# Manual Installation Guide - v1.3-R



## Purpose

This document gives the details on how to install 4Sight on a *Centos 7/RHEL 8.4* distribution.

## Software Used

The below softwares are used in 4Sight

| Software | Version |
|---|---|
| Java | jdk-11.0.13+8 |
| MariaDB | 10.4 |
| Nginx | 1.20.1 |
| Python | 3.6.12 |
| Gunicorn | 20.0.4 |

## OS Requirements

Supported Operating system

- Centos 7
- RHEL 8.4

## Directory Placeholders

Below placeholders are used for directories in this installation guide

- <installation_directory>
- <java_installation_directory>
- <mariadb_password>
- <keycloak_installation_directory>
- <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 |
|---|---|
| <installation_directory> | /4sight |
| <java_installation_directory> | /4sight |
| <keycloak_installation_directory> | /4sight |
| <4sight_server_installation_directory> | /4sight/4sightserver |
| <4sightml_server_installation_directory> | /4sight/4sightmlserver |

## 4Sight Server

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

### Install dos2unix

Dos2unix has to be installed for the shell scripts available in the release package. If it is already installed skip this step

1. Login as root user
2. Execute the below command

```
yum install dos2unix
```

### Install unzip

Unzip has to be installed to extract release package. If it is already installed skip this step

1. Login as root user
2. Execute the below command

```
yum install unzip
```

### 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

### Create Installation Directory

1. Login as root user
2. Executing the below command to create installation directory

```
mkdir -p <installation_directory>
```

### Install Java

1. Login as root user
2. Executing the below command to create java installation directory

```
mkdir -p <java_installation_directory>
```

3. Download java 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>
```

### Install MariaDB

1. Login as root user
2. Update the below line in the file /etc/yum.repos.d/MariaDB.repo
    a. For Centos 7, update the below lines

```
# MariaDB 10.4 CentOS repository list - created 2020-05-28 06:
19 UTC
# http://downloads.mariadb.org/mariadb/repositories/
[mariadb]
name = MariaDB
baseurl = http://yum.mariadb.org/10.4/centos7-amd64
gpgkey=https://yum.mariadb.org/RPM-GPG-KEY-MariaDB
gpgcheck=1
```

b. For RHEL 8.4, update the below lines

```
# MariaDB 10.4 RedHat repository list - created  UTC
# http://downloads.mariadb.org/mariadb/repositories/
[mariadb]
name = MariaDB
baseurl = https://yum.mariadb.org/10.4/rhel8-amd64
module_hotfixes=1
gpgkey=https://yum.mariadb.org/RPM-GPG-KEY-MariaDB
gpgcheck=1
```

3. Execute the below commands to install MariaDB

```
yum update
yum -y install MariaDB-server MariaDB-client
```

4. Run the command below to start the mariadb service

```
systemctl start mariadb
systemctl enable mariadb
```

5. Execute the below commands to setup root password for MariaDB

```
mysql -u root
GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost' IDENTIFIED BY
'<mariadb_password>';
FLUSH PRIVILEGES;
exit
```

**MariaDB Master-Master Setup**

**Note:** This section is applicable only if you want to configure Master-Master setup for MariaDB. If not, please ignore this section

## Configure MariaDB Master-Master replica Setup-1

Execute the below steps for configuring first MariaDB server

1. Login as root user
2. Replace the value of 'SELINUX' from **enforcing** to **disabled** in file /etc/selinux/config for disabling SE Linux and relogin to the instance
3. Run the below command to find libgalera_smm.so file path and update the value in wsrep_provider in point 3

```
find / -name libgalera_smm.so
```

4. Add these below line in the file /etc/my.cnf.d/server.cnf

```
[galera]
wsrep_on=ON
wsrep_provider=</usr/lib64/galera-4/libgalera_smm.so>
wsrep_cluster_address="gcomm://<ip1>,<ip2>"
binlog_format=row
default_storage_engine=InnoDB
innodb_autoinc_lock_mode=2
wsrep_cluster_name="MDBCLUSTER"
wsrep_sst_method=rsync
wsrep_node_address="<ip1>"
wsrep_node_name="MDB01"
bind-address="<ip1>"

