Contents

[Innoflex App Deployment Guide 2](#_Toc118475811)

[Innoflex App Installation 3](#_Toc118475812)

[1.Verify that docker service is running. 3](#_Toc118475813)

[2.Create docker secrets 3](#_Toc118475814)

[3.Create Path for Config file 6](#_Toc118475815)

[4.Create Path for log file 6](#_Toc118475816)

[5.Edit docker-compose.yml 6](#_Toc118475817)

[6.Load docker image 8](#_Toc118475818)

[7.Create services from docker-compose file. 8](#_Toc118475819)

[8.Config logs rotate 10](#_Toc118475820)

[9.Config crontab job for log rotate 11](#_Toc118475821)

[Devices Setting 12](#_Toc118475822)

[Login 12](#_Toc118475823)

[Device ID 12](#_Toc118475824)

[Door open condition 13](#_Toc118475825)

[Mqtt setting 15](#_Toc118475826)

[Verify with Heartbeat log 17](#_Toc118475827)

[Devices Name Pattern 17](#_Toc118475828)

# Innoflex App Deployment Guide

**Installation files**

In staging looking for folder innoflex, installation files can be found here.



Text

Description automatically generated

**Task to installation and manage devices**

Task 1. Innoflex App Installation (Page 3)

Task 2. Devices Setting (Page 11)

# Innoflex App Installation

## 1.Verify that docker service is running.

Command: systemctl status docker

Text

Description automatically generated

## 2.Create docker secrets

Innoflex app using docker secrets to store sensitive information.

2.1 Create a docker secret as listed below:

1. MongoDB database

**1.1 db\_user**

command: echo "<username>" | docker secret create db\_user –

**1.2 db\_pass**

command: echo "<password>" | docker secret create db\_pass -

1. AMQP on Alibaba cloud

Graphical user interface, text, application, chat or text message

Description automatically generated

**2.1 amqp\_user**

command: echo "<username>" | docker secret create amqp\_user –

**2.2 amqp\_pass**

command: echo "<password>" | docker secret create amqp\_pass –

1. Signature Verification for client in MQTT Instance (infapp01 is hostname for each environment be careful not to duplicate names in the same environment.)

Graphical user interface, text, application, email

Description automatically generated

**3.1 GID\_LAZADA\_WFM@@@infapp01-client**

command: echo "<username>" | docker secret create client\_user -

command: echo "<password>" | docker secret create client\_pass –

**3.2 GID\_LAZADA\_WFM@@@infapp01-heartbeat**

command: echo "<username>" | docker secret create hbt\_user –

command: echo "<password>" | docker secret create hbt\_pass –

**3.3 GID\_LAZADA\_WFM@@@infapp01-ack**

command: echo "<username>" | docker secret create ack\_user –

command: echo "<password>" | docker secret create ack\_pass –

**3.4 GID\_LAZADA\_WFM@@@infapp01-rec**

command: echo "<username>" | docker secret create rec\_user –

command: echo "<password>" | docker secret create rec\_pass –

**3.5 GID\_LAZADA\_WFM@@@infapp01-basic**

command: echo "<username>" | docker secret create basic\_user –

command: echo "<password>" | docker secret create basic\_pass –

* 1. Verify docker secret

command: docker secret ls

Text

Description automatically generated

Secret lists

|  |  |  |
| --- | --- | --- |
| No. | Secret Name | Objective |
| 1 | db\_user | MongoDB database username and password |
| 2 | db\_pass |
| 3 | amqp\_user | AMQP on Alibaba cloud username and password |
| 4 | amqp\_pass |
| 5 | client\_user | MQTT Client for Services  GID\_LAZADA\_WFM@@@infapp01-client |
| 6 | client\_pass |
| 7 | hbt\_user | MQTT Client for Services  GID\_LAZADA\_WFM@@@infapp01-heartbeat |
| 8 | hbt\_pass |
| 9 | ack\_user | MQTT Client for Services  GID\_LAZADA\_WFM@@@infapp01-ack |
| 10 | ack\_pass |
| 11 | rec\_user | MQTT Client for Services  GID\_LAZADA\_WFM@@@infapp01-rec |
| 12 | rec\_pass |
| 13 | basic\_user | MQTT Client for Services  GID\_LAZADA\_WFM@@@infapp01-basic |
| 14 | basic\_user |

