exist before you create the Kafka topics for voice. For more information, see "Create Kafka topics for the ecomm components" on page 44.

Procedure

Log in to the Kafka node as the root user, and run the following commands:

```
export KAFKA_OPTS="-Djava.security.auth.login.config=/usr/hdp/2.6.4.0-91/kafka/conf/
kafka_client_jaas_sifs.conf
```

```
/usr/hdp/2.6.4.0-91/kafka/bin/kafka-topics.sh --create --zookeeper <zookeeper_host>:2181 --
replication-factor 1 --partitions 1  --topic sifs.voice.in
```

Replace *<zookeeper_host>* with the appropriate host name.

If you are not using port 2181, ensure that you update the port number.

Running the WAVAdaptor Streams job

Procedure

1. As the streamsadmin user, go in to the Streams voice container.
2. Go to the /home/streamsadmin/components/Voice_203/WAVAdaptor/Voice directory.
3. Edit the submitjob.sh file to update the following properties for your environment:
   - PARTYSEARCHURL
   - EXPORTALLURL

   - EXPORTMETADATAURL
   - EXPORTTRANSCRIPTURL
   - EXPORTAUDIOURL
4. Change HDFSAUTHKEYTAB=/home/streamsadmin/sifsuser.keytab to HDFSAUTHKEYTAB=/etc/security/keytabs/sifsuser.keytab
5. Save and close the file.
6. Run the following command:

   ```
   sh submitjob.sh
   ```

Running the PCAP Streams job

Procedure

1. As the streamsadmin user, go in to the Streams voice container.
2. Go to the /home/streamsadmin/components/Voice_203/PCAPSpeech directory.
3. Edit the submitjob.sh file to update the following properties for your environment:
   - EXPORTALLURL
   - EXPORTMETADATAURL
   - EXPORTTRANSCRIPTURL
   - EXPORTAUDIOURL
4. Change HDFSAUTHKEYTAB=/home/streamsadmin/sifsuser.keytab to HDFSAUTHKEYTAB=/etc/security/keytabs/sifsuser.keytab
5. Save and close the file.
6. Run the following command:

   ```
   sh submitPCAPSpeech.sh
   ```

7. Edit the submitPrepareCall.sh file to update the PARTYSEARCHURL property for your environment.
8. Change HDFSAUTHKEYTAB=/home/streamsadmin/sifsuser.keytab to HDFSAUTHKEYTAB=/etc/security/keytabs/sifsuser.keytab
9. Save and close the file.
10. Run the following command:

    ```
    sh submitPCAPSpeech.sh
    ```

11. Go to the /home/streamsadmin/components/Voice_203/IPC directory.
12. Edit the submitjob.sh file to update the following properties for your environment:
    - BW_SERVER_URL
    - BW_USER
    - BW_PASSWORD
    - SUBNETS
    - LOGINNAMEFILTER
13. Save and close the file.
14. Run the following command:

    ```
    sh submit.sh
    ```

15. Go to the `/home/streamsadmin/components/Voice_203/PCAPAdaptor` directory.
16. Edit the `submitjob.sh` file to update the following properties for your environment:
    - SUBNETS
    - CONNHOST
    - CONNPORT
17. Save and close the file.
18. Run the following command:

    ```
    sh submit.sh
    ```

Loading voice sample data

Procedure

1. Log in to the HDP master node as the root user.
2. Go to the `/opt/IBM/FinancialMkts_SurveillanceInsight_BigDataContent_Base-2.0.3/Sample_Data/Voice` directory.
3. Open `processvoice.sh` in an editor.
4. Replace localhost with the host name or IP address of the Kubernetes master node.
5. Save and close the file.
6. Ensure that the `processCommunication.sh` Spark job is running.
7. Run each curl command that is listed in the `load_data.sh` file separately and in sequence.

Chapter 7. Installing IBM Trade Surveillance Insight for Financial Services

You install IBM Trade Surveillance Insight for Financial Services after you install the base components.

Procedure

1. Log on to the Kubernetes master node computer as the root user.
2. Create a directory for the installation files.

   ```
   mkdir /fcimedia/sifs-trade
   ```

3. Save the downloaded IBM Surveillance Insight for Financial Services eImages to the `/fcimedia/sifs-trade` directory.

   You must download `CNT5FEN.tar`, `CNT5GEN.tar`, , `CNT5HEN.tar`, and `SHA1SUM.tar` to the `/fcimedia/sifs-trade` directory.

4. Go to the `/fcimedia/sifs-trade` directory.
5. Run the following commands:

   ```
   tar xf CNT5FEN.tar
   ```

   ```
   tar xzf sifs-trade-install-kit.tar.gz
   ```

   ```
   rm -rf sifs-trade-install-kit.tar.gz
   ```

6. Run the following command to generate a checksum file for the installation:

   ```
   cd /fcimedia/sifs-trade && sha1sum *.tar > ~/SIFS_V2.0.3_Checksum.sha1 && mv ~/
   SIFS_V2.0.3_Checksum.sha1 .
   ```

7. Copy the `CNT2KEN.tar` file from the `/fcimedia/fcco` directory to the `/fcimedia/sifs-trade` directory.

   `CNT2KEN.tar` contains the Helm charts that are used for the installation.

   ```
   cp /fcimedia/fcco/CNT2KEN.tar /fcimedia/sifs-trade
   ```

8. Go to the `helm` directory where you extracted the installation files:

   ```
   cd /fcimedia/sifs-trade/sifs-trade-install-kit/helm/
   ```

9. Copy the `install.properties` file from the IBM Financial Crimes Insight installation file directory to the `helm` directory.

   ```
   cp /fcimedia/fcco/fci-install-kit/helm/install.hosts.properties .
   ```

10. Open the `install.properties` file in a text editor.

    a) Edit the following values for your environment:

    ```
    external.docker.registry.url = (your docker
    registry):5000
    external.nfsserver = (your NFS server)
    ```

    b) Uncomment the following lines:

    ```
    external.docker.registry.isSecure = false
    external.docker.registry.isPreloaded = false
    ```

c) Add the following line:

```
docker.registry.file = sifs-docker-registry.tar
```

d) Optional: If you are using Custom-NFS mounting, add the following lines. If you are not, then you can skip to step 13.

```
configure.nfsserver =
true
nfsserver.master.mount= /fci-
exports
mount_point.1 = --path /sifs-trade-instance
```

e) Save and close the file.

