Upgrade Guide - v1.3-R

Purpose

Steps to upgrade 4Sight from v1.2-R to v1.3-R

Directory Placeholders

Below placeholders are used for directories in this installation guide

- <4sight_server_installation_directory>
- <4sightml_server_installation_directory>

Please update the placeholders wherever it is used.

Note

If you dont have any custom values for these placeholders then you can use the values mentioned in the below table

Placeholder	value
<4sight_server_installation_directory>	/4sight/4sightserver
<4sightml_server_installation_directory>	/4sight/4sightmlserver
<java_installation_directory>	/4sight

Upgrade Java

1. Login as root user
2. Executing the below command to remove old java version

```
rm -rf <java_installation_directory>/jdk8u232-b09
```

3. Download new java version using the below commands (Required internet to download the binary otherwise download in prior and copy to <java_installation_directory> with root permission)

```
cd <java_installation_directory>
curl -L https://github.com/adoptium/temurin11-binaries/releases
/download/jdk-11.0.13%2B8/OpenJDK11U-jdk_x64_linux_hotspot_11.0.13
_8.tar.gz -o <java_installation_directory>/OpenJDK11U-
jdk_x64_linux_hotspot_11.0.13_8.tar.gz
```

4. Extract the downloaded binary using the below command

```
tar -xzvf <java_installation_directory>/OpenJDK11U-
jdk_x64_linux_hotspot_11.0.13_8.tar.gz -C
<java_installation_directory>
```

Upgrade Keycloak

1. Login as root user
2. Stop keycloak

```
ps -ef | grep keycloak
kill -9 <process_id>
```

3. Login as keycloak user

```
su - keycloak
```

4. Replace the existing JAVA_HOME with the below lines in *~/.bashrc* file

```
export JAVA_HOME=<java_installation_directory>/jdk-11.0.13+8
```

5. Source ~/.bashrc

```
source ~/.bashrc
```

6. Take backup of existing Realm and users in keycloak

```
cd $KEYCLOAK_HOME
mkdir -p backup/v1.2-R
sh ./bin/standalone.sh -Dkeycloak.migration.action=export -
Dkeycloak.migration.provider=dir -Dkeycloak.migration.
dir='$KEYCLOAk_HOME/backup/v1.2-R' > keycloak.out &
```

7. Set up keycloak standalone with MARIADB.

```
mysql -u root -p
```

8. Create user and database for Keycloak

```
CREATE USER '<username>'@'localhost' IDENTIFIED BY '<password>';
CREATE DATABASE keycloak;
GRANT ALL PRIVILEGES ON keycloak.* TO '<username>'@'localhost';
FLUSH PRIVILEGES;
```

9. JDBC Setup for keycloak - Download the MySQL connector.

```
mkdir -p $KEYCLOAK_HOME/modules/system/layers/keycloak/org/mariadb
/main
cd $KEYCLOAK_HOME/modules/system/layers/keycloak/org/mariadb/main
wget https://repo1.maven.org/maven2/org/mariadb/jdbc/mariadb-java-
client/2.2.6/mariadb-java-client-2.2.6.jar
```

10. Create a file named module.xml and copy the below code

```
<?xml version="1.0" ?>
<module xmlns="urn:jboss:module:1.3" name="org.mariadb">
<resources>
<resource-root path="mariadb-java-client-2.2.6.jar"/>
</resources>
<dependencies>
<module name="javax.api"/>
<module name="javax.transaction.api"/>
</dependencies>
</module>
```

11. Declare JDBC driver for keycloak

```
cd $KEYCLOAK_HOME/standalone/configuration/
cp /tmp/4sight/Installation_Scripts/deliverables/keycloak/config
/standalone.xml .
```

12. Edit the username and password in standalone.xml for keycloak database. (Edit line numbers 154,155 in standalone.xml and replace this with username and password created for keycloak in step 12)

```
<user-name>username</user-name>
<password>password</password>
```

13. Run the command

```
cd $KEYCLOAK_HOME
sh ./bin/jboss-cli.sh
```

14. Copy the code in command line and press enter

```
module add --name=org.mariadb --resources=/4sight/keycloak-15.0.2
/modules/system/layers/keycloak/org/mariadb/main/mariadb-java-
client-2.2.6.jar --dependencies=javax.api,javax.transaction.api
```

Note:

- To exit from command line press CTRL + C

15. Stop keycloak

```
ps -ef | grep keycloak
kill -9 <process_id>
```

16. Start Keycloak using the below command to import realm and users

```
cd $KEYCLOAK_HOME
sh ./bin/standalone.sh -Dkeycloak.migration.action=import -
Dkeycloak.migration.provider=dir -Dkeycloak.migration.
dir='$KEYCLOAk_HOME/backup/v1.2-R' > keycloak.out &
```