## 3.Create Path for Config file

* 1. Create new folder for config file path /opt/innoflex/config

Command: mkdir /opt/innoflex/config

* 1. Copy configfile.ini in innoflex folder to /opt/innoflex/config

Command: cp /root/innoflex/configfile.ini /opt/innoflex/config

* 1. Verify that the file exists.

command: ls -la /opt/innoflex/config

Text

Description automatically generated

## 4.Create Path for log file

* 1. Create new folder path /var/log/innoflex

Command: mkdir /var/log/innoflex

* 1. Verify folder

Command: ls -la /var/log/innoflex

Text

Description automatically generated

## 5.Edit docker-compose.yml

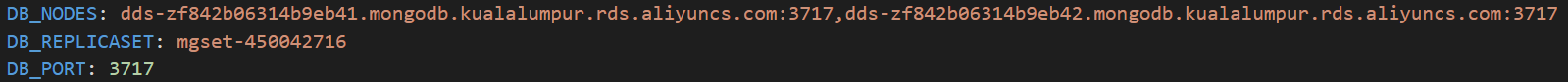
5.1 Change AMQP value: endpoint, virtualhost



5.2 Change MQTT value: endpoint, groupid



5.3 Change Database value: nodes, port, replicaset



5.4 Config container hostname

1. Name the container in every service as infapp01, which is the same name used to create the **MQTT Signature** client(infapp01 is hostname for each environment be careful not to duplicate names in the same environment.)

Example:

Text

Description automatically generated

## 6.Load docker image

* 1. Use cd command to access innoflex folder and check if infapp.tar.gz is existed

Calendar

Description automatically generated

* 1. Load docker image by use docker load command

Command: docker load < infapp.tar.gz

* 1. Check if image infapp is existed

Text

Description automatically generated

## 7.Create services from docker-compose file.

7.1 Create services

Command: docker stack deploy -c docker-compose.yml infapp

7.2 Check the operation of all services by use docker service command

Command: docker service ls

Text

Description automatically generated

7.3 Services lists

|  |  |  |
| --- | --- | --- |
| No. | Service Name | Objective |
| 1 | infapp\_attendance-get | Receive get attendance history from WFM |
| 2 | infapp\_attendance-sync | Send attendance message to WFM |
| 3 | infapp\_device-sync | Sync devices in database |
| 4 | infapp\_forwarder-ack | Receive acknowledge message from devices |
| 5 | infapp\_forwarder-hb | Receive heartbeat message from devices |
| 6 | infapp\_forwarder-rec | Receive attendance message from devices |
| 7 | infapp\_heartbeat | Log heartbeat sperate for each device |
| 8 | infapp\_worker-sync | Receive create worker message from WFM and sent MQTT message for update worker in devices |
| 9 | infapp\_devices-ack | Update acknowledge to database |
| 10 | infapp\_retry-transection | Resend MQTT message if worker is not registered in devices and send RES Message If registration is successful |
| 11 | infapp\_retry-transection | Send RES Message If registration is successful, failed or no respond after max retries |

7.4 Restart some service

Command:

1. docker ps (find container\_name of service)
2. docker rm -f <container\_name or container\_id>

Container will delete and create a new one

A computer screen capture

Description automatically generated with medium confidence

7.5 Clean all services before re-install

Clean all services command: docker stack rm infapp -f

Install service command: docker stack deploy -c docker-compose.yml infapp

7.6 See real-time log from container service

Command: docker service logs <service name>–follow

Example: docker service logs infapp\_heartbeat --follow

## 8.Config logs rotate

8.1 create new logs rotate file

Command: vi /etc/logrotate.d/innoflex.conf

8.2 insert config as shown below and save.

/var/log/innoflex/\*.log{

su root root

daily

copytruncate

create 0644 root root

rotate 15

compress

maxage 14

dateext

dateformat -%Y%m%d

}

## 9.Config crontab job for log rotate

9.1 open crontab

Command: crontab -e

9.2 add this line to crontab file and save.