11. Optional: If you are using Custom-NFS mounting, create a new file that is named `CustomNFS-si2.yaml`, and save it to the `helm` directory. Add the following contents to the file:

    ```
    streamsPvNfsPath: /fcisi/sifs-trade-instance
    ```

12. Optional: If you are using Custom-NFS mounting, modify the `chart.args` files to include the following:

    ```
    chart.args = -f sifs-trade-values.yaml --set global.coreReleaseName=fcco --set
    global.sifsBaseReleaseName=sifs-base -f CustomNFS-si2.yaml
    ```

13. Run the following command from the `/fcimedia/sifs-trade/sifs-trade-install-kit/helm/` directory:

    ```
    ./install.sh
    ```

Installing the Trade artifacts

Procedure

1. Log in to the Hadoop master node as the root user.
2. Copy the trade big data package to the Hadoop master node and run the following commands to extract the package.

   ```
   cd /opt/IBM
   ```

   ```
   unzip IBM_Trade_Surveillance_Analytics_2.0.3_Multiplatform_English.zip -d
   IBM_Trade_Surveillance_Analytics_2.0.3_Multiplatform_English
   ```

   ```
   cd /opt/IBM/IBM_Trade_Surveillance_Analytics_2.0.3_Multiplatform_English/trade/bin
   ```

3. Open `build.properties` and update the following properties:

   ```
   ANT_HOME=/opt/apache-ant-1.9.4
   BigData_Artifacts_Location=/opt/IBM/
   IBM_Trade_Surveillance_Analytics_2.0.3_Multiplatform_English/trade
   script_Location=/opt/IBM/IBM_Trade_Surveillance_Analytics_2.0.3_Multiplatform_English/trade/
   bin
   JAVA_HOME=/etc/alternatives/java_sdk_1.8.0_openjdk
   ```

   If you are using a different Java, set the JAVA_HOME value to the appropriate location. To get the path, you can run the `which java` command.

4. Run the following command to change the script properties.

   ```
   dos2unix *.sh
   ```

5. Run the following command:

   ```
   sh Install_BigData.sh
   ```

Create Kafka topics for the trade components

Procedure

1. Update the `sifs.spark.properties` file.
   a) Replace the following properties with the sample values.

   ```
   MinOrderMarketShare=4
   FrontOrderTimeRange=60
   TradeDataSource=SI
   FrontOrderMinScore=0.5
   OrderProximityMinScore=0.3
   FrontOrderQtyThreshold=30000
   ```

   b) Replace the IP and port number with the HDP node values for Hive.

   ```
   FTRHiveMetastoreUris=thrift://<IP>:<PORT>
   FTRHiveAddress=jdbc:hive2://
   <FTR_IP>:<FTR_PORT>/;serviceDiscoveryMode=zooKeeper;zooKeeperNamespace=<HIVE_SERVERNAME>
   ```

   c) Save and close the file.
2. As the sifsuser, copy the `/home/sifsuser` directory from the Hadoop master node to all of the other nodes in the HDP cluster.

   ```
   scp -r /home/sifsuser/* sifsuser@<hdp_cluster>:/home/sifsuser/
   ```

   Replace the *<hdp_cluster>* with the host names of the HDP cluster nodes.

   The Spark properties for the base and ecomm components must be configured for trade.
3. Create the Kafka topics for the trade components.
   a) Log in to the Kafka node as the root user, and run the following commands:

   ```
   export KAFKA_OPTS="-Djava.security.auth.login.config=/usr/hdp/2.6.4.0-91/kafka/conf/
   kafka_client_jaas_sifs.conf
   ```

   ```
   /usr/hdp/2.6.4.0-91/kafka/bin/kafka-topics.sh --create --zookeeper <zookeeper_host>:2181
   --replication-factor 1 --partitions 1  --topic sifs.pnd.alert.in
   ```

   ```
   /usr/hdp/2.6.4.0-91/kafka/bin/kafka-topics.sh --create --zookeeper <zookeeper_host>:2181
   --replication-factor 1 --partitions 1  --topic sifs.spoofing.alert.in
   ```

   ```
   /usr/hdp/2.6.4.0-91/kafka/bin/kafka-topics.sh --create --zookeeper <zookeeper_host>:2181
   --replication-factor 1 --partitions 1  --topic sifs.trade.evidence.in
   ```

   Replace *<zookeeper_host>* with the appropriate host name.

   If you are not using port 2181, ensure that you update the port number.

Run the Spark jobs for trade

The trade Spark jobs are located in the `/home/sifsuser/lib` directory. Run the trade Spark jobs as the sifsuser user.

1. To run PnDCollectEvidence

   ```
   ./runspark_pndcollectevidence.sh
   ```

2. To run spoofing evidence

   ```
   ./runspark_spoofingevidence.sh
   ```

3. To run the off market jobs

   Trade evidence persistence off market

   ```
   ./runspark_tradeevidencepersistance.sh
   ```

   Off market inference

   ```
   ./runspark_offmarket.sh
   ```

4. To run the front running jobs

   Front running trade data processing for Surveillance Insight

   ```
   ./runspark_frontrunning_si.sh
   ```

   Front running trade date for Financial Transaction Repository

   ```
   ./runspark_frontrunning_ftr.sh
   ```

   Front running inference

   ```
   ./runspark_frontrunning_inference.sh
   ```

Loading trade sample data

Sample data is provided for the off-market, spoofing, and pump-and-dump use cases.

