
FindFace

Release 1.3

NtechLab

Sep 11, 2023

CONTENTS

1	System Administrator's Guide	3
1.1	Architecture	3
1.2	Requirements	8
1.3	Deploy and Remove FindFace	10
1.4	Administration and Configuration	39
1.5	Maintenance and Troubleshooting	82
1.6	Appendices	101
2	User's Guide	105
2.1	Getting Started	105
2.2	Record Index	106
2.3	Case Files	113
2.4	Search Faces in System	125
2.5	Compare Two Faces	126
2.6	Reports	127
2.7	Audit Log	129
2.8	Remote Alerting and Remote Search	129
3	Integrations	133
3.1	HTTP API	133

FindFace CIBR (Criminal Investigation Biometric Registry) is a forensic platform powered by a cutting-edge facial recognition technology. It is designed to conduct criminal investigations based on associated video footage and to search individuals across Public and Transport Safety systems.

FindFace forensic functionality

- **Biometric forensic databases.** Upload the databases to FindFace by creating a Record Index and assigning records of individuals under observation to relevant watch lists (Wanted, Fugitives, etc.). It can be done in bulk. A record accommodates aggregated data about an individual: a face biometric sample, document scans, criminal record, and other information.
- **Case files.** Upload a video footage of an incident to FindFace to detect and identify human faces in it. If there are individuals whose facial biometric data are in the forensic databases, FindFace will be able to spot them during this stage.

Go ahead and process the facial recognition results, using a built-in tool. Tell apart participants from bystanders and identify suspects and victims.
- **Search.** Search the system for specific individuals.
- **Remote alerting.** Combine FindFace with Public and Transport Safety systems. Receive real-time alerts on appearance of specific individuals from remote facial recognition systems. This will help track the offender's location and routes, detect alleged accomplices, find missing people.
- **Remote search.** Search specific individuals in remote facial recognition systems.
- **Facial verification.** Verify that two given faces belong to the same individual.
- **Reports.** Detailed reports on search results and records.

Technical features

- AI-based platform.
- Developer-friendly installer and user-friendly interface.
- Single- and multi-host deployment.
- Increased performance and fault-tolerance in high load systems with numerous cameras and clients.
- Network or on-premise licensing.
- CPU- and GPU-based acceleration for your choice.

System security

- Advanced user management. Possibility of integrating with Active Directory.
- Comprehensive, friendly, searchable audit logs.
- Backup and recovery utilities.
- Possibility of monitoring user sessions and blocking devices without deactivating user accounts.

Useful little things

- Quick record index creation.
- Extended set of search filters.

Integration

- Integration via HTTP API.

SYSTEM ADMINISTRATOR'S GUIDE

This chapter is all about FindFace deployment and further updates and maintenance during exploitation.

1.1 Architecture

Though you mostly interact with FindFace through its web interface, be sure to take a minute to learn the FindFace architecture. This knowledge is essential for the FindFace deployment, integration, maintenance, and troubleshooting.

In this chapter:

- *Recognition Process*
- *Architectural Elements*
 - *Architecture scheme*
 - *FindFace Core*
 - *FindFace Application Module*
- *Single- and Multi-Host Deployment*
- *CPU- and GPU-acceleration*

1.1.1 Recognition Process

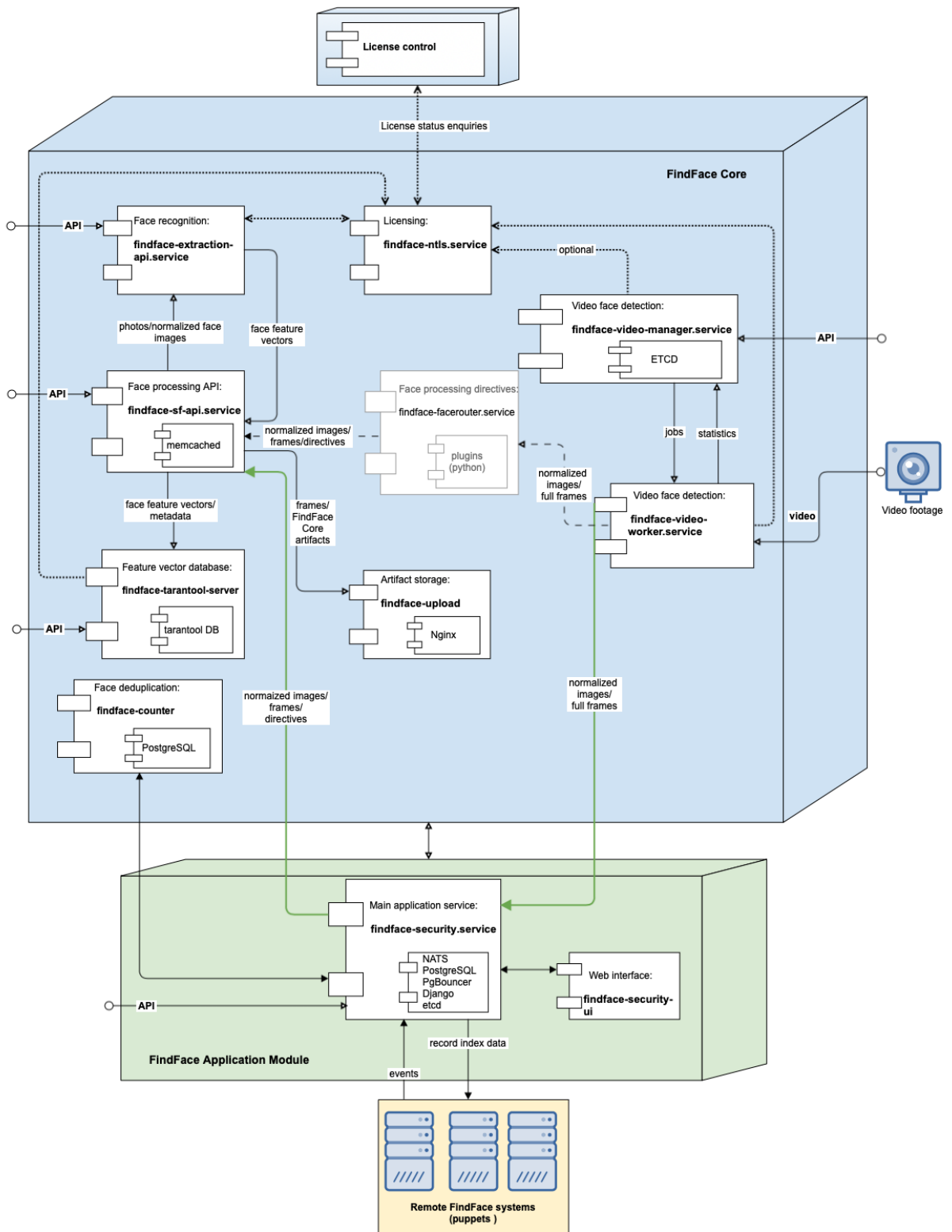
FindFace detects a human face in the photo or video and prepares its image through normalization. The normalized image is then used for extracting the face's feature vector (an n-dimensional vector of numerical features that represent the face). Face feature vectors are stored in the database and further used for verification and identification purposes.

1.1.2 Architectural Elements

FindFace consists of the following fundamental architectural elements:

- FindFace Core, a cutting-edge AI-based recognition technology that can be used as a separate product [FindFace Enterprise Server](#).
- FindFace Application Module, implementing a set of tools for criminal investigations based on video footage.

Architecture scheme



FindFace Core

The FindFace Core includes the following components:

Component	Ports in use	Description	Vendor
findface-extraction-api	18666	Service that uses neural networks to detect a face in an image and extract its feature vector. It also recognizes face attributes, for example, gender, age, emotions, beard, glasses, etc. CPU- or GPU-acceleration.	NtechLab own deployment
findface-sf-api	18411	Service that implements the internal HTTP API for face detection and recognition.	
findface-tarantool-server	32001, shard ports (default 330xx, 81xx)	Service that provides interaction between the <code>findface-sf-api</code> service and the feature vector database (the Tarantool-powered database that stores face feature vectors).	
findface-upload	3333	NginX-based web server used as a storage for original images, thumbnails, and normalized face images.	
findface-facerouter	18820	Service used to define processing directives for detected faces. In FindFace, its functions are performed by <code>findface-security</code> (see <i>FindFace Application Module</i>).	
findface-video-manager	18810, 18811	Service, part of the video face detection module, that is used for managing the video face detection functionality, configuring the video face detector settings and specifying the list of to-be-processed video files.	
findface-video-worker	18999	Service, part of the video face detection module, that recognizes a face in the video and posts its normalized image, full frame and metadata (such as detection time) to the <code>findface-facerouter</code> service for further processing according to given directives. CPU- or GPU-acceleration.	
findface-ntls	443 (TCP), 3133, 3185	License server that interfaces with the NtechLab Global License Server, a USB dongle, or hardware fingerprint to verify the <i>license</i> of your FindFace instance.	
findface-counter	18300	Service used for event deduplication.	
Tarantool	Shard ports (default 330xx, 81xx)	Third-party software that implements the feature vector database that stores extracted face feature vectors and identification events. The system data, records, user accounts are stored in PostgreSQL (part of the FindFace application module).	Tarantool
etcd	2379	Third-party software that implements a distributed key-value store for <code>findface-video-manager</code> . Used as a coordination service in the distributed system, providing the video face detector with fault tolerance.	etcd
NginX	80; SSL: 8002, 8003, 443, 80	Third-party software that implements the system web interfaces.	nginx
mem-cached	11211	Third-party software that implements a distributed memory caching system. Used by <code>findface-sf-api</code> as a temporary storage for extracted face feature vectors before they are written to the feature vector database powered by Tarantool.	mem-cached