59 23 \* \* \* /usr/sbin/logrotate -f /etc/logrotate.d/innoflex.conf

Text

Description automatically generated

9.3 log rotate will zip old log everyday

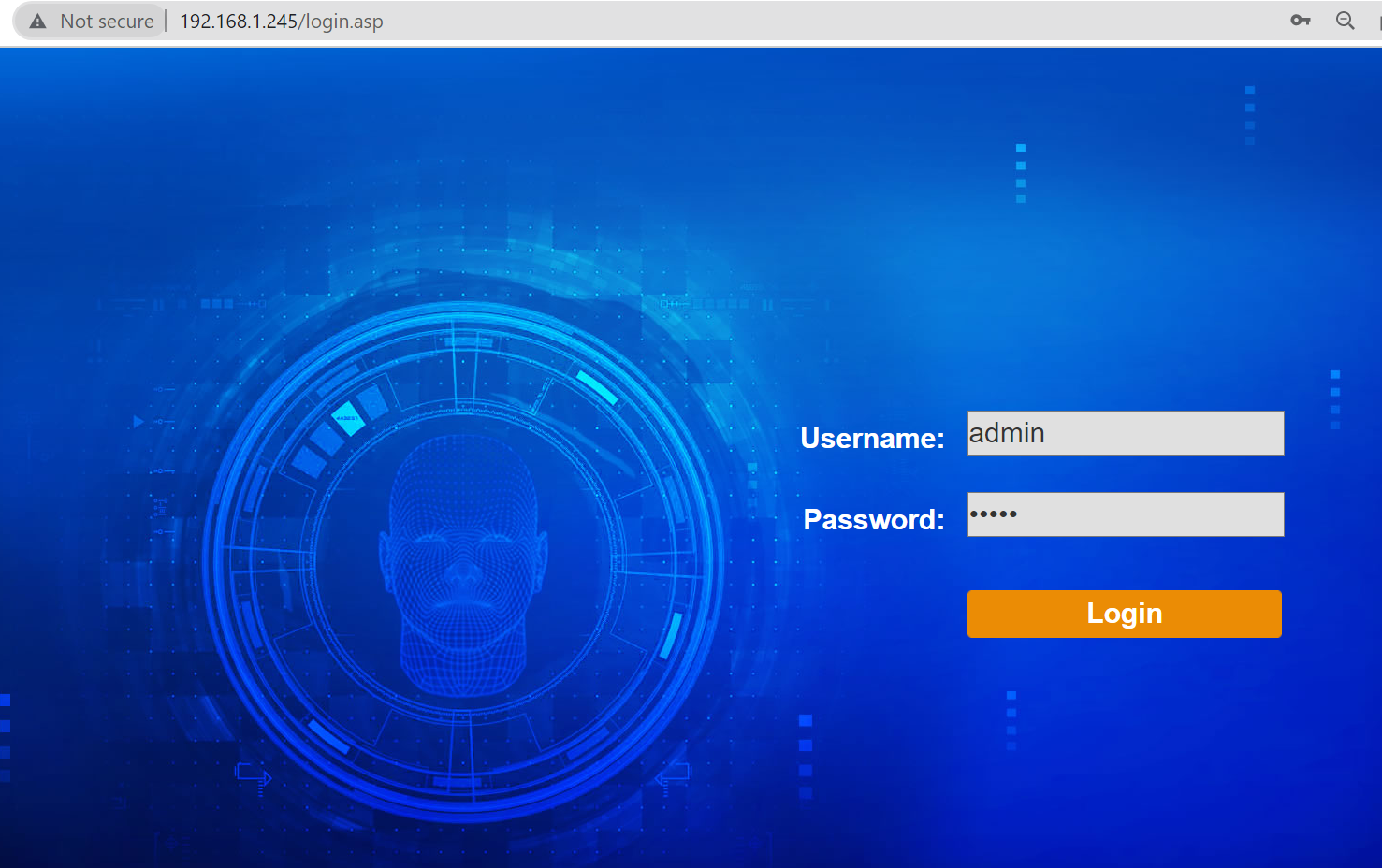
Graphical user interface

Description automatically generated with low confidence

# Devices Setting

## Login

Using device Ip to access to web service in terminal. Default username is admin, Default password is admin



## Device ID

To check device ID, go toSystem Parameter --> System Management --> System Info and IP Address