[mysqld]
log_error=/var/log/mariadb/mariadb.log
```

   **Note:** Update the placeholders in the above block. <ip1> is private IP and <ip2> is private IP of the instances. Please provide port access and privileges to the MariaDB server, if MariaDB is installed in remote server
5. Create Log file

```
mkdir -p /var/log/mariadb
touch /var/log/mariadb/mariadb.log
```

6. Stop the MariaDB

```
systemctl stop mariadb
```

7. Execute the below commands to setup cluster

```
galera_new_cluster
```

8. Start the MariaDB

```
systemctl start mariadb
```

### Configure MariaDB Master-Master replica Setup-2

Execute the below steps for configuring second MariaDB server

1. Login as root user
2. Replace the value of 'SELINUX' from **enforcing** to **disabled** in file /etc/selinux/config for disabling SE Linux and relogin to the instance
3. Run the below command to find libgalera_smm.so file path and update the value in wsrep_provider in point 3

```
find / -name libgalera_smm.so
```

4. Add these below line in the file /etc/my.cnf.d/server.cnf

```
[galera]
wsrep_on=ON
wsrep_provider=</usr/lib64/galera-4/libgalera_smm.so>
wsrep_cluster_address="gcomm://<ip1>,<ip2>"
binlog_format=row
default_storage_engine=InnoDB
innodb_autoinc_lock_mode=2
wsrep_cluster_name="MDBCLUSTER"
wsrep_sst_method=rsync
wsrep_node_address="<ip2>"
wsrep_node_name="MDB02"
bind-address="<ip2>"

[mysqld]
log_error=/var/log/mariadb/mariadb.log
```

   **Note:** Update the placeholders in the above block. <ip1> is private IP and <ip2> is private IP of the instances. Please provide port access and privileges to the MariaDB server, if MariaDB is installed in remote server
5. Create Log file

```
mkdir -p /var/log/mariadb
touch /var/log/mariadb/mariadb.log
```

6. Stop the MariaDB

```
systemctl stop mariadb
```

7. Start the MariaDB

```
systemctl start mariadb
```

**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
```

**Note:** Based on configuration MariaDB master-master setup run the below commands if no configuration done for Mariadb master master setup please ignore commands below.

a. For Configuration of Mariadb master master setup 1. Before executing the given command please check the file "**/var/lib/mysql/grastate.dat**" change **safe_to_bootstap** to 1 if it is 0.

```
systemctl stop mariadb
galera_new_cluster
systemctl start mariadb
```

b. For Configuration of Mariadb master master setup 2

```
systemctl stop mariadb
systemctl start mariadb
```

## Install Python

**Install Python**

1. Login as root user
2. Install dependencies

```
yum -y install autoconf
yum -y install automake
yum -y install binutils
yum -y install bison
yum -y install flex
yum -y install gcc
yum -y install gcc-c++
yum -y install gettext
yum -y install libtool
yum -y install make
yum -y install patch
yum -y install pkgconfig
yum -y install redhat-rpm-config
yum -y install rpm-build
yum -y install rpm-sign
yum -y install zlib-devel
yum -y install openssl-devel.x86_64 openssl.x86_64
yum -y install epel-release
yum -y install mariadb-devel
yum -y install libsqlite3x-devel.x86_64
yum -y install bzip2-devel
```

3. Download Python using the below commands (Required internet to download the binary otherwise download in prior and copy to the directory “/tmp“ with root permission)

```
curl -L https://www.python.org/ftp/python/3.6.12/Python-3.6.12.tar.
xz -o /tmp/Python-3.6.12.tar.xz
```

4. Extract and Install Python using the below commands

```
cd /tmp
tar -xJf Python-3.6.12.tar.xz
cd Python-3.6.12
./configure
make
make install
update-alternatives --install /usr/bin/python python /usr/local/bin
/python3.6 1
update-alternatives --install /usr/bin/pip pip /usr/local/bin/pip3.
6 1
```