Note:

- To start the keycloak normally use below command

```
cd $KEYCLOAK_HOME
sh ./bin/standalone.sh > keycloak.out &
```

17. Add the below lines in cron
    a. Execute the below command

```
crontab -e
```

    b. Add the below line and update the placeholder value

```
@reboot source $HOME/.bashrc && cd $KEYCLOAK_HOME && sh ./bin
/standalone.sh > keycloak.out &
```

    c. Save and close
18. Exit keycloak user

```
exit
```

Upgrade MariaDB

This section contains upgrade steps to be executed to encrypt the database

MariaDB Encryption

1. Executing the below command to create directory for storing encryption data

```
mkdir -p /etc/mysql/encrypt
cd /etc/mysql/encrypt
```

2. Execute the below command to create a keyfile

```
openssl rand -hex 32 > /etc/mysql/encrypt/keyfile
sed -i '1s/^/1;/' /etc/mysql/encrypt/keyfile
```

3. Create a Password to encrypt a above key and store it in a file

```
echo -n <PASSWORD> > /etc/mysql/encrypt/keyfile.passwd
```

Note: Provide password before executing above command

4. Now encrypt keyfile into keyfile.enc

```
openssl enc -aes-256-cbc -md sha1 -pass file:/etc/mysql/encrypt
/keyfile.passwd -in /etc/mysql/encrypt/keyfile -out /etc/mysql
/encrypt/keyfile.enc
```

5. Now remove the keyfile under encrypt folder

```
rm -f /etc/mysql/encrypt/keyfile
```

6. Change ownership of encryption directory

```
chown mysql:mysql /etc/mysql/encrypt/*
chmod 600 /etc/mysql/encrypt/*
```

7. Add the below lines in /etc/my.cnf file

```
[mariadb]
plugin_load_add             = file_key_management
file_key_management_filename = /etc/mysql/encrypt/keyfile.enc
file_key_management_filekey  = FILE:/etc/mysql/encrypt/keyfile.
passwd
file_key_management_encryption_algorithm = AES_CBC

innodb_encrypt_tables           = ON
innodb_encrypt_temporary_tables  = ON
innodb_encrypt_log              = ON
innodb_encryption_threads       = 4
innodb_encryption_rotate_key_age = 1
encrypt-tmp-disk-tables         = 1
encrypt-tmp-files               = 1
encrypt-binlog                  = 1
aria_encrypt_tables             = ON
```

8. Run the command below to restart the mariadb service

```
systemctl restart mariadb
```

Upgrade Nginx

1. Login as nginx user

```
su nginx
```

2. Replace the following lines in /usr/local/nginx/conf/nginx.conf file under the http block in map $request_uri $csp section

```
~/* "default-src 'self' data: 'unsafe-inline';style-src 'self'
'unsafe-inline'; frame-ancestors 'self';";
```

with the lines given below

```
~/* "default-src 'self'; style-src 'self' 'unsafe-inline'; script-
src 'self' 'unsafe-eval' 'unsafe-inline'; img-src 'self' data:;
connect-src 'self'; frame-ancestors 'self'; object-src 'none'";
'self'; frame-ancestors 'self'; object-src 'none'";
```

3. Exit nginx user

```
exit
```

4. Restart nginx

```
service nginx restart
```

Upgrade 4Sight-Server

This section contains upgrade steps to be executed in 4Sight-Server instance

Copy Release Package

1. Copy the release package ***4sight_v1.3-R.zip*** to the “/tmp“ directory of 4Sight-Server instance
2. Extract using the below commands

```
cd /tmp
unzip 4sight_v1.3-R.zip
```

Note: Provide password if prompted

Upgrade 4Sight-Server

1. Login as 4sight user

```
su - 4sight
```

2. Stop 4Sight-Server

```
ps -ef | grep gunicorn
kill -9 <process_id>
```

3. Take a backup of existing 4Sight-Server

```
cd $HOME
mkdir -p backup/v1.2-R
cd backup/v1.2-R
mv <4sight_server_installation_directory>/4sightserver .
cp ~/4sight.cfg .
mysqldump --host <mariadb_host> --port <mariadb_port> -u root -p
<database_name> > backup.sql
```