Graphical user interface, application

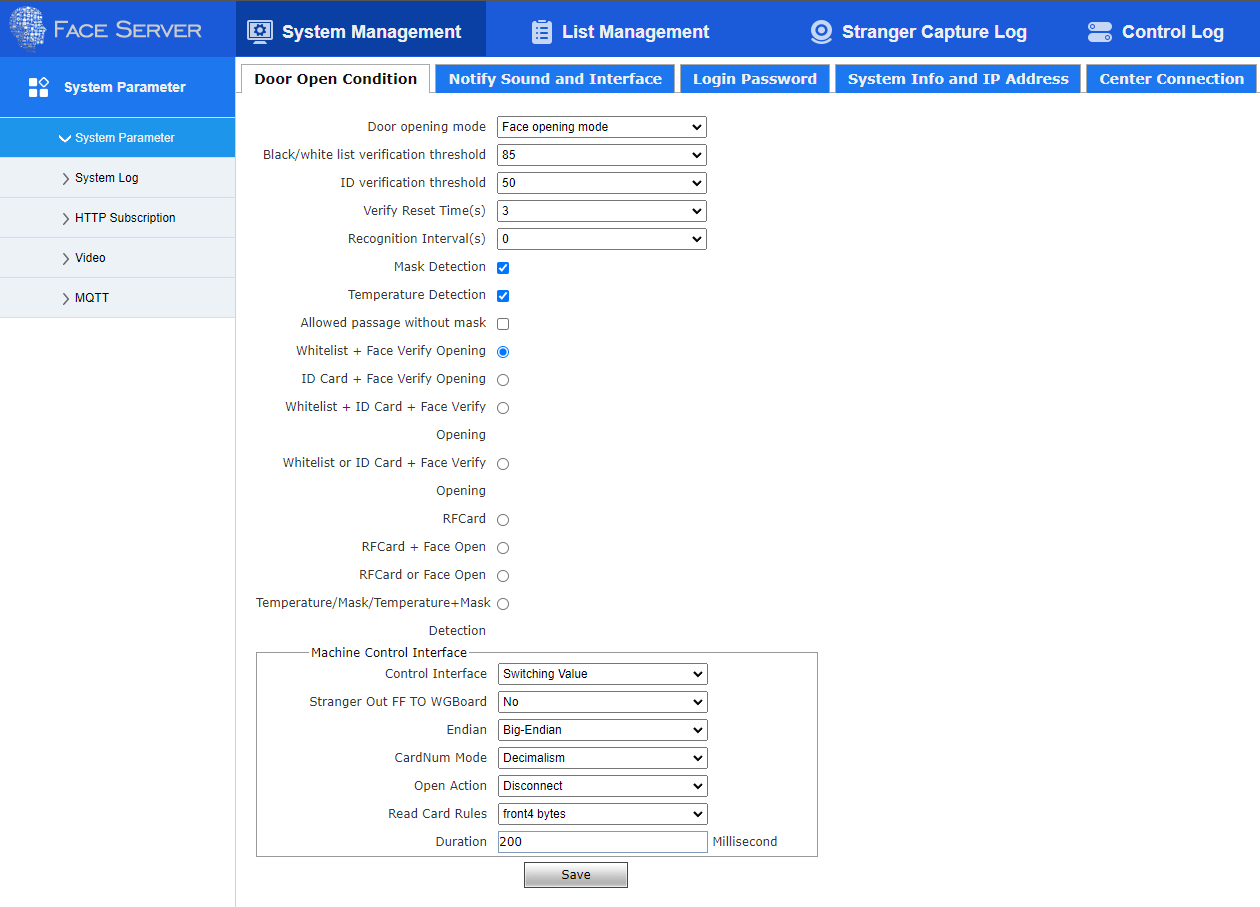
Description automatically generated

## Door open condition