5. Execute the below given command to check the current version of Python being used. Press enter to keep the current selection[+], or type selection number.

```
update-alternatives --config python
update-alternatives --config pip
```

6. Edit Python dependency for Yum

   **Note**: Step 6 is applicable only for CentOS operating system. Please ignore this step for other operating system.
   a. Open the file */usr/bin/yum*
   b. Replace the first line "#!*/usr/bin/python*" to "#!*/usr/bin/python2*"
   c. Save and Continue
   d. Open the file */usr/libexec/urlgrabber-ext-down*
   e. Replace the first line "#!*/usr/bin/python*" to "#!*/usr/bin/python2*"
   f. Save and Continue
7. Upgrade pip by executing below command

```
pip install --upgrade pip
```

**Install Gunicorn and Wheel**

Gunicorn will be used as the Web Server. Wheel will to be used for binary. To install Gunicorn and wheel, please execute the below steps,

1. Login as root user
2. Execute the below commands,

```
pip install gunicorn==20.0.4
pip install wheel
pip install --upgrade pip wheel setuptools
pip install tqdm
pip install --user --upgrade twine
ln -fs /usr/local/bin/gunicorn /usr/bin/gunicorn
```

## Install Nginx

1. Login as root user
2. Install dependencies

```
yum -y install wget
yum -y install git
yum -y install gcc make automake autoconf libtool
yum -y install pcre pcre-devel libxml2 libxml2-devel curl curl-
devel httpd-devel
yum -y install libpcre3 libpcre3-dev
yum -y install yajl
yum -y install yajl-devel
yum -y install gcc gcc-c++ make zlib-devel pcre-devel openssl-devel
yum -y install unzip
```

3. Download nginx and openSSL to /tmp

```
cd /tmp
wget http://nginx.org/download/nginx-1.20.1.tar.gz
wget https://ftp.openssl.org/source/old/1.0.2/openssl-1.0.2.tar.gz
```

4. Install Nginx and ModSecurity

```
cd /usr/src/
cp /tmp/nginx-1.20.1.tar.gz .
cp /tmp/openssl-1.0.2.tar.gz .
tar -xzvf openssl-1.0.2.tar.gz
cd openssl-1.0.2
./config
make
cd /usr/src
git clone --depth 1 https://github.com/SpiderLabs/ModSecurity-
nginx.git
git clone --depth 1 -b v3/master --single-branch https://github.com
/SpiderLabs/ModSecurity
mv ModSecurity/ modsecurity
cd modsecurity
git submodule init
git submodule update
./build.sh
./configure
make
make install
cd /usr/src
tar -xzvf nginx-1.20.1.tar.gz
cd nginx-1.20.1
./configure --with-compat --add-dynamic-module=../ModSecurity-
nginx --with-debug --with-http_ssl_module --with-openssl=..
/openssl-1.0.2
make modules
make install
```

5. Install Headers-more

```
cd /usr/src
git clone https://github.com/openresty/headers-more-nginx-module.
git
cd nginx-1.20.1
./configure --with-compat --add-dynamic-module=/usr/src/headers-
more-nginx-module
make modules
cp /usr/src/nginx-1.20.1/objs/ngx_http_headers_more_filter_module.
so /usr/local/nginx/modules/
```

6. Configure Nginx as a service

```
ln -s /usr/local/nginx/sbin/nginx /bin/nginx
cp /tmp/4sight/Installation_Scripts/deliverables/nginx/nginx.
service /usr/lib/systemd/system
systemctl daemon-reload
systemctl enable nginx.service
chkconfig nginx on
```

7. Add the below lines in /usr/local/nginx/conf/nginx.conf file in start of the conf file like given below

```
load_module /usr/local/nginx/modules
/ngx_http_headers_more_filter_module.so;
#user  nobody;
```

8. Add the below line under the line `worker_processes  1;` in /usr/local/nginx/conf/nginx.conf file

```
user  nginx;
```