Procedure

1. Load the off-market sample data:
   a) Log on to the HDFS master node as the sifsuser.
   b) Run the following commands to create the directories:

   ```
   hdfs dfs -mkdir /user/sifsuser/transactions/
   ```

   ```
   hdfs dfs -mkdir /user/sifsuser/marketReference/
   ```

   ```
   hdfs dfs -mkdir /user/sifsuser/transactions/
   ```

   c) Run the following commands to load the data:

   ```
   hdfs dfs -put /opt/IBM/FinancialMkts_SurveillanceInsight_BigDataContent_Base-2.0.3/
   Sample_Data/OffMarketData/Off-MarketData/transactions_2017-02-22.csv /user/sifsuser/
   transactions
   ```

   ```
   hdfs dfs -put /opt/IBM/FinancialMkts_SurveillanceInsight_BigDataContent_Base-2.0.3/
   Sample_Data/OffMarketData/Off-MarketData/marketReference_2017-02-22.csv /user/sifsuser/
   transactions
   ```

   ```
   hdfs dfs -put /opt/IBM/FinancialMkts_SurveillanceInsight_BigDataContent_Base-2.0.3/
   Sample_Data/OffMarketData/Off-MarketData/transactions_2017-02-22.csv /user/sifsuser/
   transactions
   ```

2. Load the spoofing sample data:
   a) Log on to the HDFS node as the sifsuser.
   b) Go to the `/home/sifsuser/data` directory.
   c) Run the following commands to create the directories:

      ```
      hdfs dfs -mkdir /user/sifsuser/trade
      ```

      ```
      hdfs dfs -mkdir /user/sifsuser/quote
      ```

      ```
      hdfs dfs -mkdir /user/sifsuser/order
      ```

      ```
      hdfs dfs -mkdir /user/sifsuser/exection
      ```

   d) Run the following commands:

      ```
      hdfs dfs -put /opt/IBM/FinancialMkts_SurveillanceInsight_BigDataContent_Base-2.0.3/
      Sample_Data/SpoofingData/Trade_2017-04-10.csv /user/sifsuser/trade/
      ```

      ```
      hdfs dfs -put /opt/IBM/FinancialMkts_SurveillanceInsight_BigDataContent_Base-2.0.3/
      Sample_Data/SpoofingData/Quote_2017-04-10.csv /user/sifsuser/quote/
      ```

      ```
      hdfs dfs -put /opt/IBM/FinancialMkts_SurveillanceInsight_BigDataContent_Base-2.0.3/
      Sample_Data/SpoofingData/Order_2017-04-10.csv /user/sifsuser/order/
      ```

      ```
      hdfs dfs -put /opt/IBM/FinancialMkts_SurveillanceInsight_BigDataContent_Base-2.0.3/
      Sample_Data/SpoofingData/Execution_2017-04-10.csv /user/sifsuser/execution/
      ```

      ```
      hdfs dfs -put /opt/IBM/FinancialMkts_SurveillanceInsight_BigDataContent_Base-2.0.3/
      Sample_Data/SpoofingData/Trade_2017-04-12.csv /user/sifsuser/trade/
      ```

      ```
      hdfs dfs -put /opt/IBM/FinancialMkts_SurveillanceInsight_BigDataContent_Base-2.0.3/
      Sample_Data/SpoofingData/Quote_2017-04-12.csv /user/sifsuser/quote/
      ```

      ```
      hdfs dfs -put /opt/IBM/FinancialMkts_SurveillanceInsight_BigDataContent_Base-2.0.3/
      Sample_Data/SpoofingData/Order_2017-04-12.csv /user/sifsuser/order/
      ```

      ```
      hdfs dfs -put /opt/IBM/FinancialMkts_SurveillanceInsight_BigDataContent_Base-2.0.3/
      Sample_Data/SpoofingData/Execution_2017-04-12.csv /user/sifsuser/execution/
      ```

3. Load the front running sample data:
   a) Log on to the HDFS node as the sifsuser.
   b) Run the following commands to create the directories:

      ```
      hdfs dfs -mkdir /user/sifsuser/trade
      ```

      ```
      hdfs dfs -mkdir /user/sifsuser/order
      ```

   c) Run the following commands to load the data:

      ```
      hdfs dfs -put /opt/IBM/FinancialMkts_SurveillanceInsight_BigDataContent_Base-2.0.3/
      Sample_Data/FrontRunning/Order_2018-01-24_FR.csv /user/sifsuser/order/
      ```

      ```
      hdfs dfs -put /opt/IBM/FinancialMkts_SurveillanceInsight_BigDataContent_Base-2.0.3/
      Sample_Data/FrontRunning/trade_summary_2018-01-24.csv /user/sifsuser/trade/
      ```

      ```
      hdfs dfs -put /opt/IBM/FinancialMkts_SurveillanceInsight_BigDataContent_Base-2.0.3/
      Sample_Data/FrontRunning/employees.csv /user/sifsuser/trade/
      ```

4. Load the pump-and-dump sample data:
   a) Log on to the HDFS node as the sifsuser.

b) Run the following commands to create the directories:

```
hdfs dfs -mkdir /user/sifsuser/trade
```

```
hdfs dfs -mkdir /user/sifsuser/quote
```

```
hdfs dfs -mkdir /user/sifsuser/order
```

```
hdfs dfs -mkdir /user/sifsuser/exection
```

c) Go to the `/opt/IBM/FinancialMkts_SurveillanceInsight_BigDataContent_Base-2.0.3/Sample_Data/PumpDumpSolution/PDZDataSet/Pnd_dailydata` directory.

d) Run the following command:

```
sh PnD_Load.sh
```

Chapter 8. Installing IBM Complaints Surveillance Insight for Financial Services

Procedure

1. Log in to the Hadoop master node as the root user.
2. Save the complaints big data package (CNT58EN.tar) to the /root directory.
3. Run the following commands

   ```
   cd /root
   ```

   ```
   tar -xvf CNT58EN.tar
   ```

   ```
   unzip IBM_Complaints_Surveillance_Analytics_2.0.3_Multiplatform_English.zip -d /opt/IBM/
   IBM_Complaints_Surveillance_Analytics_2.0.3_Multiplatform_English
   ```
4. Run the following commands to set the following environment variables:

   ```
   cd /opt/IBM/IBM_Complaints_Surveillance_Analytics_2.0.3_Multiplatform_English/Complaints/
   install
   export BigData_Artifacts_Location=/opt/IBM/
   IBM_Complaints_Surveillance_Analytics_2.0.3_Multiplatform_English/Complaints
   export script_Location=/opt/IBM/
   IBM_Complaints_Surveillance_Analytics_2.0.3_Multiplatform_English/Complaints/install/
   analytics.complaintspipeline
   ```
5. Run the following commands to copy the libraries and sample data and change the ownership to sifsuser.