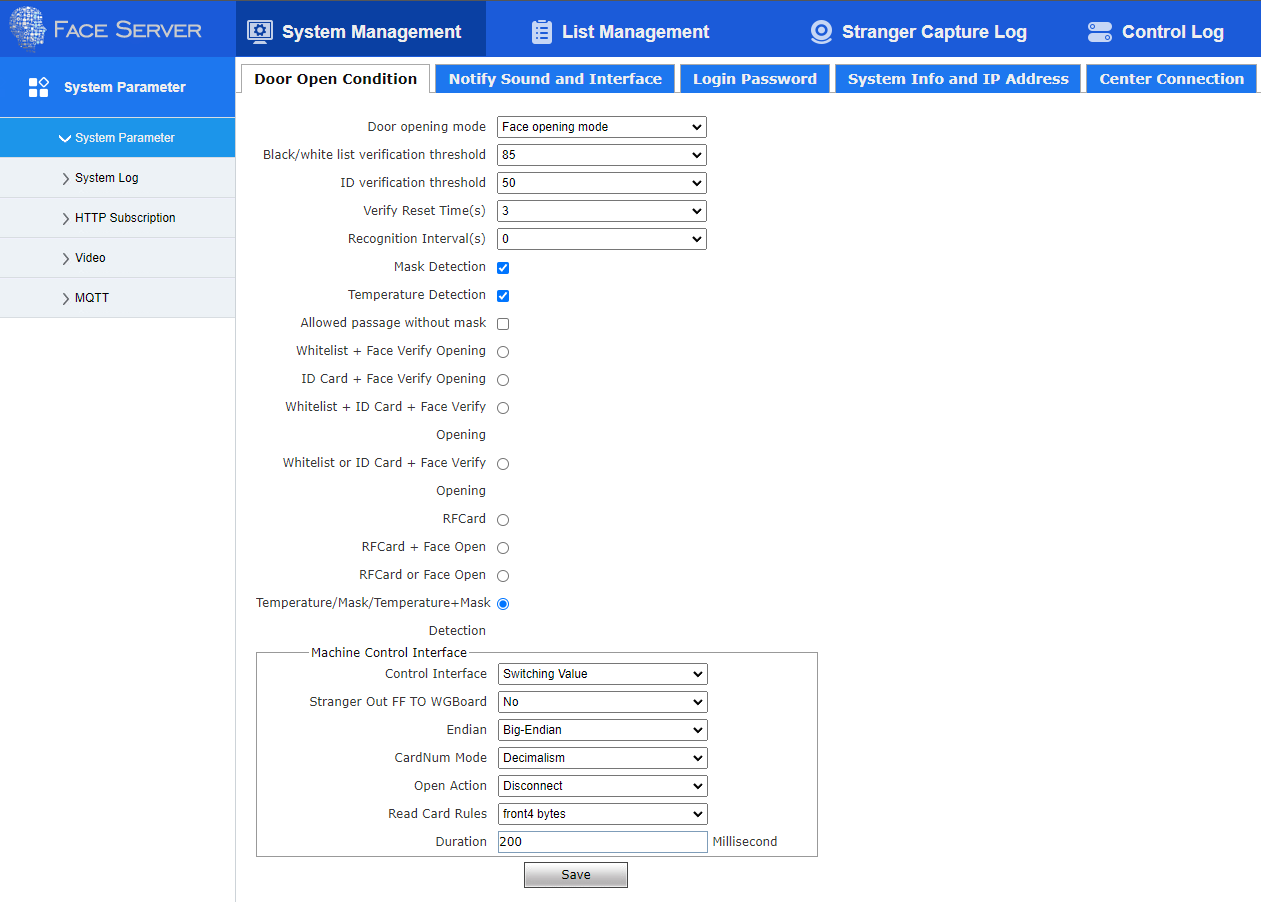
For integration with innoflex app the scanner should be set according to the instructions as shown in the picture.