FindFace Application Module

The FindFace application module includes the following components:

Component	Ports in use	Description	Vendor
findface-security	Configurable	Component that serves as a gateway to the FindFace Core. Provides interaction between the FindFace Core and the web interface, the system functioning as a whole, HTTP and web socket, face monitoring, event notifications, etc.	Ntech-Lab own deployment
findface-security-ui	Configurable	Main web interface used to interact with FindFace. Based on the Django framework . Allows you to work with face recognition events, search for faces, manage cases, users, record index, watch lists, and many more.	
NATS	4222	Third-party software that implements a message broker inside <code>findface-security</code> .	NATS
etcd	2379	Third-party software that implements locks in the <code>findface-security</code> service, such as locks in NTLs checker, reports, video processing, etc.	etcd
Pg-bouncer	5439	Third-party software, a lightweight connection pooler for PostgreSQL. Optional, used to increase the database performance under high load.	Pg-Bouncer
PostgreSQL	5432	Third-party software that implements the main system database. This database stores records of individuals and data for internal use. The face feature vectors and face identification events are stored in Tarantool (part of the FindFace Core).	PostgreSQL

See also:

- [FindFace Data Storages](#)

1.1.3 Single- and Multi-Host Deployment

You can deploy FindFace on a single host or in a multi-host environment. If you opt for the latter, we offer you one of the following deployment schemes:

- Deploy FindFace standalone and distribute additional `findface-video-worker` components across multiple hosts.
See [Additional findface-video-worker Deployment on Remote Hosts](#).
- Distribute the FindFace components across multiple hosts. If necessary, set up load balancing.
See [Guide to Typical Multi-Host Deployment](#).

1.1.4 CPU- and GPU-acceleration

The `findface-extraction-api` and `findface-video-worker` services can be either CPU- or GPU-based. During installation from the developer-friendly *installer*, you will have an opportunity to choose the acceleration type you need.

If you opt to install FindFace from the *repository package*, deploy the `findface-extraction-api` and `findface-video-worker-cpu` packages on a CPU-based server, and the `findface-extraction-api-gpu` and/or `findface-video-worker-gpu` packages on a GPU-based server.

Important: Refer to [Requirements](#) when choosing hardware configuration.

Important: If video resolution is more than 1280x720px, it is strongly recommended to use the GPU-accelerated package `findface-video-worker-gpu`.

1.2 Requirements

In this chapter:

- *System Requirements for Basic Configuration*
- *Required Administrator Skills*
- *Video File Formats*

1.2.1 System Requirements for Basic Configuration

To calculate the FindFace host(s) characteristics, use the requirements provided below.

Tip: Be sure to learn about the FindFace *architecture* first.

Important: If the video resolution is more than 1280x720px, it is strongly recommended to use the GPU-accelerated package `findface-video-worker-gpu`.

Important: On AMD CPU servers, the full functionality of the CPU-accelerated `findface-extraction-api` service is not guaranteed. Use the GPU-accelerated service `findface-extraction-api-gpu` along with the GPU-version of neural networks instead.

Note: In the case of a high-load system (~> 15 events per second), we recommend using an SSD.

	Minimum	Recommended
CPU	Intel Core i5 CPU with 4+ physical cores 3+ GHz. AVX2 support	Intel Xeon Silver/Gold with 6+ physical cores
	The own needs of FindFace require 2 cores HT > 2.5 GHz. The characteristics also depend on the number of video files in process. A single video file 720p@25FPS requires 2 cores >2.5 GHz. AVX2 support	
GPU (optional)	Nvidia Geforce® GTX 1060 6 GB	Nvidia Geforce® GTX 1080Ti+ with 11+ GB RAM
	Supported series: GeForce (Maxwell, Pascal, Turing, and above), Tesla (Maxwell, Pascal, Volta v100, Turing, and above)	
RAM	10 Gb	16+ Gb
	The own needs of FindFace require 8 Gb. The RAM consumption also depends on the number of video files in process. A single video file 720p@25FPS requires 2 GB RAM	
HDD (SSD for best performance)	16 Gb	16+ Gb
	The own needs of the operating system and FindFace require 15 GB. The total volume is subject to the required depth of the event archive in the database and in the log, at the rate of 1.5 Mb per 1 event	
Operating system	Ubuntu 18.04, x64 only	

Note: You can also use an Intel-based VM if there is AVX2 support, and eight physical cores are allocated exclusively to the VM.

Tip: For more accurate hardware selection, contact our support team by support@ntechlab.com.

1.2.2 Required Administrator Skills

A FindFace administrator must know and understand OS Ubuntu at the level of an advanced user.

1.2.3 Video File Formats

Video footage used for case investigations is accepted in a wide variety of formats, depending on the acceleration type, CPU or GPU.

Both CPU- and GPU-accelerated instances support all FFmpeg codecs. In addition to that, the following codecs are supported:

- *CPU-based acceleration:* FLV (both as a codec and as a container), H263, H264, H265, MJPEG, VP8, VP9, MPEG1VIDEO, MPEG2VIDEO, MSMPEG4v2, MSMPEG4v3.
- *GPU-based acceleration:* MJPEG, H264, H265, VP9, and others, depending on the list of codecs supported by the used video card. Apart from that, for GPU-accelerated instances, the CPU-based acceleration can be enabled, thus adding FLV support, which is not available by default.

1.3 Deploy and Remove FindFace

FindFace provides two basic deployment options:

- from a console installer
- step-by-step from an APT repository

These options fork into a variety of deployment cases covered by this section. You will also learn how to remove a FindFace instance from your server.

Important: Starting the GPU-accelerated services `findface-extraction-api` and `findface-video-worker-gpu` for the first time after deployment may take up a considerable amount of time due to the caching process (up to 45 minutes).

Important: Although FindFace provides tools to ensure its protection from unauthorized access, they are not replacing a properly configured firewall. Be sure to use a firewall to heighten the FindFace network protection.

1.3.1 Deploy from Console Installer

To deploy FindFace, use a developer-friendly console installer.

Tip: Before deployment, be sure to consult the *system requirements*.

Important: The FindFace host must have a static IP address in order to be running successfully. To make the IP address static, open the `/etc/network/interfaces` file and modify the current primary network interface entry as shown in the case study below. Be sure to substitute the suggested addresses with the actual ones, subject to your network specification.

```
sudo vi /etc/network/interfaces

iface eth0 inet static
address 192.168.112.144
netmask 255.255.255.0
gateway 192.168.112.254
dns-nameservers 192.168.112.254
```

Restart networking.

```
sudo service networking restart
```

Be sure to edit the `etc/network/interfaces` file with extreme care. Please refer to the Ubuntu [guide on networking](#) before proceeding.

To deploy FindFace from the console installer, do the following:

1. Download the installer file `findface-*.run`.
2. Put the `.run` file into some directory on the designated host (for example, `/home/username`).

- From this directory, make the `.run` file executable.

Note: Be sure to specify the actual file name instead of `findface-*`.

```
chmod +x findface-*.run
```

- Execute the `.run` file.

```
sudo ./findface-*.run
```

The installer will ask you a few questions and perform several automated checks to ensure that the host meets the system requirements. After filling out each prompt, press Enter. The questions and answers are the following:

- Q: Which product should be installed?

A: 1

```
Which product should be installed?

1. [security] FindFace Multi
2. [server ] FindFace Server
3. [video-worker] FindFace Video Worker
4. [nvidia-drivers] NVIDIA CUDA drivers (installed automatically when you
↳install gpu-variant of the products above)

(default: security)
product> 1
```

- Q: Please choose installation type:

A: Choose one of the following variants, subject to your architecture outline and deployment plan:

- 1: install FindFace standalone. Being the simplest, this installation type is excellent to start off with FindFace. The rest of the section covers the situation when you choose this installation type.
- 2: install FindFace and configure it to interact with additional remote `findface-video-worker` instances. See [Guide to Typical Multi-Host Deployment](#) for the detailed description.