   ```
   sudo cp -avr $BigData_Artifacts_Location/lib/ComplaintsPipeline-2.0.3-SNAPSHOT.jar /home/
   sifsuser/lib
   sudo cp -avr $BigData_Artifacts_Location/lib/opennlp-tools-1.5.3.jar /home/sifsuser/lib
   sudo cp -avr $BigData_Artifacts_Location/lib/opennlp-maxent-3.0.3.jar /home/sifsuser/lib
   cp -avr $script_Location/data/*.csv /home/sifsuser/
   sudo chown -R sifsuser:sifsuser /home/sifsuser/
   ```
6. Download the sentence model en-sent.bin file from http://opennlp.sourceforge.net/models-1.5/, and save it to the /home/sifsuser/config/model directory.
7. Run the following commands to create the HDFS directories:

   ```
   kinit -kt /etc/security/keytabs/sifsuser.keytab sifsuser@IBM.COM
   ```

   ```
   hdfs dfs -mkdir -p /user/sifsuser/complaints/
   ```

   ```
   hdfs dfs -mkdir -p /user/sifsuser/complaints/raw
   ```

   ```
   hdfs dfs -mkdir -p /user/sifsuser/complaints/output
   ```
8. Run the following command to copy the sample data to HDFS:

   ```
   hdfs dfs -put /home/sifsuser/CFPB2017_Clean.csv /user/sifsuser/complaints/raw
   ```

Installing the complaints machine learning artifacts

Procedure

1. Log on to the Hadoop master node as the root user.
2. Install the on-demand trend detection service:
   a) Copy `/opt/IBM/Complaints/ml.trend-detection` to the `/home/sifsuser` directory.

   ```
   cp -R /opt/IBM/IBM_Complaints_Surveillance_Analytics_2.0.3_Multiplatform_English/
   Complaints/ml.trend-detection /home/sifsuser/
   ```

   b) Run the following command to change the ownership:

   ```
   sudo chown -R sifsuser:sifsuser /home/sifsuser/ml.trend-detection
   ```

   c) Go to the `ml.trend-detection/src` directory.

   ```
   cd /home/sifsuser/ml.trend-detection/src
   ```

   d) Run the following command:

   ```
   /usr/local/bin/python3.5 onDemandTrendDetectionRestAPI.py &
   ```

Updating the Spark properties for the complaints components

Procedure

1. Log on to the Hadoop master node as the sifsuser user.
2. You must build your Watson models for NLC and NLU, and configure the models with the following properties:

   ```
   WatsonNLCComplaintsClassifierREST=https://gateway.watsonplatform.net/natural-language-
   classifier/api/v1/classifiers/<COMPLAINTS_CLASSIFIER_MODEL_ID>/classify
   WatsonNLCCredentials=<nlc_username>:<nlc_password>
   WatsonNLUREST=https://gateway.watsonplatform.net/natural-language-understanding/api/v1/
   analyze?version=2017-02-27
   WatsonNLUModelID=alchemy
   WatsonNLUCredentials=<nlu_username>:<nlu_password>
   ```

   Replace the IP and port number with the HDP master node and Python Keywords Service port. The default is 5005.

   ```
   keywordsServiceUrl=https://<IP>:<PORT>/analytics/models/v1/get_keywords/
   ```

   Replace the IP and port number with the Docker host name for Solr.

   ```
   SolrREST=https://<IP>:8984/solr/complaints/update?commit=true
   ```

   Update model path.

   ```
   ComplaintSentenceModel=/home/sifsuser/config/model/en-sent.bin
   ```

Run the Spark jobs for complaints

Run the following job to run the complaints pipeline.

```
cd /home/sifsuser/lib
```

```
./run_complaints_pipeline.sh
```

To run the complaints trend detection Spark job, see Trend detection.

Chapter 9. Use SLM tags to track licensing

Software License Metric (SLM) tag files provide a standardized capability for a product to report its consumption of license metrics (resources that are related to the use of the software asset). After SLM is enabled in a product, a runtime XML file is generated to self-report its license usage. The SLM tag files are based on the ISO/IEC 19770-4 standard draft for Resource Utilization Measurement.

SLM tag files

The SLM tag files are stored in XML format, and new metric records are appended to the end of the file by cron jobs. For IBM Surveillance Insight for Financial Services, the SLM tag files (`*.slmtag`) are available on the Kubernetes master node in the `/var/ibm/common/slm` directory.

The following is a sample SLM tag for the IBM Electronic Communication Surveillance Analytics component. Licensing is based on the number of active parties for a day (24 hrs). A cron job runs every day at midnight to calculate this value.

```
<SchemaVersion>2.1.1</SchemaVersion>
<SoftwareIdentity>
        <PersistentId>fe953daa1dbc4446905c4b3dd21e8f81</PersistentId>
        <Name>IBM Electronic Communication Surveillance Analytics</Name>
        <InstanceId>/home/sifsuser/</InstanceId>
</SoftwareIdentity>
<Metric logTime="2018-06-26T05:39:58-04:00">
        <Type>USER</Type>
        <SubType>NO_OF_PARTY</SubType>
        <Value>154</Value>
        <Period>
                <StartTime>2018-06-25T05:39:58-04:00</StartTime>
                <EndTime>2018-06-26T05:39:58-04:00</EndTime>
        </Period>
</Metric>
```

The following is a sample SLM tag for the IBM Trade Surveillance Analytics component. Licensing is based on the number of active parties for a day (24 hrs). A cron job runs every day at midnight to calculate this value.

```
<SchemaVersion>2.1.1</SchemaVersion>
<SoftwareIdentity>
        <PersistentId>c6ede63c6002493f82281c89982fcc32</PersistentId>
        <Name>IBM Trade Surveillance Analytics</Name>
        <InstanceId>/home/sifsuser/</InstanceId>
</SoftwareIdentity>
<Metric logTime="2018-06-26T05:39:58-04:00">
        <Type>USER</Type>
        <SubType>NO_OF_PARTY</SubType>
        <Value>154</Value>
        <Period>
                <StartTime>2018-06-25T05:39:58-04:00</StartTime>
                <EndTime>2018-06-26T05:39:58-04:00</EndTime>
        </Period>
</Metric>
```