9. Add the below lines in /usr/local/nginx/conf/nginx.conf file under the http block

```
server_tokens off;
more_clear_headers Server;
proxy_read_timeout 300s;

map $request_uri $xfo {
    ~/4sight/view/* "SAMEORIGIN";
    #~/4sight/view/* "ALLOW FROM <domain>";
    ~/4sight/menu/* "SAMEORIGIN";
    #~/4sight/menu/* "ALLOW FROM <domain>";
    ~/4sight/api/* "SAMEORIGIN";
    #~/4sight/api/* "ALLOW FROM <domain>";
    ~/* "DENY";
}

map $request_uri $csp {
    ~/4sight/view/* "default-src 'self' data:;style-src 'self'
'unsafe-inline'; frame-ancestors 'self';";
    #~/4sight/view/* "default-src '<domain>' data:;style-src
'<domain>' 'unsafe-inline'; frame-ancestors '<domain>';";
    ~/4sight/menu/* "default-src 'self' data:;style-src 'self'
'unsafe-inline'; frame-ancestors 'self';";
    #~/4sight/menu/* "default-src '<domain>' data:;style-src
'<domain>' 'unsafe-inline'; frame-ancestors '<domain>';";
    ~/4sight/api/* "default-src 'self' data:; frame-ancestors
'self';";
    #~/4sight/api/* "default-src '<domain>' data:; frame-ancestors
'<domain>';";
    ~/* "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'";
}
```

10. Add the below lines in /usr/local/nginx/conf/nginx.conf file under 80 server block

```
add_header X-Frame-Options $xfo;
add_header Content-Security-Policy $csp;
add_header Strict-Transport-Security max-age=31536000;
add_header X-Content-Type-Options nosniff;
```

11. Replace the line `listen  80;` with `listen 80 default_server;`
12. If SSL is required, then add the below lines in /usr/local/nginx/conf/nginx.conf file under http block and update the placeholder values

```
server {
        listen       443 ssl;
        server_name  <server_name>;

        add_header X-Frame-Options $xfo;
        add_header Content-Security-Policy $csp;
        add_header Strict-Transport-Security max-age=31536000;
        add_header X-Content-Type-Options nosniff;

        ssl_protocols TLSv1.2 TLSv1.3;

        ssl_certificate     <Path to ssl_certificate>;
        ssl_certificate_key <Path to ssl_certificate_key>;

        ssl_ciphers ECDHE-RSA-AES256-GCM-SHA512:DHE-RSA-AES256-GCM-
SHA512:ECDHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES256-GCM-SHA384:ECDHE-
RSA-AES256-SHA384;
        ssl_prefer_server_ciphers  on;
        #charset koi8-r;

        #access_log  logs/host.access.log  main;
        error_page   500 502 503 504  /50x.html;
        location = /50x.html {
            root   html;
        }

}
```

13. Create user for Nginx

```
useradd nginx
```

14. Change ownership of Nginx directory

```
chown -R nginx:nginx /usr/local/nginx
```

15. Start Nginx using the below command

```
service nginx start
```

**Note:**

- To Stop Nginx execute the below command

```
service nginx stop
```

Install Keycloak

1. Login as root user
2. Executing the below command to create keycloak installation directory

```
mkdir -p <keycloak_installation_directory>
```

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

```
cd <keycloak_installation_directory>
curl -L https://github.com/keycloak/keycloak/releases/download/15.0.2/keycloak-15.0.2.tar.gz -o <keycloak_installation_directory>/keycloak-15.0.2.tar.gz
```

4. Extract the downloaded binary using the below command

```
tar -xzvf keycloak-15.0.2.tar.gz -C <keycloak_installation_directory>
```

5. Create user for Keycloak

```
useradd keycloak
```

6. Change ownership of Keycloak directory

```
chown -R keycloak:keycloak <keycloak_installation_directory>/keycloak-15.0.2
```