Tip: To install only `findface-video-worker` on a host, refer to [Additional findface-video-worker Deployment on Remote Hosts](#).

- 3: install the apt repository for the step-by-step deployment. See [Step-by-Step Deployment from Repository](#) for the detailed description.
- 4: fully customized installation. See [Fully Customized Installation](#) for the detailed description.

Note: If you select the installation type #3 or #4, keep in mind to install necessary neural network models along with the `findface-extraction-api` component.

```
Please choose installation type:

- 1 [stand-alone ] Single Server
- 2 [multi-worker] Single Server, Multiple video workers
```

(continues on next page)

(continued from previous page)

```
- 3 [repo      ] Don't install anything, just set up the APT repository
- 4 [custom    ] Fully customized installation

(default: stand-alone)
type> 1
```

3. Q: Do you want to install Video Recorder?(y/n)

A: n

```
Do you want to install Video Recorder?(y/n)
install_video_recorder> n
```

4. Q: Found X interface(s). Which one should we announce as our external address?

A: Choose the interface that you are going to use as the instance IP address.

```
Found 1 interface(s). Which one should we announce as our external address?

- 1 [lo       ] 127.0.0.1
- 2 [ens3     ] 192.168.112.254

(default: 192.168.112.254)
ext_ip.advertised> 2
```

5. Q: Which variant of Video Worker should be installed?

A: Specify the findface-video-worker package type, CPU or GPU.

```
Which variant of Video Worker should be installed?

- 1 [cpu] CPU-based implementation, slower but doesn't require GPU
- 2 [gpu] CUDA-based implementation of video detector, requires NVIDIA GPU

(default: cpu)
findface-video-worker.variant> 1
```

6. Q: Which variant of Extraction API should be installed?

A: Specify the findface-extraction-api package type, CPU or GPU.

```
Which variant of Extraction API should be installed?

- 1 [cpu] CPU-only implementation, slower but doesn't require GPU
- 2 [gpu] CUDA-based implementation, faster, requires NVIDIA GPU (supports
↳both CPU and GPU models)

(default: cpu)
findface-extraction-api.variant> 1
```

7. Q: Do you want to configure detectors and features right now?(y/n)

A: y

```
Do you want to configure detectors and features right now?(y/n)
configure> y
```


8. Q: Do you want to configure detectors and features right now?(y/n)

A: y

```
Do you want to configure detectors and features right now?(y/n)
configure> y
```

9. Q: Please select detectors to install:

A: The face detector is selected by default. Enter done to proceed.

```
Please select detectors to install:

- 1 [v] Face
- 2 [ ] Body
- 3 [ ] Car

Enter keyword to select matching choices or -keyword to clear selection.
Enter "done" to save your selection and proceed to another step.
detectors>
- 1 [v] Face
- 2 [ ] Body
- 3 [ ] Car

detectors> done
```

10. Q: Please select face features to install:

A: By default, all face attributes are subject to installation. We recommend leaving it as is by answering done. If some attribute is not necessary, you can enter the keyword (number) related to it. For example, enter 7 to exclude the head pose recognition. Then enter done.

```
Please select face features to install:

- 1 [v] Age
- 2 [v] Gender
- 3 [v] Emotions
- 4 [v] Beard
- 5 [v] Glasses
- 6 [v] Medicine masks
- 7 [v] Headpose

Enter keyword to select matching choices or -keyword to clear selection.
Enter "done" to save your selection and proceed to another step.
face_features> done
```

The FindFace components will be automatically installed, configured and/or started in the following configuration:

Service	Configuration
postgresql-10	Installed and started.
nats-server	Installed and started.
etcd	Installed and started.
pg-bouncer	Installed and started.
memcached	Installed and started.
nginx	Installed and started.
django	Installed and started as a web framework for the FindFace web interface.
findface-ntls	Installed and started.
findface-tarantool-server	Installed and started. The number of instances (shards) is calculated using the formula: $N = \min(\max(\min(\text{mem_mb} // 2000, \text{cpu_cores}), 1), 16 * \text{cpu_cores})$. I.e., it is equal to the RAM size in MB divided by 2000, or the number of CPU physical cores (but at least one shard), or the number of CPU physical cores multiplied by 16, if the first obtained value is greater.
findface-extraction-api	Installed and started (CPU/GPU-acceleration).
findface-sf-api	Installed and started.
findface-upload	Installed.
findface-video-manager	Installed and started.
findface-video-worker-*	Installed and started (CPU/GPU-acceleration).
findface-data-*	Neural network models for object and object attribute recognition. Installed.
findface-security	Installed and started.
findface-security-onvif	Installed and started.
findface-counter	Installed and started.
findface-liveness-api	Installed and started.
jq	Installed. Used to pretty-print API responses from FindFace.
python3-ntech.*	Internal and auxiliary services. Installed and started.

After the installation is complete, the following output is shown on the console:

Tip: Be sure to save this data: you will need it later.

```
#####
#                               Installation is complete                               #
#####
- upload your license to http://192.168.112.254/#/license/
- user interface: http://192.168.112.254/
  superuser:      admin
  password:       admin
  documentation:  http://192.168.112.254/doc/
```

- Specify your time zone in the `/etc/findface-security/config.py` configuration file, either in the Region/Country/City or Etc/GMT+H format. The time zone determines the time in reports, logs, and names of such FindFace artifacts as event full frames and thumbnails, etc.

Tip: The best way to do so is to copy/paste your time zone from [this table](#) on Wikipedia.

```
sudo vi /etc/findface-security/config.py

# time zone
TIME_ZONE = 'America/Argentina/Buenos_Aires'
```

- Restart the `findface-security` service.

```
sudo systemctl restart findface-security.service
```

- Upload the FindFace license file via the main web interface `http://<Host_IP_address>/#/license`. To access the web interface, use the provided superuser credentials.

Important: Do not disclose the superuser (Super Administrator) credentials to others. To administer the system, create a new user with administrator privileges. Whatever the role, the Super Administrator cannot be deprived of its rights.

- The answers to the installer questions were saved to a file `/tmp/<findface-installer-*.json`. You can edit this file and use it to install FindFace on other hosts without having to answer the questions again.

To do so, execute:

```
sudo ./<findface-*.run -f /tmp/<findface-installer-*.json
```

Tip: You can find an example of the installation file in *Installation File*.

Important: To preserve the FindFace compatibility with the installation environment, we highly recommend you to disable the Ubuntu automatic update. In this case, you will be able to update your OS manually, fully controlling which packages to update.

To disable the Ubuntu automatic update, execute the following commands:

```
sudo apt-get remove unattended-upgrades
sudo systemctl stop apt-daily.timer
sudo systemctl disable apt-daily.timer
sudo systemctl disable apt-daily.service
sudo systemctl daemon-reload
```

Important: The FindFace services log a large amount of data, which can eventually lead to disc overload. To prevent this from happening, we advise you to disable `rsyslog` due to its suboptimal log rotation scheme and use the appropriately configured `systemd-journal` service instead. See *Service Logs* for the step-by-step instructions.

1.3.2 Step-by-Step Deployment from Repository

This section will guide you through the FindFace step-by-step deployment process. Follow the instructions below minding the sequence.

Tip: Be sure to learn the FindFace *architecture* first.

In this section:

- *Install APT Repository*
- *Prerequisites*
- *Deploy License Server*
- *Deploy Main Database*
- *Deploy FindFace Core*
- *Deploy FindFace Application Module and Feature Vector Database*

Install APT Repository

First of all, install the FindFace apt repository as follows:

1. Download the installer file `findface-*.run`.
2. Put the `.run` file into some directory on the designated host (for example, `/home/username`).
3. From this directory, make the `.run` file executable.