The following is a sample SLM tag for the IBM Voice Surveillance Analytics component. Licensing is based on the total voice duration that is processed over a month (30 days). A cron job runs every 30 days at midnight to calculate this value.

```
<SchemaVersion>2.1.1</SchemaVersion>
<SoftwareIdentity>
        <PersistentId>a490d40f839049ea881d9aedf8b3d60f </PersistentId>
        <Name>IBM Voice Surveillance Analytics</Name>
        <InstanceId>/home/sifsuser/</InstanceId>
</SoftwareIdentity>
<Metric logTime="2018-06-26T05:39:58-04:00">
        <Type>FEED</Type>
        <SubType>TOTAL_VOICE_SECONDS</SubType>
        <Value>1821</Value>
        <Period>
                <StartTime>2018-06-25T05:39:58-04:00</StartTime>
                <EndTime>2018-06-26T05:39:58-04:00</EndTime>
        </Period>
</Metric>
```

The following is a sample SLM tag for the IBM Complaints Analytics component. Licensing is based on the number of active parties related to a complaint for a month (30 days). A cron job runs every 30 days at midnight to calculate this value.

```
<SchemaVersion>2.1.1</SchemaVersion>
<SoftwareIdentity>
        <PersistentId>0b27eb2a4fdc429b89c36a8ae8d263ac</PersistentId>
        <Name>IBM Complaints Surveillance Analytics</Name>
        <InstanceId>/home/sifsuser/</InstanceId>
</SoftwareIdentity>
<Metric logTime="2018-06-26T05:39:58-04:00">
        <Type>USER</Type>
        <SubType>NO_OF_COMPLAINT_PARTY</SubType>
        <Value>264</Value>
        <Period>
                <StartTime>2018-06-25T05:39:58-04:00</StartTime>
                <EndTime>2018-06-26T05:39:58-04:00</EndTime>
        </Period>
</Metric>
```

Installing and configuring the SLM tag component

Procedure

1. Log on to the Kubernetes master node as the root user.
2. Copy the SLM related folders from the unzipped installation package directory to the `/home/sifsuser` directory.

   ```
   cp -r <extract-location>/IBM_Surveillance_Analytics_2.0.3_Multiplatform_English/
   SIFSSLMTag/* /home/sifsuser/
   ```

3. Go to the `/home/sifsuser/SIFSSLMTag` directory and create a `lib` directory.

   ```
   cd /home/sifsuser/SIFSSLMTag
   ```

   ```
   mkdir lib
   ```

   ```
   mv *jar lib/
   ```

4. Modify the scripts files: `SIFSEcommSlmTag.sh`, `SIFSTradeSlmTag.sh`, `SIFSVoiceSlmTag.sh`, and `SIFSComplaintsSlmTag.sh`.

   a) Replace `/home/sifsuser/slmtag/lib/` with `/home/sifsuser/SIFSSLMTag/lib/`.

   b) Update the "-Djavax.net.ssl.trustStore" value with the appropriate values.

   c) Update the "-Djavax.net.ssl.trustStorePassword" value with the appropriate values.

5. Edit the root user's `.bashrc` file.

   ```
   export JAVA_HOME=<Default Java installed on your system>
   export PATH=${JAVA_HOME}/bin:$PATH
   ```

6. Go to the `/home/sifsuser/SLM_Scripts` and modify the scripts files.

   a) If the IBM Electronic Communication Surveillance Analytics component is installed, open the `SIFSEcommSlmTag.sh` script, and change the JDBC URL hostname to the Kubernetes master node.

   b) If the IBM Trade Surveillance Analytics component is installed, open the `SIFSTradeSlmTag.sh` script, and change the JDBC URL hostname to the Kubernetes master node.
   c) If the IBM Voice Surveillance Analytics component is installed, open the `SIFSVoiceSlmTag.sh` script, and change the Solr URL hostname to the Kubernetes master node.
   d) If the IBM Complaints Surveillance Analytics component is installed, open the `SIFSComplaintsSlmTag.sh` script, and change the JDBC URL hostname to the Kubernetes master node.

7. Create a `sifscron` file.

   ```
   touch /home/sifsuser/sifscron
   ```

8. Edit the `sifscron` file.

   a) For IBM Electronic Communication Surveillance Analytics:

   ```
   '0 0 * * * /home/sifsuser/SLM_Scripts/SIFSEcommSlmTag.sh' >> /home/sifsuser/sifscron
   ```

   b) For IBM Trade Surveillance Analytics:

   ```
   '0 0 * * * /home/sifsuser/SIFSSLMTag/SLM_Scripts/SIFSTradeSlmTag.sh' >> /home/sifsuser/
   sifscron
   ```

   c) For IBM Voice Surveillance Analytics:

   ```
   '0 0 1 * * /home/sifsuser/SIFSSLMTag/SLM_Scripts/SIFSVoiceSlmTag.sh' >> /home/sifsuser/
   sifscron
   ```

   d) For IBM Complaints Surveillance Analytics:

   ```
   '0 0 1 * * /home/sifsuser/SIFSSLMTag/SLM_Scripts SIFSComplaintsSlmTag.sh' >> /home/
   sifsuser/sifscron
   ```

9. Run the following command to add a new cron job:

   ```
   crontab /home/sifsuser/sifscron
   ```

Chapter 10. Reference

The reference topics provide information about IBM Surveillance Insight for Financial Services.

Port reference

IBM Surveillance Insight for Financial Services uses several ports to communicate between components.

The following is a list of ports, by component. In the table, E indicates an external port and I indicates an internal port.

Table 1: Surveillance Insight port reference

Service	Port
MQTT Messaging	6883 (External/Internal)
Case Manager	9443 (E/I)
CEDM	9444 (E/I)
IGA-UI	3061 (E/I)
RMS Design Studio	9445 (E/I)
Kibana	5601 (E/I)
Security-Auth-nodejs	3000 (E/I)
IGA-Reader	9461 (E/I)
SI Liberty	8981 (E/I)
SI DB2	9401 (E/I)
SI Solr	8984 (E/I)
SI ElasticSearch	9201 (I)
SI Logstash	5045 (I)
SI Kibana	5602 (E/I)
SI Filebeat	6060 (E/I)

Update keystores

IBM Surveillance Insight for Financial Services uses a self-signed keystore for all data-at-rest and data-in-motion security. You can replace the default keystore with a certificate authority (CA) approved keystore.