7. Login as keycloak user

```
su - keycloak
```

8. Add the below lines in *~/.bashrc* file

```
export JAVA_HOME=<java_installation_directory>/jdk-11.0.13+8
export PATH=$PATH:$JAVA_HOME/bin
export KEYCLOAK_HOME=<keycloak_installation_directory>/keycloak-
15.0.2
```

9. Source ~/.bashrc

```
source ~/.bashrc
```

10. Configure themes

```
cd $KEYCLOAK_HOME/themes
cp -R /tmp/4sight/keycloak/4sight .
```

11. Set up keycloak standalone with MARIADB.

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

12. Create user and database for Keycloak
    a. For Standalone MariaDB, execute the below commands,

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

    b. For HA MariaDB setup, execute the below commands,

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

13. 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
```

14. 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>
```

15. Declare JDBC driver for keycloak

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

16. 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>
```

17. For HA MariaDB setup, edit the connection url in 151 line in standalone.xml and change to the MariaDB ip address.
18. Run the command

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

19. 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 type exit and enter

20. Start Keycloak using the below command

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

21. 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
22. Exit keycloak user

```
exit
```

23. Login as nginx user

```
su nginx
```

24. Add the below lines in /usr/local/nginx/conf/nginx.conf file under both 80 and ssl server block

```
location /auth {
  proxy_set_header X-Forwarded-For $proxy_protocol_addr;
  proxy_set_header X-Forwarded-Proto $scheme;
  proxy_set_header Host $host;
  proxy_pass http://localhost:8080;
}
```

25. Exit nginx user

```
exit
```

26. Reload nginx

```
service nginx restart
```

## Install 4Sight-Server

1. Login as root user
2. Install MariaDB libs using the below commands

```
yum update
yum -y install MariaDB-shared MariaDB-client
```

3. Create <4sight_server_installation_directory>

```
mkdir -p <4sight_server_installation_directory>
```

4. Create user for 4Sight-Server

```
useradd 4sight
```

5. Change ownership of <4sight_server_installation_directory>

```
chown -R 4sight:4sight <4sight_server_installation_directory>
```

6. Login as 4sight user

```
su - 4sight
```

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. Copy the property file

```
cd
cp /tmp/4sight/4Sight-Server/properties/4sight.cfg .
```

9. Update the placeholder values in ~/4sight.cfg
10. Create empty database

```
mysql --host <mariadb_host> --port <mariadb_port> -u root -p -e "CREATE DATABASE <database_name>;"
```

**Note:**

a. Update the placeholder in the commands before executing
b. <database_name> should be same as configured in ~/4sight.cfg file
c. Please provide port access and privileges to the MariaDB server, if MariaDB is installed in remote server
d. Provide password when prompted

11. Import Skeletal dump

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

**Note:**

a. Update the placeholder in the commands before executing
b. <database_name> should be same as configured in ~/4sight.cfg file
c. Please provide port access and privileges to the MariaDB server, if MariaDB is installed in remote server
d. Provide password when prompted

12. 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 &
```

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

```
crontab -e
```

b. Add the below line and update the placeholder value

```
0 * * * * cd <4sight_server_installation_directory>
/4sightserver/ && sh train_models.sh &
* * * * * cd <4sight_server_installation_directory>
/4sightserver/ && sh trigger_alerts.sh &
@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

14. Exit 4sight user

```
exit
```

15. Login as nginx user

```
su nginx
```

16. Add the below lines in /usr/local/nginx/conf/nginx.conf file under both 80 and ssl server block

```
location /4sight/api {
  root html;
  index index.html index.htm;
  proxy_pass http://localhost:5000;
  proxy_force_ranges on;
}
```

17. Exit nginx user

```
exit
```

18. Login as root user
19. Reload Nginx

```
service nginx restart
```

### Install 4Sight-UI

1. Login as nginx user

```
su nginx
```

2. Deploy UI to Nginx html folder

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

3. Replace the “location /” block with the lines in /usr/local/nginx/conf/nginx.conf file under both 80 and ssl server block