Note: Be sure to specify the actual file name instead of `findface-*`.

```
chmod +x findface-*.run
```

4. Execute the `.run` file.

```
sudo ./findface-*.run
```

The installer will ask you a few questions and perform several automated checks to ensure that the host meets the system requirements. After filling out each prompt, press Enter. The questions and answers are the following:

1. Q: Which product should be installed?

A: 1

```
Which product should be installed?

1. [security] FindFace Multi
2. [server ] FindFace Server
3. [video-worker] FindFace Video Worker
4. [nvidia-drivers] NVIDIA CUDA drivers (installed automatically when you
↳install gpu-variant of the products above)

(default: security)
product> 1
```

2. Q: Please choose installation type:

A: 3

```
Please choose installation type:

- 1 [stand-alone ] Single Server
- 2 [multi-worker] Single Server, Multiple video workers
- 3 [repo         ] Don't install anything, just set up the APT repository
- 4 [custom       ] Fully customized installation

(default: stand-alone)
type> 3
```

3. Q: APT repository doesn't include recognition models. Do you want to install them now?(y/n)

A: y or n, subject to your deployment plan. Whatever the case, keep in mind to install necessary neural network models along with the `findface-extraction-api` component. To install them later, refer to *Installation of Neural Network Models*.

```
APT repository doesn't include recognition models. Do you want to install them
↳now?(y/n)
(default: yes)
repo_data> y
```

4. Q: Select models to install (if you entered y on the previous step)

A: By default, all neural network models are subject to installation. You can leave it as is by entering `done`, or select specific models. To do so, deselect all those on the list by entering `-*` in the command line, then select the required models by entering their sequence numbers (keyword): for example, `1 3 4`. Enter `done` to save your selection and proceed to another step.

```
Select models to install.
Note that you will need to accordingly edit extraction-api and tntapi
↳configuration files.
```

(continues on next page)

(continued from previous page)

```

At least one of recognition models has to be enabled.

- 1 [v] ./findface-data-age.v2-cpu_3.0.0_all.deb
...
...
- 68 [v] ./findface-data-quality.v1-gpu_3.0.0_all.deb

Enter keyword to select matching choices or -keyword to clear selection.
Enter "done" to save your selection and proceed to another step.
findface-data.models> done

```

After that, the FindFace apt repository will be automatically installed.

Prerequisites

FindFace requires such third-party software as PostgreSQL, PgBouncer, NATS, etcd, and memcached. Do the following:

1. Install the prerequisite packages as such:

```

sudo apt update
sudo apt install -y postgresql-10 nats-server etcd memcached pgbouncer

```

2. Open the `/etc/memcached.conf` configuration file. Set the maximum memory in megabytes to use for memcached items: `-m 1024`. Set the maximum item size: `-I 16m`. If one or both of these parameters are absent, add them to the file.

```

sudo vi /etc/memcached.conf

-m 1024
-I 16m

```

3. Give a strong password to the `ntech` user (`9T3g1nXy9yx3y8MIGm9fbef3dia8UTc3` in the example below). Output the credentials to the `pgbouncer` user list.

```

echo "ntech" "9T3g1nXy9yx3y8MIGm9fbef3dia8UTc3" | sudo tee -a /etc/pgbouncer/
↪userlist.txt

```

4. Configure `pgbouncer`. In `/etc/pgbouncer/pgbouncer.ini`, paste the following content instead of the existing one, as shown in the example below.

```

sudo vi /etc/pgbouncer/pgbouncer.ini

[databases]
ffsecurity = dbname=ffsecurity host=localhost port=5432 user=ntech
ffsecurity_session = dbname=ffsecurity host=localhost port=5432 user=ntech pool_
↪mode=session pool_size=10
[pgbouncer]
pidfile = /var/run/postgresql/pgbouncer.pid
listen_addr = 127.0.0.1
listen_port = 5439
unix_socket_dir = /var/run/postgresql

```

(continues on next page)

(continued from previous page)

```

auth_type = plain
auth_file = /etc/pgbouncer/userlist.txt
pool_mode = transaction
server_reset_query = DISCARD ALL
max_client_conn = 16384
default_pool_size = 70
syslog = 1
log_connections = 0
log_disconnections = 0
stats_period = 300

```

5. Enable the prerequisite services autostart on boot and re-launch the services:

```

sudo systemctl enable postgresql@10-main.service nats-server etcd.service memcached.
↔service pgbouncer.service
sudo systemctl restart postgresql@10-main.service nats-server etcd.service
↔memcached.service pgbouncer.service

```

Deploy License Server

Important: See *Licensing* to learn about the NtechLab licensing policy.

To provide the FindFace licensing, deploy `findface-ntls`, license server in the FindFace core.

Important: There must be only one `findface-ntls` instance in each FindFace installation.

```

sudo apt update
sudo apt install -y findface-ntls
sudo systemctl enable findface-ntls.service && sudo systemctl start findface-ntls.service

```

Deploy Main Database

In FindFace, the main system database is based on PostgreSQL. To deploy the main database, do the following:

1. Open the pgbouncer list of users `/etc/pgbouncer/userlist.txt`. Copy the ntech user's password (9T3g1nXy9yx3y8MIGm9fbef3dia8UTc3 in the example below).

```

sudo cat /etc/pgbouncer/userlist.txt

"ntech" "9T3g1nXy9yx3y8MIGm9fbef3dia8UTc3"

```

2. Using the **PostgreSQL** console, create a new user `ntech` with the copied password, and databases `ffsecurity` and `ffcounter` in PostgreSQL.

```

sudo -u postgres psql

postgres=# CREATE ROLE ntech WITH LOGIN PASSWORD '9T3g1nXy9yx3y8MIGm9fbef3dia8UTc3';

```

(continues on next page)

(continued from previous page)

```
postgres=# CREATE DATABASE ffsecurity WITH OWNER ntech ENCODING 'UTF-8' LC_COLLATE=
↳ 'en_US.UTF-8' LC_CTYPE='en_US.UTF-8' TEMPLATE template0;

postgres=# CREATE DATABASE ffcounter WITH OWNER ntech ENCODING 'UTF-8' LC_COLLATE='C.
↳ UTF-8' LC_CTYPE='C.UTF-8' TEMPLATE template0;
```

Tip: To quit from the **PostgreSQL** console, type `\q` and press Enter.

3. Allow authentication by UID of a socket client in **PostgreSQL**. Restart **PostgreSQL**.

```
echo 'local all ntech peer' | sudo tee -a /etc/postgresql/10/main/pg_hba.conf

sudo systemctl restart postgresql@10-main.service
```

Deploy FindFace Core

To deploy the FindFace core, do the following:

Tip: You can find the description of the FindFace core components in *Architecture*.

1. For FindFace on GPU, *install NVIDIA drivers*.

Important: Be sure to restart the server after the NVIDIA drivers installation is complete. Otherwise, the subsequent installation of the GPU-based components will experience a failure.

2. Install the FindFace core components:

On CPU:

```
sudo apt update
sudo apt install -y findface-tarantool-server findface-extraction-api findface-sf-
↳ api findface-upload findface-video-manager findface-video-worker-cpu
```

On GPU:

```
sudo apt update
sudo apt install -y findface-tarantool-server findface-extraction-api-gpu findface-
↳ sf-api findface-upload findface-video-manager findface-video-worker-gpu
```

Note: If you have several video cards on your server, see *Multiple Video Cards Usage*.

Important: Keep in mind to *manually install* neural network models on the host(s) with `findface-extraction-api` later.

3. In the `/etc/findface-extraction-api.ini` configuration file, switch the neural network model for face recognition to `mango_320`.

On CPU:

```
sudo vi /etc/findface-extraction-api.ini

extractors:
    ...
models:
    ...
    face_emben: face/mango_320.cpu.fnk
    ...
```

On GPU:

```
sudo vi /etc/findface-extraction-api.ini

extractors:
    ...
models:
    ...
    face_emben: face/mango_320.gpu.fnk
    ...
```

4. In the `/etc/findface-sf-api.ini` configuration file, enable the `allow-return-facen` parameter.

```
sudo vi /etc/findface-sf-api.ini

...
limits:
    ...
    allow-return-facen: true
    ...
```

5. Open the `/etc/findface-video-worker-cpu.ini` (`/etc/findface-video-worker-gpu.ini`) configuration file. Specify the following parameters:

- In the `mgr-static` parameter, specify the `findface-video-manager` host IP address, which provides `findface-video-worker` with settings and the video stream list.
- In the `capacity` parameter, specify the maximum number of video streams to be processed by `findface-video-worker`.

```
sudo vi /etc/findface-video-worker-cpu.ini
sudo vi /etc/findface-video-worker-gpu.ini

mgr-static=127.0.0.1:18811

capacity=10

[streamer]
```

(continues on next page)

(continued from previous page)

```
#-----  
## streamer/shots webserver port, 0=disabled  
## type:number env:CFG_STREAMER_PORT longopt:--streamer-port  
port = 18999  
  
## streamer url - how to access this worker on streamer_port  
## type:string env:CFG_STREAMER_URL longopt:--streamer-url  
url = 127.0.0.1:18999
```

6. Enable the FindFace core services autostart and launch the services.

On CPU:

```
sudo systemctl enable findface-extraction-api findface-sf-api findface-video-  
↪manager findface-video-worker-cpu  
sudo systemctl start findface-extraction-api findface-sf-api findface-video-manager_↪  
↪findface-video-worker-cpu
```

On GPU:

```
sudo systemctl enable findface-extraction-api findface-sf-api findface-video-  
↪manager findface-video-worker-gpu  
sudo systemctl start findface-extraction-api findface-sf-api findface-video-manager_↪  
↪findface-video-worker-gpu
```

Deploy FindFace Application Module and Feature Vector Database

To deploy the FindFace application module, do the following:

1. Install the `findface-security`, `findface-security-ui`, and `findface-counter` components. Enable the `findface-counter` autostart and launch the service.

```
sudo apt update  
sudo apt install -y findface-security findface-security-ui findface-counter  
sudo systemctl enable findface-counter && sudo systemctl start findface-counter
```

2. Create a structure of the Tarantool-based feature vector database by executing the command below.

```
sudo findface-security make_tnt_schema | sudo tee /etc/findface-security/tnt_schema.  
↪lua
```

3. Open the `/etc/tarantool/instances.available/FindFace.lua` configuration file. Check whether it contains the `dofile` command and the `spaces` definition, as in the example below.

```
sudo vi /etc/tarantool/instances.available/FindFace.lua  
  
dofile("/etc/findface-security/tnt_schema.lua")
```

(continues on next page)

(continued from previous page)

```
-- host:port to bind, HTTP API
FindFace = require("FindFace")
FindFace.start("127.0.0.1", 8101, {
license_ntls_server="127.0.0.1:3133",
replication = replication_master,
spaces = spaces
})
```

Important: The IP address and port number specified in the shards section of the `/etc/findface-sf-api.ini` configuration file must be identical to those in the `FindFace.start` section.

```
sudo vi /etc/tarantool/instances.available/FindFace.lua

...

FindFace.start("127.0.0.1", 8101...)
```

```
sudo vi /etc/findface-sf-api.ini

storage-api:
...
shards:
- master: http://127.0.0.1:8101/v2/
...
```

Important: If you change the `/etc/findface-sf-api.ini` configuration file, be sure to restart the `findface-sf-api` service:

```
sudo systemctl restart findface-sf-api.service
```

4. Enable the `findface-tarantool-server` service autostart and launch the service.

```
sudo systemctl enable tarantool@FindFace.service && sudo systemctl start_
↪tarantool@FindFace.service
```

5. Open the `/etc/findface-security/config.py` configuration file. Specify the following parameters:

Tip: You can find the `/etc/findface-security/config.py` default version [here](#).

- `EXTERNAL_ADDRESS`: (Optional) IP address or URL used to access the FindFace web interface. If this parameter is not manually set, the system auto-detects it as the external IP address of the host.

Note: To access FindFace, you can use both the auto-detected and manually set IP addresses.

- `VIDEO_DETECTOR_TOKEN`: to authorize the video object detection module, come up with a token and specify it here.

Tip: It's a good idea to generate a token by executing:

```
pwgen -sncy 50 1|tr "" "."
```

- VIDEO_MANAGER_ADDRESS: IP address of the `findface-video-manager` host.
- NTLS_HTTP_URL: IP address of the `findface-ntls` host.
- ROUTER_URL: IP address of the `findface-security` host that will receive detected objects from the `findface-video-worker` instance(s). Specify either external or internal IP address, subject to the network through which `findface-video-worker` interacts with `findface-security`. Change the default port, subject to the *redirect settings* from HTTP to HTTPS, or omit it leaving only the IP address.
- SF_API_ADDRESS: IP address of the `findface-sf-api` host.
- DATABASES (section): fill it in as such: 'PORT': 5439, 'USER': 'ntech', 'PASSWORD': '<password from /etc/pgbouncer/userlist.txt>' (see *Prerequisites*).

Tip: If necessary, ensure data security by enabling *SSL*.

6. Generate a signature key for the session encryption (used by Django) by executing the command below. Specify this key as `SECRET_KEY`.

```
pwgen -sncy 50 1|tr "" "."
```

7. Migrate the database architecture from FindFace to **PostgreSQL**, create *predefined* user roles and the first administrator (a.k.a. Super Administrator).

Important: The Super Administrator cannot be deprived of its rights, whatever the role.

```
sudo findface-security migrate
sudo findface-security create_groups
sudo findface-security create_default_user
```

8. Start the services.

```
sudo systemctl enable findface-security
sudo systemctl start findface-security
```

9. Disable the default nginx server and add the `findface-security` server to the list of enabled servers. Restart nginx.

```
sudo rm /etc/nginx/sites-enabled/default

sudo ln -s /etc/nginx/sites-available/ffsecurity-nginx.conf /etc/nginx/sites-
->enabled/

sudo nginx -s reload
```

10. Provide licensing:

- Use the FindFace main web interface to *upload the license file* you have prior received from your manager (*Settings -> License*).

- For the on-premise licensing via a USB dongle, insert it into a USB port.
- For the on-premise licensing via hardware fingerprint, refer to *Offline Licensing via Hardware Fingerprint*.

Important: To log in for the first time, use the default Super Administrator account `admin:admin`.

Note: To create more users or change the Super Administrator password, refer to *Role and User Management*.

Important: To preserve the FindFace compatibility with the installation environment, we highly recommend you to disable the Ubuntu automatic update. In this case, you will be able to update your OS manually, fully controlling which packages to update.

To disable the Ubuntu automatic update, execute the following commands:

```
sudo apt-get remove unattended-upgrades
sudo systemctl stop apt-daily.timer
sudo systemctl disable apt-daily.timer
sudo systemctl disable apt-daily.service
sudo systemctl daemon-reload
```

Important: The FindFace services log a large amount of data, which can eventually lead to disc overload. To prevent this from happening, we advise you to disable `rsyslog` due to its suboptimal log rotation scheme and use the appropriately configured `systemd-journal` service instead. See *Service Logs* for the step-by-step instructions.

1.3.3 Additional findface-video-worker Deployment on Remote Hosts

Important: Before deploying `findface-video-worker` instances on remote hosts, do the following:

1. Allow accessing the `findface-ntls` license server from any IP address. To do so, open the `/etc/findface-ntls.cfg` configuration file on the server with `findface-ntls` and set `listen = 0.0.0.0:3133`. Restart the `findface-ntls` service.

```
sudo vi /etc/findface-ntls.cfg

## Address to accept incoming client connections (IP:PORT)
## type:string env:CFG_LISTEN longopt:--listen
listen = 0.0.0.0:3133
```

```
sudo systemctl restart findface-ntls.service
```

2. Allow accessing the `findface-video-manager` service from any IP address. To do so, open the `/etc/findface-video-manager.conf` configuration file on the server with `findface-video-manager` and set `listen: 0.0.0.0:18810` and `rpc:listen: 0.0.0.0:18811`. Restart the `findface-video-manager` service.

```
sudo vi /etc/findface-video-manager.conf
```

```
listen: 0.0.0.0:18810
...
rpc:
  listen: 0.0.0.0:18811
```

```
sudo systemctl restart findface-video-manager.service
```

3. On the FindFace server, open the `/etc/findface-security/config.py` configuration file and make sure that the `ROUTER_URL` parameter contains the external IP address of the FindFace server and not the localhost. The `findface-video-worker` instances on the remote hosts will be using this address for posting objects.

```
sudo vi /etc/findface-security/config.py
```

```
...
'ROUTER_URL': 'http://192.168.0.12',
...
```

To install only a `findface-video-worker` service, do the following:

Tip: Before deployment, be sure to consult the [system requirements](#).

Tip: If you have several video cards on your server, see [Multiple Video Cards Usage](#) before deploying `findface-video-worker-gpu`.

1. Download the installer file `findface-*.run`.
2. Put the `.run` file into some directory on the designated host (for example, `/home/username`).
3. From this directory, make the `.run` file executable.

Note: Be sure to specify the actual file name instead of `findface-*`.

```
chmod +x findface-*.run
```

4. Execute the `.run` file.

```
sudo ./findface-*.run
```

The installer will ask you a few questions and perform several automated checks to ensure that the host meets the system requirements. After filling out each prompt, press `Enter`. The questions and answers are the following:

1. Product to install: FindFace Video Worker.
2. Type of `findface-video-worker` package: CPU or GPU.
3. IP address of the `findface-security` host.

After that, the installation process will automatically begin.

Note: If you chose to install `findface-ntls` and/or `findface-video-manager` on different hosts than that with `findface-security`, specify their IP addresses in the `/etc/findface-video-worker-cpu.ini` (`/etc/findface-video-worker-gpu.ini`) configuration file after the installation.

```
sudo vi /etc/findface-video-worker-cpu.ini
sudo vi /etc/findface-video-worker-gpu.ini
```

In the `ntls-addr` parameter, specify the `findface-ntls` host IP address.

```
ntls-addr=127.0.0.1:3133
```

In the `mgr-static` parameter, specify the `findface-video-manager` host IP address, which provides `findface-video-worker` with settings and the video stream list.

```
mgr-static=127.0.0.1:18811
```

Tip: To automatically install `findface-video-worker` on another host without answering the installation questions, use the `/tmp/<findface-installer-*>.json` file. Execute:

```
sudo ./<findface-*>.run -f /tmp/<findface-installer-*>.json
```

You can find an example of the installation file in *Installation File*.