Prerequisites

1. You must get the certificate from the client with the following name and type. If the names are different, then you must rename them to match the following.
   - `fci_universal_ks.crl`
   - `fci_universal_ks.crt.pem`
   - `fci_universal_ks.jks`

- `fci_universal_ks.kdb`
- `fci_universal_ks.p12`
- `fci_universal_ks.pvtk.pem`
- `fci_universal_ks.rdb`
- `fci_universal_ks.sth`

2. Confirm that the certificate files are created for the label as fci_universal_label.
3. Create a copy of the `fci_universal_ks.jks` certificate with the following names:
   - `kafka.client.keystore.jks`
   - `kafka.client.truststore.jks`
4. Copy these files to the `/tmp/cloud_keys` directory on the Kubernetes master node, NFS node, and HDP master node.

   The `/tmp/cloud_keys` directory is a temporary directory. You can use another directory, but commands used in the section use the `/tmp/cloud_keys` directory.

Managing the Kubernetes cluster certificates

Procedure

1. Log on to the Kubernetes master node as the root user.
2. Copy the keys and certificates on to the Kubernetes master node in `$HOME/fci-install-kit/helm`.
3. Go to the `helm` directory.

   ```
   cd $HOME/fci-install-kit/helm
   ```

4. Edit the `$HOME/fci-install-kit/helm/fcco-platform-secrets-files.cfg` script file and uncomment the following lines. Keep the other lines commented.

   ```
   f  ./fci_universal_ks.crt.pem
   f  ./fci_universal_ks.jks
   f  ./fci_universal_ks.kdb
   f  ./fci_universal_ks.pvtk.pem
   f  ./fci_universal_ks.sth
   ```

5. Run the following command:

   ```
   $HOME/fci-install-kit/helm/helm/update-secret -b -f ./fcco-platform-secrets-files.cfg -n
   fcco-platform-secrets-files
   ```

6. Use the following steps for the Audit database pods:

   a) Run the following command:

   ```
   kubectl exec -it datastore_pod bash
   su - db2inst1
   ```

   b) Run the following scripts (using the new encryption key in the last command which must be in clear text format):

   ```
   /home/db2inst1-anchor/db2inst1/maintenance/fci-new-db-kek.sh -l cloud_sl
   ```

   ```
   /home/db2inst1-anchor/db2inst1/maintenance/fci-dump-dbm-ks.sh
   ```

   ```
   /home/db2inst1-anchor/db2inst1/maintenance/fci-set-dbm-kek.sh -l cloud_sl
   ```

   ```
   /home/db2inst1-anchor/db2inst1/maintenance/fci-rekey-dbm-ks.sh -p PLT_dbmKs_Pa55 -n
   your_new_encryption_key_password
   ```

c) Exit from the pod.

7. Get the base64 keystore password for your keystore by using the following command:

```
echo -n 'newpassword' | base64
```

8. Run the following command and modify the value for FCI_JKS_PASSWORD to be the base64-encoded keystore password for your keystore:

```
kubectl edit secret fcco-platform-secrets-env
```

9. Restart all Liberty, MQ, Node.js, Db2, Kibana, and Nginx pods:

```
kubectl delete po -l 'app in(case-manager-fci-messaging,case-manager-fci-solution,cedm-
datastore,cedm-integration,cedm-ui,common-ui-nginx,common-ui-nodejs,logging-kb,security-
audit-app,security-audit-datastore,security-auth-nodejs)'
```

10. Copy the keys and certificates on to the Kubernetes master node in `/fcimedia/sifs-base/sifs-install-kit/helm/`.

11. Go to the `helm` directory.

```
cd /fcimedia/sifs-base/sifs-install-kit/helm/
```

12. Edit the `/fcimedia/sifs-base/sifs-install-kit/helm/fcco-platform-secrets-files.cfg` script file and uncomment the following lines. Keep the other lines commented.

```
f  ./fci_universal_ks.crt.pem
f  ./fci_universal_ks.pvtk.pem
```

13. Run the following commands:

```
cd /fcimedia/sifs-base/sifs-install-kit/helm
```

```
./update-secret -b -f ./fcco-platform-secrets-files.cfg -n fcco-platform-secrets-files
```

14. Modify the keys for data-at-rest in Db2. Use these steps for the SI database.

    a) Connect to database pod as the db2inst1 user.

    ```
    kubectl exec -it <datastore_pod> bash
    sudo su - db2inst1
    ```

    b) Run the following command to generate new encryption keys.

    ```
    /home/db2inst1-anchor/db2inst1/maintenance/fci-new-db-kek.sh -l cloud_sl
    ```

    ```
    /home/db2inst1-anchor/db2inst1/maintenance/fci-dump-dbm-ks.sh
    ```

    ```
    /home/db2inst1-anchor/db2inst1/maintenance/fci-set-dbm-kek.sh -l cloud_sl
    ```

    c) Run the following command to change the default password for the database manager keystore. The default password is PLT_dbmKs_Pa55. When you enter a new password, ensure that it is strong. Special characters other than _ are not recommended.

    ```
    /home/db2inst1-anchor/db2inst1/maintenance/fci-rekey-dbm-ks.sh -p PLT_dbmKs_Pa55 -n
    n0tY0ur0r1g1nalPa55Phrase
    ```

    d) Exit from the pod.

15. Modify the following keys in the configmap.

    a) Run the following command to open the configmap.

    ```
    kubectl edit cm fcco-security-audit
    ```

    b) Modify the com_fci_keystore_password value. Enter the keystore password value in xor encoded format.

c) Run the following command to open the configmap.

```
kubectl edit cm sifs-base
```

d) Modify the SERVER_TRUSTSTORE_PWD and fci_keystore_password values. Enter the keystore password values in xor encoded format.