Note:

a. <database_name> should be same as configured in ~/4sight.cfg file

4. Upgrade DB schema

```
mysql --host <mariadb_host> --port <mariadb_port> -u root -p
<database_name> < /tmp/4sight/4Sight-Server/database/migrate_data.
sql
```

a. **Note:**

i. <database_name> should be same as configured in ~/4sight.cfg file

5. Remove the below property from the file ~/4sight.cfg

```
EMAIL_REGEX
```

6. Replace the below properties from the file ~/4sight.cfg

```
ELASTICSEARCH_HOST="<Elasticsearch host. Example: http://localhost:
9200>"
MAIL_PORT=<mail port. Example:25>
```

with the lines given below and update the placeholders

```
ELASTICSEARCH_HOST=[<Comma seprated list of elasticsearch hosts.
Examples: "http://host1:port1", "http://host2:port2">]
MAIL_PORT=<mail port. Example:587>
```

7. Extract the binary

```
cd <4sight_server_installation_directory>
mkdir 4sightserver
pip install /tmp/4sight/4Sight-Server/4sight_server-1.3-py3-none-
any.whl --target 4sightserver
```

8. Start 4Sight-Server using the below commands

```
cd <4sight_server_installation_directory>
cd 4sightserver
dos2unix train_models.sh
dos2unix trigger_alerts.sh
nohup gunicorn -b 0.0.0.0:5000 "app:create_app()" >> /dev/null
2>&1 &
```

9. Add the below lines in cron
    a. Execute the below command

```
crontab -e
```

    b. Add the below line and update the placeholder value

```
@reboot cd <4sight_server_installation_directory>
/4sightserver/ &&  nohup gunicorn -b 0.0.0.0:5000 "app:
create_app()" >> /dev/null 2>&1 &
```

    c. Save and close
10. Exit 4sight user

```
exit
```

Upgrade 4Sight-UI

1. Login as root user
2. Stop Nginx

```
service nginx stop
```

3. Login as nginx user

```
su - nginx
```

4. Take a backup of existing 4Sight-UI

```
cd $HOME
mkdir -p backup/v1.2-R
cd backup/v1.2-R
mv /usr/local/nginx/html/4Sight-UI .
```

5. Deploy UI to Nginx html folder

```
cp -R /tmp/4sight/4Sight-UI /usr/local/nginx/html/
```

6. Exit nginx user

```
exit
```

7. Start Nginx

```
service nginx start
```

Upgrade 4Sight-ML

This section contains upgrade steps to be executed in 4Sight ML Server instance

Copy Release Package

1. Copy the release package ***4sight_v1.2.1-R.zip*** to the “/tmp“ directory of 4Sight ML Server instance
2. Extract using the below commands

```
cd /tmp
unzip 4sight_v1.2.1-R.zip
```

Note: Provide password if prompted

Upgrade 4Sight-ML

1. Login as 4sighml user

```
su - 4sightml
```

2. Stop 4Sight-ML

```
ps -ef | grep gunicorn
kill -9 <process_id>
```

3. Take a backup of existing 4Sight-ML server

```
cd $HOME
mkdir -p backup/v1.2-R
cd backup/v1.2-R
mv <4sightml_server_installation_directory>/4sightmlserver .
mysqldump --host <mariadb_host> --port <mariadb_port> -u root -p
<database_name> > backup.sql
```

Note:
   a. <database_name> should be same as configured in ~/4sight_ml.cfg file

4. Upgrade DB schema

```
mysql --host <mariadb_host> --port <mariadb_port> -u root -p
<database_name> < /tmp/4sight/4Sight-ML/database/migrate_data.sql
```

   a. **Note:**
      i. <database_name> should be same as configured in ~/4sight_ml.cfg file

5. Replace the below properties from the file ~/4sight_ml.cfg

```
ELASTICSEARCH_HOST="<Elasticsearch host. Example: http://localhost:
9200>"
```

with the lines given below and update the placeholders

```
ELASTICSEARCH_HOST=[<Comma seprated list of elasticsearch hosts.
Examples: "http://host1:port1", "http://host2:port2">]
```

6. Extract the binary

```
cd <4sightml_server_installation_directory>
mkdir 4sightmlserver
pip install /tmp/4sight/4Sight-ML/4sight_ml-1.3-py3-none-any.whl --
target 4sightmlserver
```

7. Start 4Sight-ML using the below commands

```
cd <4sightml_server_installation_directory>
cd 4sightmlserver
dos2unix run_tox.sh
nohup gunicorn -b 0.0.0.0:6000 "app:create_app()" >> /dev/null
2>&1 &
```

8. Add the below lines in cron
    a. Execute the below command

```
crontab -e
```

    b. Add the below lines and update the placeholder value

```
@reboot cd <4sightml_server_installation_directory>
/4sightmlserver/ &&  nohup gunicorn -b 0.0.0.0:6000 "app:
create_app()" >> /dev/null 2>&1 &
```

    c. Save and close
9. Exit 4sightml user

```
exit
```