Important: To preserve the FindFace compatibility with the installation environment, we highly recommend you to disable the Ubuntu automatic update. In this case, you will be able to update your OS manually, fully controlling which packages to update.

To disable the Ubuntu automatic update, execute the following commands:

```
sudo apt-get remove unattended-upgrades
sudo systemctl stop apt-daily.timer
sudo systemctl disable apt-daily.timer
sudo systemctl disable apt-daily.service
sudo systemctl daemon-reload
```

Important: The FindFace services log a large amount of data, which can eventually lead to disc overload. To prevent this from happening, we advise you to disable `rsyslog` due to its suboptimal log rotation scheme and use the appropriately configured `systemd-journal` service instead. See *Service Logs* for the step-by-step instructions.

1.3.4 Installation of Neural Network Models

To detect and recognize faces and face attributes, `findface-extraction-api` uses neural networks.

If you want to manually initiate the installation of neural network models, use the console installer as follows:

1. Execute the `findface-*` file.

Note: Be sure to specify the actual file name instead of `findface-*`.

```
sudo ./findface-*.run
```

2. Product to install: `FindFace Multi`
3. Select the installation type: `Fully customized installation`.
4. Select a FindFace component to install: `findface-data`. To do so, first, deselect all the listed components by entering `*` in the command line, then select the required component by entering its sequence number (keyword). Enter `done` to save your selection and proceed to another step.
5. In the same manner, select models to install. After that, the installation process will automatically begin.

You can find installed models for the object and object attribute recognition at `/usr/share/findface-data/models/`. See *Neural Network Models*.

1.3.5 Fully Customized Installation

The FindFace developer-friendly *installer* provides you with a few installation options, including the fully customized installation. This option is mostly used when deploying FindFace in a highly distributed environment and requires a certain level of knowledge and experience.

To initiate the fully customized installation, do the following:

1. Download the installer file `findface-*.run`.
2. Put the `.run` file into some directory on the designated host (for example, `/home/username`).
3. From this directory, make the `.run` file executable.

Note: Be sure to specify the actual file name instead of `findface-*`.

```
chmod +x findface-*.run
```

4. Execute the `.run` file.

```
sudo ./findface-*.run
```

The installer will ask you a few questions and perform several automated checks to ensure that the host meets the system requirements. After filling out each prompt, press `Enter`. The questions and answers are the following:

1. Q: Which product should be installed?

A: 1


```
Which product should be installed?

1. [security] FindFace Multi
2. [server ] FindFace Server
3. [video-worker] FindFace Video Worker
4. [nvidia-drivers] NVIDIA CUDA drivers (installed automatically when you
↳install gpu-variant of the products above)

(default: security)
product> 1
```

2. Q: Please choose installation type:

A: 4

```
Please choose installation type:

- 1 [stand-alone ] Single Server
- 2 [multi-worker] Single Server, Multiple video workers
- 3 [repo        ] Don't install anything, just set up the APT repository
- 4 [custom      ] Fully customized installation

(default: stand-alone)
type> 4
```

3. Q: Found X interface(s). Which one should we announce as our external address?

A: Choose the interface that you are going to use as the instance IP address.

```
Found 1 interface(s). Which one should we announce as our external address?

- 1 [lo        ] 127.0.0.1
- 2 [ens3      ] 192.168.112.254

(default: 192.168.112.254)
ext_ip.advertised> 2
```

4. Q: Found X interface(s). Which one should we announce as our inter-service communication address?

A: Choose the interface for inter-service communication.

```
Found 1 interface(s). Which one should we announce as our inter-service
↳communication address?

- 1 [lo        ] 127.0.0.1
- 2 [ens3      ] 192.168.112.254

(default: 192.168.112.254)
inter_ip.advertised> 2
```

5. Q: Please select FindFace Multi components to install:

A: Choose FindFace components to install. By default, all components are subject to installation. You can leave it as is by entering done, or select specific components. Whenever you have to make a selection, first, deselect all the listed components by entering * in the command line, then select required components by

entering their sequence number (keyword), for example: 1 7 13, etc. Enter done to save your selection and proceed to another step.

```
Please select FindFace Multi components to install:

- 1 [v] findface-data      - Face recognition models
...
...

Enter keyword to select matching choices or -keyword to clear selection.
Enter "done" to save your selection and proceed to another step.
components> done
```

6. Specific questions related to the selected components: acceleration type, the required number of component instances, neural network models, etc. If you are experiencing a difficulty answering them, try to find an answer in this documentation, or submit your question to support@ntechlab.com.

1.3.6 Guide to Typical Multi-Host Deployment

This section is all about deploying FindFace in a multi-host environment.

Tip: If after having read this section, you still have questions, do not hesitate to contact our experts by support@ntechlab.com.

The reasons for deploying FindFace in a multi-host environment are the following:

- The necessity to distribute the video processing high load.
- The necessity to distribute the feature vector extraction high load.
- Large number of objects to search through, that requires implementation of a distributed object database.

Before you start the deployment, outline your system architecture, depending on its load and allotted resources (see [Requirements](#)). The most common distributed scheme is as follows:

- One principal server with the following components: `findface-ntls`, `findface-security`, `findface-sf-api`, `findface-video-manager`, `findface-upload`, `findface-video-worker`, `findface-extraction-api`, `findface-tarantool-server`, and third-parties.
- Several additional video processing servers with installed `findface-video-worker`.
- (If needed) Several additional extraction servers with installed `findface-extraction-api`.
- (If needed) Additional database servers with multiple Tarantool shards.

This section describes the most common distributed deployment. In high load systems, it may also be necessary to distribute the API processing (`findface-sf-api` and `findface-video-manager`) across several additional servers. This procedure requires a high level of expertise and some extra coding. Please do not hesitate to contact our experts for help (support@ntechlab.com).

To deploy FindFace in a multi-host environment, follow the steps below:

- *Deploy Principal Server*
- *Deploy Video Processing Servers*
- *Deploy Extraction Servers*

- *Distribute Load across Extraction Servers*
- *Deploy Additional Database Servers*
- *Configure Network*

Deploy Principal Server

To deploy the principal server as part of a distributed architecture, do the following:

1. On the designated physical server, *install* FindFace from installer as follows:
 - Product to install: FindFace Multi.
 - Installation type: Single server, multiple video workers. In this case, FindFace will be installed and configured to interact with additional remote `findface-video-worker` instances.
 - Type of the `findface-video-worker` acceleration (on the principal server): CPU or GPU, subject to your hardware configuration.
 - Type of the `findface-extraction-api` acceleration (on the principal server): CPU or GPU, subject to your hardware configuration.

After the installation is complete, the following output will be shown on the console:

```
#####
#                               #
#           Installation is complete           #
#####
- upload your license to http://192.168.0.5/#!/license/
- user interface: http://192.168.0.5/
  superuser:      admin
  password:       admin
  documentation:  http://192.168.0.5/doc/
```

2. Upload the FindFace license file via the main web interface `http://<Host_IP_address>/#!/license`. To access the web interface, use the provided superuser credentials.

Important: Do not disclose the superuser (Super Administrator) credentials to others. To administer the system, create a new user with the administrator privileges. Whatever the role, Super Administrator cannot be deprived of its rights.

3. Allow the licensable services to access the `findface-ntls` license server from any IP address, To do so, open the `/etc/findface-ntls.cfg` configuration file and set `listen = 0.0.0.0:3133`. Restart `findface-ntls`. service.

```
sudo vi /etc/findface-ntls.cfg

## Address to accept incoming client connections (IP:PORT)
## type:string env:CFG_LISTEN longopt:--listen
listen = 0.0.0.0:3133
```

```
sudo systemctl restart findface-ntls.service
```

4. Allow accessing the `findface-video-manager` service from any IP address. To do so, open the `/etc/findface-video-manager.conf` configuration file and set `listen: 0.0.0.0:18810` and `rpc:listen: 0.0.0.0:18811`. Restart the `findface-video-manager` service.

```
sudo vi /etc/findface-video-manager.conf
```

```
listen: 0.0.0.0:18810
...
rpc:
  listen: 0.0.0.0:18811
```

```
sudo systemctl restart findface-video-manager.service
```

Deploy Video Processing Servers

On an additional video processing server, install only a `findface-video-worker` instance following the *step-by-step instructions*. Answer the installer questions as follows:

- Product to install: FindFace Video Worker.
- Type of the `findface-video-worker` acceleration: CPU or GPU, subject to your hardware configuration.
- FindFace IP address: IP address of the principal server.