```
location / {
  root /usr/local/nginx/html/4Sight-UI;
  index index.html;
  try_files $uri /index.html;
}
```

4. Exit nginx user

```
exit
```

5. Login as root user
6. Reload Nginx

```
service nginx restart
```

## 4Sight ML Server

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

### Install dos2unix

Dos2unix has to be installed for the shell scripts available in the release package. If it is already installed skip this step

1. Login as root user
2. Execute the below command

```
yum install dos2unix
```

## Install unzip

Unzip has to be installed to extract release package. If it is already installed skip this step

1. Login as root user
2. Execute the below command

```
yum install unzip
```

## Copy Release Package

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

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

**Note:** Provide password if prompted

## Create Installation Directory

1. Login as root user
2. Executing the below command to create installation directory

```
mkdir -p <installation_directory>
```

## Install Python

#### Install Python

1. Login as root user
2. Install dependencies

```
yum -y install autoconf
yum -y install automake
yum -y install binutils
yum -y install bison
yum -y install flex
yum -y install gcc
yum -y install gcc-c++
yum -y install gettext
yum -y install libtool
yum -y install make
yum -y install patch
yum -y install pkgconfig
yum -y install redhat-rpm-config
yum -y install rpm-build
yum -y install rpm-sign
yum -y install zlib-devel
yum -y install openssl-devel.x86_64 openssl.x86_64
yum -y install epel-release
yum -y install mariadb-devel
yum -y install libsqlite3x-devel.x86_64
yum -y install bzip2-devel
```

3. Download Python using the below commands (Required internet to download the binary otherwise download in prior and copy to the directory “/tmp“ with root permission)

```
curl -L https://www.python.org/ftp/python/3.6.12/Python-3.6.12.tar.
xz -o /tmp/Python-3.6.12.tar.xz
```

4. Extract and Install Python using the below commands

```
cd /tmp
tar -xJf Python-3.6.12.tar.xz
cd Python-3.6.12
./configure
make
make install
update-alternatives --install /usr/bin/python python /usr/local/bin
/python3.6 1
update-alternatives --install /usr/bin/pip pip /usr/local/bin/pip3.
6 1
```

5. Execute the below given command to check the current version of Python being used. Press enter to keep the current selection[+], or type selection number.

```
update-alternatives --config python
update-alternatives --config pip
```

6. Edit Python dependency for Yum

   **Note**: Step 6 is applicable only for CentOS operating system. Please ignore this step for other operating system.
   a. Open the file */usr/bin/yum*
   b. Replace the first line "#!*/usr/bin/python*" to "#!*/usr/bin/python2*"
   c. Save and Continue
   d. Open the file */usr/libexec/urlgrabber-ext-down*
   e. Replace the first line "#!*/usr/bin/python*" to "#!*/usr/bin/python2*"
   f. Save and Continue
7. Upgrade pip by executing below command

```
pip install --upgrade pip
```

**Install Gunicorn and Wheel**

Gunicorn will be used as the Web Server. Wheel will to be used for binary. To install Gunicorn and wheel, please execute the below steps,

1. Login as root user
2. Execute the below commands,

```
pip install gunicorn==20.0.4
pip install wheel
pip install --upgrade pip wheel setuptools
pip install tqdm
pip install --user --upgrade twine
ln -fs /usr/local/bin/gunicorn /usr/bin/gunicorn
```

## Install Nginx

1. Login as root user
2. Install dependencies

```
yum -y install wget
yum -y install git
yum -y install gcc make automake autoconf libtool
yum -y install pcre pcre-devel libxml2 libxml2-devel curl curl-
devel httpd-devel
yum -y install libpcre3 libpcre3-dev
yum -y install yajl
yum -y install yajl-devel
yum -y install gcc gcc-c++ make zlib-devel pcre-devel openssl-devel
yum -y install unzip
```

3. Download nginx and openSSL to /tmp

```
cd /tmp
wget http://nginx.org/download/nginx-1.20.1.tar.gz
wget https://ftp.openssl.org/source/old/1.0.2/openssl-1.0.2.tar.gz
```

4. Install Nginx and ModSecurity