16. Restart all of the Liberty, Node.JS, Db2, Kibana, and Nginx pods.

```
kubectl delete po -l 'app in(base-liberty,base-datastore)'
```

Managing the HDP cluster certificates

Procedure

1. Log in to the HDP master node as sifsuser.
2. Run the following command to copy the certificate.

```
cp /tmp/cloud_cert/fci_universal_ks.jks /home/sifsuser/security/
```

```
cp /tmp/cloud_cert/fci_universal_ks.crt.pem /home/sifsuser/security/
```

3. Open the `/home/sifsuser/lib/sifs.spark.properties` file, and edit the db2TrustStorePassword value with the new keystore password in xor encoded format.
4. Copy the certificate to all of the HDP cluster nodes by using the following command.

```
scp -r /home/sifsuser/security/ sifsuser@<hdp-cluser-ip>:/home/sifsuser/security/
```

5. Copy the modified properties file to all of the HDP cluster nodes by using the following command.

```
scp /home/sifsuser/lib/sifs.spark.properties sifsuser@<hdp-cluser-ip>:/home/sifsuser/lib/
sifs.spark.properties
```

Managing the NFS certificates

Procedure

1. Log in to the NFS node as the root user.
2. Replace the existing keys in the following locations with the new keys from the `/tmp/cloud_cert` directory:
   - `/fcisi/sifs-liberty-instance/usr/servers/SIFSServer/resources/security/`
   - `/fcisi/sifs-db2-instance/ks/`
   - `/fcisi/sifs-solr-instance/etc/`

Managing the SLM tag certificates

Procedure

1. Log on to the Kubernetes master node as the root user.
2. Run the following command to copy the certificates.

```
cp /tmp/cloud_cert/fci_universal_ks.jks /home/sifsuser/lib/
```

3. Go to the `/home/sifsuser/SLM_Scripts` directory and modify the scripts files as described below.

   a) If the IBM Electronic Communication Surveillance Analytics component is installed, open the `SIFSEcommSlmTag.sh` script, and change the Djavax.net.ssl.trustStorePassword value to the new password.

   b) If the IBM Trade Surveillance Analytics component is installed, open the `SIFSTradeSlmTag.sh` script, and change the Djavax.net.ssl.trustStorePassword value to the new password.

   c) If the IBM Complaints Surveillance Analytics component is installed, open the `SIFSComplaintsSlmTag.sh` script, and change the Djavax.net.ssl.trustStorePassword value to the new password.

Updating the Db2 password

Procedure

1. Use the following steps in the Liberty container.

   a) Get the Db2 password in xor format.

   b) Use the following commands to update the Liberty pod:

   ```
   kubectl edit cm sifs-base
   ```

   ```
   sifs_db2_password: '{xor}Lz4sLCgwLTs='
   ```

   c) Exit the pod.

   d) Restart the pod:

   ```
   kubectl delete pod $(kubectl get pods -l app=sifs-base-liberty -o
   jsonpath='{.items[*].metadata.name}')
   ```

2. Use the following steps on the Hadoop cluster nodes.

Deleting sample data

Use the following steps to permanently delete the sample data.

Procedure

1. Log in to the Kubernetes master node as the root user.
2. Connect to the database pod and log in as the db2inst1 user.

   ```
   kubectl exec -it <datastore_pod> bash
   ```

   ```
   sudo su - db2inst1
   ```

   a) Run the following commands to delete the data:

   ```
   db2 drop database SIFS
   ```

   ```
   sudo sh /tmp/sifs-config-db2.sh
   ```

   ```
   db2start
   ```

   ```
   db2terminate
   ```

3. Log on to the Solr container.

a) Run the following command:

```
curl -k -H "Content-Type: text/xml" --data-binary '<delete><query>*:*</query></delete>'
https://<localhost>:<port>/solr/sifs/update?commit=true
```

Change the *<localhost>* and *<port>* values to the Kubernetes master node IP address or hostname and port number where Apache Solr is running.

4. Log on to the HDP master node as the sifsuser user.

a) Run the following commands:

```
hdfs dfs –rm –r /user/sifsuser/transactions/*
```

```
hdfs dfs –rm –r /user/sifsuser/marketReference/*
```

```
hdfs dfs –rm –r /user/sifsuser/transactions/*
```

```
hdfs dfs –rm –r /user/sifsuser/trade/*
```

```
hdfs dfs –rm –r /user/sifsuser/order/*
```

```
hdfs dfs –rm –r /user/sifsuser/positions/*
```

```
hdfs dfs –rm –r /user/sifsuser/quote/*
```

```
hdfs dfs –rm –r /user/sifsuser/execution/*
```

```
hdfs dfs –rm –r /user/sifsuser/EOD/*
```

```
hdfs dfs –rm –r /user/sifsuser/voice/*
```

```
hdfs dfs –rm –r /user/sifsuser/voice_archive/*
```

```
hdfs dfs –rm –r /user/sifsuser/comm/*
```

```
hdfs dfs –rm –r /user/sifsuser/ discovery-model/*
```

Appendix A. Troubleshooting

This section provides troubleshooting information.

Failed on local exception

During any HDFS operation, you receive the following error:

```
Failed on local exception: java.io.IOException: javax.security.sasl.SaslException: GSS initiate
failed [Caused by   GSSException: No valid credentials provided (Mechanism level: Failed to
find any Kerberos tgt)
```

To resolve this problem, you must identify the principal of the keytab file, and then renew the keytab.

1. Run the following command to identify the principal of the keytab file:

   ```
   klist -k -t <location of keytab file>/<keytab fle name>
   ```

2. Run the following command to renew the keytab:

   ```
   kinit -kt <location of keytab file>/<keytab fle name> <principal>
   ```

   - To renew the keytab file for the sifsuser, run fun the following commands:

     ```
     su - sifsuser
     ```

     ```
     kinit –kt /etc/security/keytabs/sifsuser.keytab sifsuser@IBM.COM
     ```

   - To renew the keytab file for the solruser, run fun the following commands:

     ```
     su - solruser
     ```

     ```
     kinit -kt /etc/security/keytabs/solr.keytab solruser/<ambari hostname>@IBM.COM
     ```

No data on the Health Check tab

This problem can occur if the ES Data volume value in the Helm chart points to an empty Elasticsearch database.