After that, the installation process will automatically begin. The answers will be saved to a file `/tmp/<findface-installer-*)>.json`. Use this file to install FindFace Video Worker on other hosts without having to answer the questions again, by executing:

```
sudo ./findface-multi-1.2-and-server-5.2.run -f /tmp/<findface-installer-*)>.
↪ json
```

Note: If `findface-ntls` and/or `findface-video-manager` are installed on a different host than that with `findface-security`, specify their IP addresses in the `/etc/findface-video-worker-gpu.ini` (`/etc/findface-video-worker-cpu.ini`) configuration file after the installation.

```
sudo vi /etc/findface-video-worker-cpu.ini
sudo vi /etc/findface-video-worker-gpu.ini
```

In the `ntls-addr` parameter, specify the `findface-ntls` host IP address.

```
ntls-addr=127.0.0.1:3133
```

In the `mgr-static` parameter, specify the `findface-video-manager` host IP address, which provides `findface-video-worker` with settings and the video stream list.

```
mgr-static=127.0.0.1:18811
```

Deploy Extraction Servers

On an additional extraction server, install only a `findface-extraction-api` instance from the console installer. Answer the installer questions as follows:

- Product to install: FindFace Multi.
- Installation type: Fully customized installation.
- FindFace components to install: `findface-extraction-api` and `findface-data`. To make a selection, first, deselect all the listed components by entering `-*` in the command line, then select `findface-extraction-api` and `findface-data` by entering their sequence number (keyword). Enter `done` to save your selection and proceed to another step.
- Type of `findface-extraction-api` acceleration: CPU or GPU.
- Modification of the `/etc/findface-extraction-api.ini` configuration file: specify the IP address of the `findface-ntls` server.
- Neural network models to install: CPU or GPU model for face biometrics (mandatory). Be sure to choose the right acceleration type for each model, matching the acceleration type of `findface-extraction-api`: CPU or GPU. Be aware that `findface-extraction-api` on CPU can work only with CPU-models, while `findface-extraction-api` on GPU supports both CPU- and GPU-models.

Tip: See *Neural Network Models* for details.

After that, the installation process will automatically begin. The answers will be saved to a file `/tmp/<findface-installer-*)>.json`. Use this file to install `findface-extraction-api` on other hosts without having to answer the questions again.

```
sudo ./findface-multi-1.2-and-server-5.2.run -f /tmp/<findface-installer-*)>.
↪ json
```

After all the extraction servers are deployed, distribute load across them by using a *load balancer*.

Distribute Load across Extraction Servers

To distribute load across several extraction servers, you need to set up load balancing. The following step-by-step instructions demonstrate how to set up `nginx` load balancing in a round-robin fashion for 3 `findface-extraction-api` instances located on different physical hosts: one on the FindFace principal server (192.168.0.5), and 2 on additional remote servers (192.168.0.6, 192.168.0.7). Should you have more extraction servers in your system, load-balance them by analogy.

Tip: You can use any load balancer according to your preference. Please refer to the relevant official documentation for guidance.

To set up load balancing, do the following:

1. Designate the FindFace principal server (recommended) or any other server with `nginx` as a gateway to all the extraction servers.

Important: You will have to specify the gateway server IP address when configuring the FindFace *network*.

Tip: You can install nginx as such:

```
sudo apt update
sudo apt install nginx
```

2. On the gateway server, create a new nginx configuration file.

```
sudo vi /etc/nginx/sites-available/extapi
```

3. Insert the following entry into the just created configuration file. In the `upstream` directive (`upstream extapibackends`), substitute the exemplary IP addresses with the actual IP addresses of the extraction servers. In the `server` directive, specify the gateway server listening port as `listen`. You will have to enter this port when configuring the FindFace *network*.

```
upstream extapibackends {
    server 192.168.0.5:18666; ## `findface-extraction-api` on principal server
    server 192.168.0.6:18666; ## 1st additional extraction server
    server 192.168.0.7:18666; ## 2nd additional extraction server
}
server {
    listen 18667;
    server_name extapi;
    client_max_body_size 64m;
    location / {
        proxy_pass http://extapibackends;
        proxy_next_upstream error;
    }
    access_log /var/log/nginx/extapi.access_log;
    error_log /var/log/nginx/extapi.error_log;
}
```

4. Enable the load balancer in nginx.

```
sudo ln -s /etc/nginx/sites-available/extapi /etc/nginx/sites-enabled/
```

5. Restart nginx.

```
sudo service nginx restart
```

6. On the principal server and each additional extraction server, open the `/etc/findface-extraction-api.ini` configuration file. Substitute `localhost` in the `listen` parameter with the relevant server address that you have specified in `upstream extapibackends` (`/etc/nginx/sites-available/extapi`) before. In our example, the address of the 1st additional extraction server has to be substituted as such:

```
sudo vi /etc/findface-extraction-api.ini

listen: 192.168.0.6:18666
```

7. Restart the `findface-extraction-api` on the principal server and each additional extraction server.

```
sudo systemctl restart findface-extraction-api.service
```

The load balancing is now successfully set up. Be sure to specify the actual gateway server IP address and listening port, when configuring the FindFace *network*.

Deploy Additional Database Servers

The `findface-tarantool-server` component connects the Tarantool-based feature vector database and the `findface-sf-api` component, transferring search results from the database to `findface-sf-api` for further processing.

To increase search speed, you can allocate several additional servers to the feature vector database and create multiple `findface-tarantool-server` shards on each additional server. The concurrent functioning of multiple shards will lead to a remarkable increase in performance, as each shard can handle up to approximately 10,000,000 feature vectors.

To deploy additional database servers, do the following:

1. Install the `findface-tarantool-server` component on the first designated server. Answer the installer questions as follows:
 - Product to install: `FindFace Multi`.
 - Installation type: `Fully customized installation`.
 - FindFace components to install: `findface-tarantool-server`. To make a selection, first, deselect all the listed components by entering `-*` in the command line, then select `findface-tarantool-server` by entering its sequence number (keyword). Enter `done` to save your selection and proceed to another step.

After that, the installation process will automatically begin.

As a result of the installation, the `findface-tarantool-server` shards will be automatically installed in the amount of $N = \min(\max(\min(\text{mem_mb} // 2000, \text{cpu_cores}), 1), 16 * \text{cpu_cores})$. I.e., it is equal to the RAM size in MB divided by 2000, or the number of CPU physical cores (but at least one shard), or the number of CPU physical cores multiplied by 16 if the first obtained value is greater.

2. Use the created `/tmp/<findface-installer-*)>.json` file to install `findface-tarantool-server` on other servers without answering the questions again. To do so, execute:

```
sudo ./findface-multi-1.2-and-server-5.2.run -f /tmp/<findface-installer-*)>.json
```

3. Be sure to specify the IP addresses and ports of the shards later on when configuring the FindFace *network*. To learn the port numbers, execute on each database server:

```
sudo cat /etc/tarantool/instances.enabled/*shard* | grep -E ".start|(listen =)"`
```

You will get the following result:

```
listen = '127.0.0.1:33001',
FindFace.start("127.0.0.1", 8101, {
listen = '127.0.0.1:33002',
FindFace.start("127.0.0.1", 8102, {
```

You can find the port number in the `FindFace.start` section, for example, 8101, 8102, etc.

Configure Network

After all the FindFace components are deployed, configure their interaction over the network. Do the following:

1. Open the `/etc/findface-sf-api.ini` configuration file:

```
sudo vi /etc/findface-sf-api.ini
```

Specify the following parameters:

Parameter	Description
extraction-api -> extraction-api	IP address and listening port of the <i>gateway extraction server</i> with set up load balancing.
storage-api -> shards -> master	IP address and port of the <code>findface-tarantool-server</code> master shard. Specify each shard by analogy.
upload_url	WebDAV NginX path to send original images, thumbnails and normalized object images to the <code>findface-upload</code> service.

```
...
extraction-api:
  extraction-api: http://192.168.0.5:18667
...
webdav:
  upload-url: http://127.0.0.1:3333/uploads/
...
storage-api:
  ...
  shards:
    - master: http://192.168.0.10:8101/v2/
      slave: ''
    - master: http://192.168.0.10:8102/v2/
      slave: ''
    - master: http://192.168.0.11:8101/v2/
      slave: ''
    - master: http://192.168.0.11:8102/v2/
      slave: ''
    - master: http://192.168.0.12:8101/v2/
      slave: ''
    - master: http://192.168.0.12:8102/v2/
      slave: ''
```

2. Open the `/etc/findface-security/config.py` configuration file.

```
sudo vi /etc/findface-security/config.py
```

Specify the following parameters:

Parameter	Description
EXTERNAL_ADDRESS	(Optional) IP address or URL that can be used to access the FindFace web interface. Once this parameter not specified, the system auto-detects it as the external IP address. To access FindFace, you can use both the auto-detected and specified IP addresses.
VIDEO_DETECTOR_TOKEN	For the video object detection module, come up with a token and specify it here.
VIDEO_MANAGER_ADDRESS	The findface-video-manager host.
NTLS_HTTP_URL	Address of the findface-ntls host.
ROUTER_URL	External IP address of the findface-security host that will receive detected objects from the findface-video-worker instance(s).
SF_API_ADDRESS	Address of the findface-sf-api host.

```

sudo vi /etc/findface-security/config.py

...
# SERVICE_EXTERNAL_ADDRESS prioritized for webhooks and genetec
SERVICE_EXTERNAL_ADDRESS = 'http://localhost'
EXTERNAL_ADDRESS = 'http://127.0.0.1'

...
FFSECURITY = {
    'VIDEO_DETECTOR_TOKEN': '7ce2679adfc4d74edcf508bea4d67208',
    ...
    'VIDEO_MANAGER_ADDRESS': 'http://127.0.0.1:18810',
    ...
    'NTLS_HTTP_URL': 'http://127.0.0.1:3185',
    'ROUTER_URL': 'http://192.168.0.5',
    ...
    'SF_API_ADDRESS': 'http://127.0.0.1:18411',
    ...
}

```

The FindFace components interaction is now set up.