```
cd /usr/src/
cp /tmp/nginx-1.20.1.tar.gz .
cp /tmp/openssl-1.0.2.tar.gz .
tar -xzvf openssl-1.0.2.tar.gz
cd openssl-1.0.2
./config
make
cd /usr/src
git clone --depth 1 https://github.com/SpiderLabs/ModSecurity-
nginx.git
git clone --depth 1 -b v3/master --single-branch https://github.com
/SpiderLabs/ModSecurity
mv ModSecurity/ modsecurity
cd modsecurity
git submodule init
git submodule update
./build.sh
./configure
make
make install
cd /usr/src
tar -xzvf nginx-1.20.1.tar.gz
cd nginx-1.20.1
./configure --with-compat --add-dynamic-module=../ModSecurity-
nginx --with-debug --with-http_ssl_module --with-openssl=..
/openssl-1.0.2
make modules
make install
```

5. Install Headers-more

```
cd /usr/src
git clone https://github.com/openresty/headers-more-nginx-module.
git
cd nginx-1.20.1
./configure --with-compat --add-dynamic-module=/usr/src/headers-
more-nginx-module
make modules
cp /usr/src/nginx-1.20.1/objs/ngx_http_headers_more_filter_module.
so /usr/local/nginx/modules/
```

6. Configure Nginx as a service

```
ln -s /usr/local/nginx/sbin/nginx /bin/nginx
cp /tmp/4sight/Installation_Scripts/deliverables/nginx/nginx.
service /usr/lib/systemd/system
systemctl daemon-reload
systemctl enable nginx.service
chkconfig nginx on
```

7. Add the below lines in /usr/local/nginx/conf/nginx.conf file in start of the conf file

```
load_module /usr/local/nginx/modules
/ngx_http_headers_more_filter_module.so;
#user  nobody;
```

8. Add the below line under the line `worker_processes  1;` in /usr/local/nginx/conf/nginx.conf file

```
user  nginx;
```

9. Add the below lines in /usr/local/nginx/conf/nginx.conf file under the http block

```
server_tokens off;
more_clear_headers Server;
proxy_read_timeout 300s;

server_tokens off;
more_clear_headers Server;

map $request_uri $xfo {
    ~/4sightml/* "SAMEORIGIN";
    #~/4sightml/* "ALLOW FROM <domain>";
    ~/* "DENY";
}

map $request_uri $csp {
    ~/4sightml/* "default-src 'self' data:; frame-ancestors
'self';";
    #~/4sightml/* "default-src '<domain>' data:; frame-ancestors
'<domain>';";
    #~/* "default-src 'self' data:;style-src 'self' 'unsafe-
inline'; frame-ancestors 'self';";
    ~/* "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'";
```

10. Add the below lines in /usr/local/nginx/conf/nginx.conf file under 80 server block

```
add_header X-Frame-Options $xfo;
add_header Content-Security-Policy $csp;
add_header Strict-Transport-Security max-age=31536000;
add_header X-Content-Type-Options nosniff;
```

11. Replace the line `listen  80;` with `listen 80 default_server;`
12. If SSL is required, then add the below lines in /usr/local/nginx/conf/nginx.conf file under http block and update the placeholder values

```
server {
        listen       443 ssl;
        server_name  <server_name>;

        add_header X-Frame-Options $xfo;
        add_header Content-Security-Policy $csp;
        add_header Strict-Transport-Security max-age=31536000;
        add_header X-Content-Type-Options nosniff;

        ssl_protocols TLSv1.2 TLSv1.3;

        ssl_certificate     <Path to ssl_certificate>;
        ssl_certificate_key <Path to ssl_certificate_key>;

        ssl_ciphers ECDHE-RSA-AES256-GCM-SHA512:DHE-RSA-AES256-GCM-
SHA512:ECDHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES256-GCM-SHA384:ECDHE-
RSA-AES256-SHA384;
        ssl_prefer_server_ciphers  on;
        #charset koi8-r;