System Parameter --> System Management --> Door Open Condition

1. **Whitelist face detection**



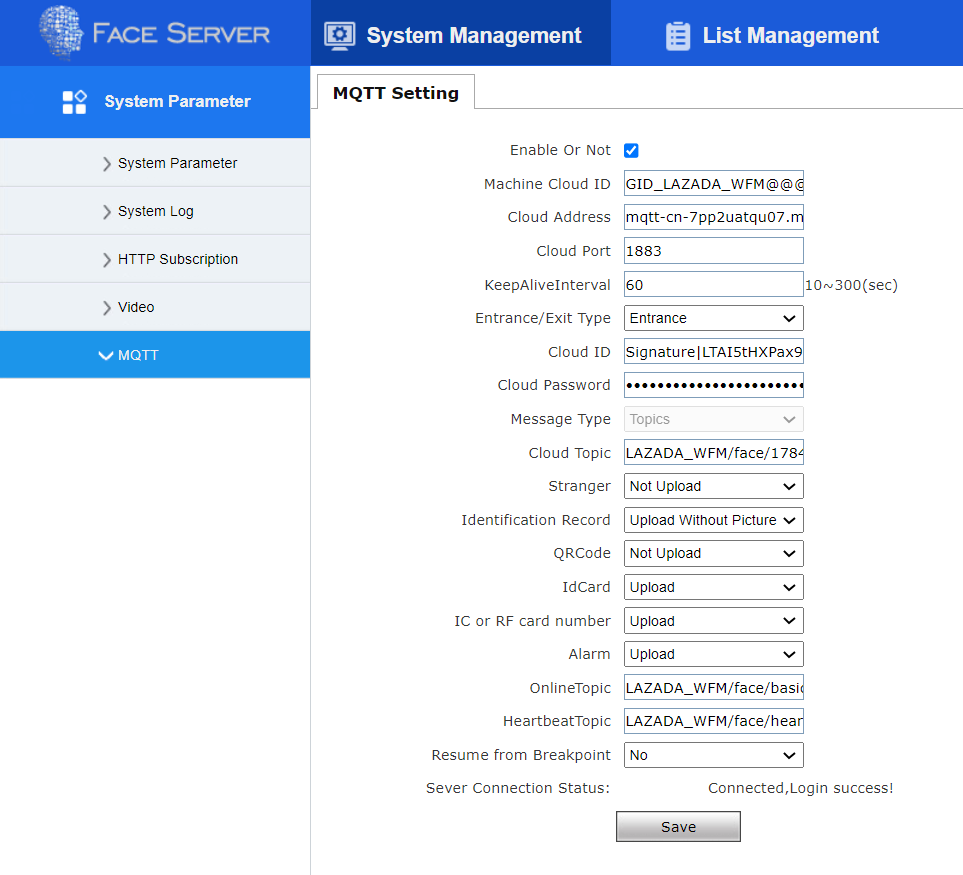
1. **Mask/thermal detection**



## Mqtt setting

For integration with innoflex app the scanner should be set according to the instructions as shown in the picture.

System Parameter --> MQTT



**Setting mqtt for each device. (1930215 = Device ID)**

**Example for SSW pilot site**

|  |  |  |
| --- | --- | --- |
| **Setting/Device ID** | **1930215** | **1930222** |
| Enable Or Not: | check | |
| Machine Cloud ID: | GID\_LAZADA\_WFM@@@1930215 | GID\_LAZADA\_WFM@@@1930222 |
| Cloud Address: | mqtt-cn-7pp2uatqu07.mqtt.aliyuncs.com | |
| Cloud Port: | 1883 | |
| KeepAliveInterval: | 60 | |
| Entrance/Exit Type: | Entrance | Exit |
| Cloud ID:  (Generate from Mqtt) | Signature|LTAI5tHXPax9Qmz7tbc9mZDs|mqtt-cn-7pp2uatqu07 | |
| Cloud Password:  (Generate from Mqtt) | vqJa4iwpGfSqcOGY4Bk9XMoA3CU= | VWTGQOsff6pR8hn/emz7lAC7QwU= |
| Message Type: | Topics | |
| Cloud Topic: | LAZADA\_WFM/face/1930215 | LAZADA\_WFM/face/1930222 |
| Stranger: | Not Upload | |
| Identification Record: | Upload Without Picture | |
| QRCode: | Upload | |
| IdCard: | Upload | |
| IC or RF card number: | Upload | |
| Alarm: | Upload | |
| OnlineTopic: | LAZADA\_WFM/face/basic | |
| HeartbeatTopic: | LAZADA\_WFM/face/heartbeat | |
| Resume from Breakpoint: | No | |

And the save. If device can connect the mqtt it will show

**Sever Connection Status: Connected, Login success!**

## Verify with Heartbeat log

After setting the MQTT, if the system is working properly, there will be heartbeat.log of that device occurs in the log folder

Command : ls -la /var/log/innoflex | grep <deviceID>\_heartbeat.log



Or check logs in infapp\_forwarder-hb  
 command: docker service logs infapp\_forwarder-hb –follow  