To verify the configuration, you can log in to the NFS server and check the contents of the `/fci-exports/sifs-elasticsearch-instance` directory. If it is empty, use the following steps to correct the problem.

Note: If you are using a custom-NFS mount, then `sifs-elasticsearch-instance` would be on `/fcisi/`. You can replace `/fci-exports` with `/fcisi` in the following steps.

1. Log in to the NFS server as the root user.
2. Get the latest `sifs-data-store-elasticsearch.tar.gz` file from the `root` folder, and copy it to the mount directory on the NFS server.

   ```
   cp /root/sifs-data-store-elasticsearch.tar.gz /fcisi/sifs-elasticsearch-instance/
   ```

3. Log in to the Kubernetes master node as the root user.
4. Use the following command to get the name of the Elasticsearch pod:

   ```
   kubectl get pods
   ```

5. Use the following command to log in to the Elasticsearch pod:

```
kubectl exec -it <pod-name-of-elasticsearch> bash
```

6. Use the following command to get the process ID for Elasticsearch:

```
ps -ef | grep /usr/share/elasticsearch
```

For example:

```
[elasticsearch@sifs-base-elasticsearch]$ ps -ef | grep usr/share/elasticsearch
elastic+     9     1 11 12:58 ?        00:00:39 /usr/lib/jvm/jre-1.8.0-openjdk/bin/java -
Xms1g -Xmx1g   -XX:+UseConcMarkSweepGC -XX:CMSInitiatingOccupancyFraction=75 -
XX:+UseCMSInitiatingOccupancyOnly -XX:+AlwaysPreTouch -Xss1m -Djava.awt.headless=true -
Dfile.encoding=UTF-8 -Djna.nosys=true -XX:-OmitStackTraceInFastThrow -
Dio.netty.noUnsafe=true -Dio.netty.noKeySetOptimization=true -
Dio.netty.recycler.maxCapacityPerThread=0 -Dlog4j.shutdownHookEnabled=false -
Dlog4j2.disable.jmx=true -Djava.io.tmpdir=/tmp/elasticsearch.blpQLfyJ -
XX:+HeapDumpOnOutOfMemoryError -XX:+PrintGCDetails -XX:+PrintGCDateStamps -
XX:+PrintTenuringDistribution -XX:+PrintGCApplicationStoppedTime -Xloggc:logs/gc.log -
XX:+UseGCLogFileRotation -XX:NumberOfGCLogFiles=32 -XX:GCLogFileSize=64m -
Des.cgroups.hierarchy.override=/ -Des.path.home=usr/share/elasticsearch -Des.path.conf=/usr/
share/elasticsearch/config -cp /usr/share/elasticsearch/lib/*
org.elasticsearch.bootstrap.Elasticsearch
elastic+   510   342  0 13:03 pts/0    00:00:00 grep --color=auto usr/share/elasticsearch
```

7. Kill the Elasticsearch process.

```
kill -9 9
```

Where 9 is the process ID in the example.

8. Remove existing Elasticsearch data directory from the pod.

```
rm -rf /usr/share/elasticsearch/data
```

9. Extract the `tar.gz` file.

```
tar -xzf /home/elasticsearch-anchor/sifs-data-store-elasticsearch.tar.gz -C /home/
elasticsearch-anchor/
```

10. Create a symbolic link between `/home/elasticsearch-anchor` and `/usr/share/elasticsearch`.

```
ln -s /home/elasticsearch-anchor/data /usr/share/elasticsearch/data
```

11. Ensure that the data directory is available at `/usr/share/elasticsearch/data`
12. Restart Elasticsearch:

```
cd /opt/ibm/fci/scripts/ nohup ./start.sh &
```

13. Exit the Elasticsearch pod.
14. Open the Surveillance Workbench, and click the **Health Check** tab. Data should appear for ecomm and voice.

CDISI5060E No default Java found

You receive the following message: `CDISI5060E No default Java found`.

To resolve this error, install Java version 1.6 or later and set it as the default version. Then, try the command again.

Update the PATH variable in your `.bashrc` file to point to the JAVA location.

```
export PATH=<location of jre/bin directory>:$PATH
```

CDISI3059W You may be running a firewall which may prevent communication between the cluster hosts

You receive the following error message: `Warning: CDISI3059W You may be running a firewall which may prevent communication between the cluster hosts`

To resolve this, run the following command to stop the firewall service, and try the command again:

```
systemctl stop firewalld
```

For more information, see Firewall configuration guidelines for IBM Streams.

CDISI5070E The perl-XML-Simple software dependency is not installed

You receive the following error message: `Error: CDISI5070E The perl-XML-Simple software dependency is not installed`

To resolve this error, install the following RPMs as the root user:

- `perl-XML-NamespaceSupport-1.11-10.el7.noarch.rpm`
- `perl-XML-SAX-0.99-9.el7.noarch.rpm`
- `perl-XML-SAX-Base-1.08-7.el7.noarch.rpm`
- `perl-XML-Simple-2.20-5.el7.noarch.rpm`

Use the following command to install each RPM:

```
rpm -ivh rpm_name
```

Notices

This information was developed for products and services offered worldwide.

This material may be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service. This document may describe products, services, or features that are not included in the Program or license entitlement that you have purchased.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Software Group
Attention: Licensing
3755 Riverside Dr.
Ottawa, ON
K1V 1B7
Canada

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

IBM Surveillance Insight for Financial Services includes Brat (v 1.3) from the following source and licensed under the following agreement:

- http://weaver.nlplab.org/~brat/releases/brat-v1.3_Crunchy_Frog.tar.gz
- https://creativecommons.org/licenses/by-sa/3.0/legalcode

IBM Surveillance Insight for Financial Services includes spaCy Models (v 2.0.0) from the following source and licensed under the following agreement:

- https://spacy.io/models/en (en_core_web_sm 2.0.0)
- https://creativecommons.org/licenses/by-sa/3.0/legalcode

Trademarks

IBM, the IBM logo and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at " Copyright and trademark information " at www.ibm.com/legal/copytrade.shtml.

The following terms are trademarks or registered trademarks of other companies:

- Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

- Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

IBM®