        #access_log  logs/host.access.log  main;

        error_page   500 502 503 504  /50x.html;
        location = /50x.html {
            root   html;
        }

}
```

13. Create user for Nginx

```
useradd nginx
```

14. Change ownership of Nginx directory

```
chown -R nginx:nginx /usr/local/nginx
```

15. Start Nginx using the below command

```
service nginx start
```

**Note:**

- To Stop Nginx, execute the below command

```
service nginx stop
```

Install 4Sight-ML

1. Login as root user
2. Install MariaDB libs
    a. Update the below line in the file /etc/yum.repos.d/MariaDB.repo
        i. For Centos 7, update the below lines

```
# MariaDB 10.4 CentOS repository list - created 2020-05-
28 06:19 UTC
# http://downloads.mariadb.org/mariadb/repositories/
[mariadb]
name = MariaDB
baseurl = http://yum.mariadb.org/10.4/centos7-amd64
gpgkey=https://yum.mariadb.org/RPM-GPG-KEY-MariaDB
gpgcheck=1
```

        ii. For RHEL 8.4, update the below lines

```
# MariaDB 10.4 RedHat repository list - created  UTC
# http://downloads.mariadb.org/mariadb/repositories/
[mariadb]
name = MariaDB
baseurl = https://yum.mariadb.org/10.4/rhel8-amd64
module_hotfixes=1
gpgkey=https://yum.mariadb.org/RPM-GPG-KEY-MariaDB
gpgcheck=1
```

    b. Execute the below commands

```
yum update
yum -y install MariaDB-shared MariaDB-client
```

3. Create <4sightml_server_installation_directory>

```
mkdir -p <4sightml_server_installation_directory>
```

4. Create user for 4Sight-ML

```
useradd 4sightml
```

5. Change ownership of <4sight_server_installation_directory>

```
chown -R 4sightml:4sightml <4sightml_server_installation_directory>
```

6. Login as 4sightml user

```
su - 4sightml
```

7. Install tox

```
pip install tox
```

8. Add the below lines in *~/.bashrc* file

```
export PATH=$PATH:$HOME/.local/bin
```

9. Source ~/.bashrc

```
source ~/.bashrc
```

10. 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
```

11. Copy the property file

```
cd
cp /tmp/4sight/4Sight-ML/properties/4sight_ml.cfg .
```

12. Update the placeholder values in ~/4sight_ml.cfg
13. Create Empty Database by executing the below command

```
mysql --host <mariadb_host> --port <mariadb_port> -u root -p -e
"CREATE DATABASE <database_name>;"
```

**Note:**

a. Update the placeholder in the commands before executing
b. <database_name> should be same as configured in ~/4sight_ml.cfg file
c. Please provide port access and privileges to the MariaDB server, if MariaDB is installed in remote server
d. Provide password when prompted

14. Import skeletal dump

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

**Note:**

a. Update the placeholder in the commands before executing
b. <database_name> should be same as configured in ~/4sight_ml.cfg file
c. Please provide port access and privileges to the MariaDB server, if MariaDB is installed in remote server
d. Provide password when prompted

15. 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 &
```

16. Create directory for tox logs

```
mkdir -p $HOME/tox_logs
```

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

```
crontab -e
```

    b. Add the below lines and update the placeholder value

```
* * * * * cd <4sightml_server_installation_directory>
/4sightmlserver && sh run_tox.sh &
0 0 * * * cd $HOME/tox_logs && find -name "*" -type f -mtime
+3 | xargs rm -f
@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

18. Exit 4sightml user

```
exit
```

19. Login as nginx user

```
su nginx
```

20. Add the below lines in /usr/local/nginx/conf/nginx.conf file under both 80 and ssl server block

```
location /4sightml {
  root html;
  index index.html index.htm;
  proxy_pass http://localhost:6000;
  proxy_force_ranges on;
}
```

21. Exit nginx user

```
exit
```

22. Login as root user
23. Reload Nginx

```
service nginx restart
```