## Devices Name Pattern

All devices have to config name with pattern <facility>\_<direction>\_<number>

Example for SSW pilot site

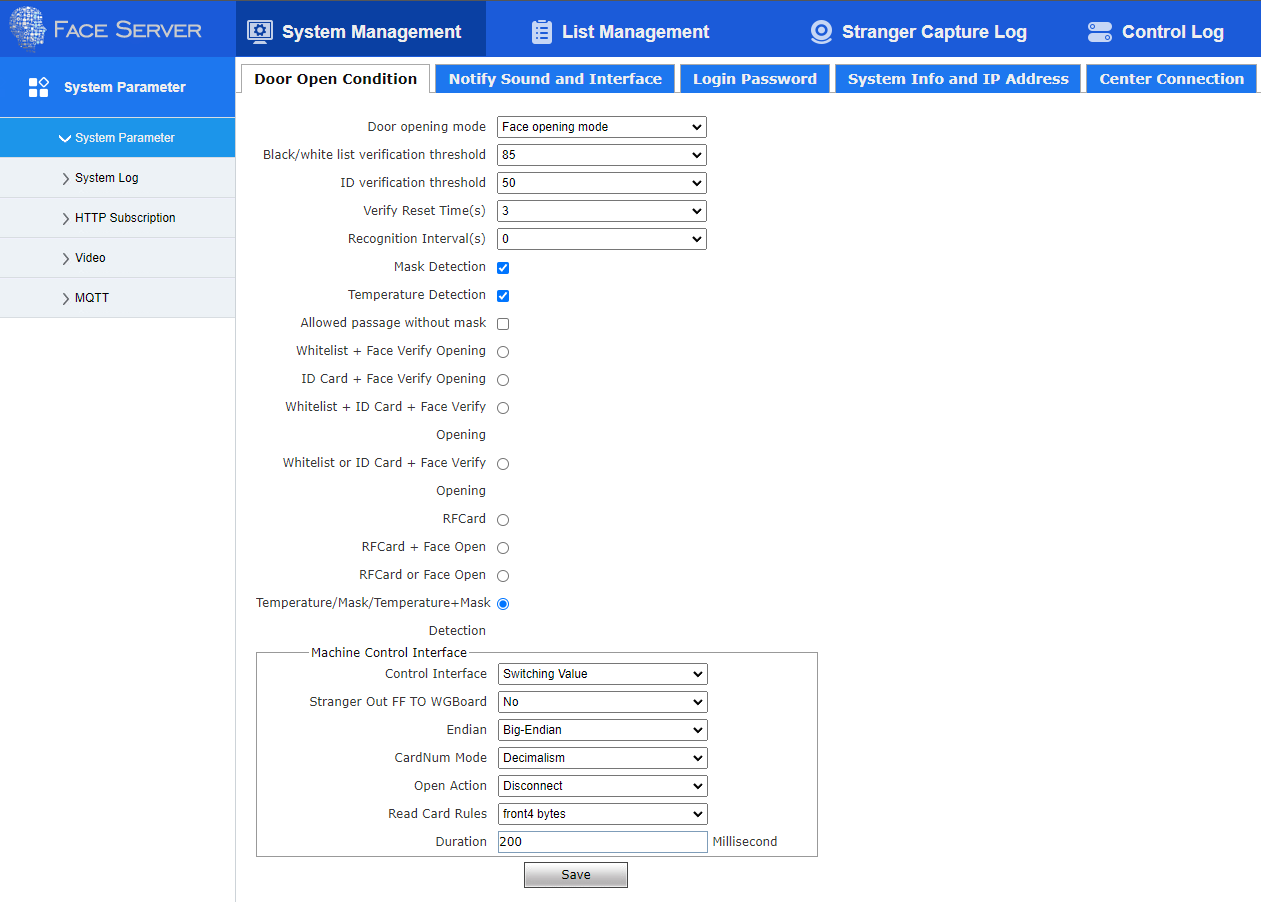
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Serial Number | Name | IP Address | Direction | Facility |
| 1 | 1930212 | SSW\_OUT\_01 | 30.125.78.208 | OUT | SSW |
| 2 | 1930214 | SSW\_OUT\_02 | 30.125.78.209 | OUT | SSW |
| 3 | 1930215 | SSW\_OUT\_03 | 30.125.78.210 | OUT | SSW |
| 4 | 1930217 | SSW\_IN\_04 | 30.125.78.211 | IN | SSW |
| 5 | 1930222 | SSW\_IN\_05 | 30.125.78.212 | IN | SSW |
| 6 | 1930226 | SSW\_IN\_06 | 30.125.78.213 | IN | SSW |

# Guide on Face Webservice to eliminate duplicate attendance

For eliminate duplicate attendance the scanner should be set according to the instructions as shown in the picture.

System Parameter --> System Management --> Door Open Condition

Set Verify Reset Time(s) = 3 (increase time to 3s)



# Guide on Face webservice to resync mis-attendance

For resync mis-attendance the scanner should be set according to the instructions as shown in the picture.

System Parameter --> MQTT

Set Resume from breakpoint = Enable

Select Date and time for resume attendance record

Click OK and save

Graphical user interface, application

Description automatically generated

# Worker sync flow