Important: To preserve the FindFace compatibility with the installation environment, we highly recommend you to disable the Ubuntu automatic update. In this case, you will be able to update your OS manually, fully controlling which packages to update.

To disable the Ubuntu automatic update, execute the following commands:

```

sudo apt-get remove unattended-upgrades
sudo systemctl stop apt-daily.timer
sudo systemctl disable apt-daily.timer
sudo systemctl disable apt-daily.service
sudo systemctl daemon-reload

```

Important: The FindFace services log a large amount of data, which can eventually lead to disc overload. To prevent this from happening, we advise you to disable rsyslog due to its suboptimal log rotation scheme and use the appropriately configured systemd-journal service instead. See *Service Logs* for the step-by-step instructions.

1.3.7 Add NVIDIA Repository and Install Drivers (GPU only)

FindFace on GPU requires the prior installation of NVIDIA drivers.

To add the NVIDIA repository and install the drivers, do the following:

Important: You will need a stable Internet connection, as the driver packages will be downloaded from the NVIDIA web resource.

1. Download the installer file `<findface-*>.run`.
2. Put the `.run` file into some directory on the designated host (for example, `/home/username`).
3. From this directory, make the `.run` file executable.

```
chmod +x <findface-*>.run
```

4. Execute the `.run` file.

```
sudo ./<findface-*>.run
```

5. Choose the product to install: `NVIDIA CUDA drivers`.
6. After the NVIDIA drivers installation is complete, restart the server.

1.3.8 Remove FindFace Instance

You can automatically remove FindFace along with the *data storages* by using the `findface_uninstall.sh` script. The FindFace configuration files and data storages will be backed up if you request it.

Important: Regarding directories with the FindFace *artifacts*, the script will back up and clean only `/var/lib/findface-security/uploads` and `/var/lib/ffupload/` (set by default). If you modified the artifacts' location, you will have to create a backup and purge them by yourself. See *Back Up and Recover FindFace and Data* for reference.

Do the following:

1. Download the `findface_uninstall.sh` script to some directory on a designated host (for example, to `/home/username/`).
2. From this directory, make the script executable.

```
sudo chmod +x findface_uninstall.sh
```

3. Run the script.

```
sudo ./findface_uninstall.sh
```

4. Answer **backup** to create a backup and then remove FindFace along with the data storages. Answer **all** to completely wipe out FindFace and the data storages without a backup.
5. The script purges all content from the `/var/lib/ffupload` directory. However, the directory itself will remain in your file system. Remove `/var/lib/ffupload` manually by executing:

```
sudo rmdir /var/lib/ffupload
```

Important: To recover FindFace from the backup, see *Back Up and Recover FindFace and Data*.

1.4 Administration and Configuration

1.4.1 Licensing

In this chapter:

- *Licensing Principles*
- *View and Update License*
- *Offline Licensing via Hardware Fingerprint*

Licensing Principles

The FindFace licensing is granted using the following criteria:

1. The overall number of extracted feature vectors.

Note: The feature vectors are extracted from objects detected in the video, from images in the record index, and user photos, and when building so-called cluster centroids.

The licensing scheme is the following:

- Events: 1 event of video object detection = 1 object in a license.
 - Record Index: 1 photo in a record = 1 object in a license.
 - Clusters: 1 person = 1 object in a license.
 - Users: 1 photo of a user = 1 object in a license.
2. The number of video sources currently in use (i.e., active video processing jobs for video files).
 3. The number of model instances in use in the `findface-extraction-api` component.

You can choose between the following licensing methods:

- The online licensing is provided by interaction with the NtechLab Global License Manager `license.ntechlab.com` and requires a stable internet connection, DNS, and open port 443 TCP. Upon being disconnected from the internet, the system will continue working off-grid for 4 hours.

Note: It is possible to prolongate the off-grid period for up to 2 days. Inform your manager if you need that.

- The offline licensing via a USB dongle requires a USB port on the physical server with the `findface-ntls` component (license server in the *FindFace core*).
- The offline licensing via hardware fingerprint requires Sentinel drivers installed on the physical server with the `findface-ntls` component.

Important: For the system to function, a single instance of `findface-ntls` should be enough. If your system requires more license servers, contact your NtechLab manager beforehand to prevent your system from being blocked.

View and Update License

After installing FindFace, upload the license file you obtained from the manager into the system. To do so, navigate to *Settings* -> *License*.

Preferences	Common	Limits	Services
General	Valid	Yes	
Roles	Type of license	online	
Users	License ID	ab9aae2c48034a65b18cc5c5b87f2a98	
Sessions	File	/opt/ntech/license/import_60362836b7afa9911a168683b4e8e7e89d2abeb2957a4c70e2b4901d085a7efc.lic	
Blocklist records	Generated	18.07.2022, 18:22:39	
Camera groups	Last updated	1 seconds ago (09.09.2022, 15:38:17)	
Watch lists			
License			
Documentation			
API documentation			

Use the same tab to consult current licensing information and upgrade your license.

Offline Licensing via Hardware Fingerprint

Note: Sentinel is a type of offline licenses that, unlike guardant licenses, do not require any physical media for its work.

Glossary:

- Sentinel is a software protection and licensing system by [Thales](#). It allows you to implement offline licensing without access to a global server.
- The C2V file is a file, containing data about a hardware fingerprint of the client's machine, for binding the license only to this machine. This file is generated by the sentinel library. The C2V file is generated on the client's machine where the license key will be installed later.

To implement the fingerprint licensing to your system, do the following:

1. Inform your manager that you intend to apply this licensing method and request your unique license id.
2. Install the Sentinel drivers on the physical server with the `findface-ntls` component.

Do the following:

1. Download [Sentinel drivers](#) from the official website.
2. Unzip the downloaded archive and browse to it.

```
tar -xvzf Sentinel_LDK_Linux_Runtime_Installer_script.tar.gz
cd Sentinel_LDK_Linux_Runtime_Installer_script/
```

3. There is another archive `aksusbd-8.31.1.tar.gz` inside the archive. Unzip it and browse to the resulting directory.

```
tar -xvzf aksusbd-8.31.1.tar.gz
cd aksusbd-8.31.1/
```

4. Run the installation command.

```
sudo ./dinst
```

5. Run and check the statuses of the Sentinel services.

```
sudo systemctl start aksusbd.service hasplmd.service
sudo systemctl status aksusbd.service hasplmd.service
```

3. Put the `findface-sentinel-lib_*.deb` package received from your manager into some directory on the same host. Install the package.

```
sudo dpkg -i /path/to/findface-sentinel-lib_*.deb
```

4. In the FindFace web interface, navigate to *Preferences* -> *License*. Take a hardware fingerprint (C2V file) by clicking the *Download C2V for activation* button.

Tip: If you prefer working with the console, you can send the following API request to `findface-ntls` instead:

```
curl 'http://<findface-ntls-server-ip>/ntls/c2v' >my_pc.c2v
```

5. Send the License ID and the C2V file to your manager and receive your license file in return.
6. Upload the license file on the *License* tab.

1.4.2 General Settings

The FindFace general settings, such as generic confidence thresholds for face recognition and thumbnail JPEG quality, determine your system functioning and resource consumption.

To configure the general settings, navigate *Settings* -> *General*. After you are finished with adjustments, click *Update*. Find the detailed explanation of each general setting below.

Preferences

General

Roles

Users

Sessions

Blocklist records

Camera groups

Watch lists

License

Documentation

API documentation

Faces

Generic similarity threshold

 0.698

Thumbnail JPEG quality

 %

Update

In this section:

- *Generic Confidence Threshold*
- *Thumbnail JPEG Quality*

Generic Confidence Threshold

FindFace verifies that selected faces belong to the same person (i.e. match), based on the pre-defined similarity threshold. The default threshold is set to the optimum value. If necessary, you can change it.

Note: The higher is the threshold, the less are chances that a wrong person will be positively verified, however, some valid photos may also fail verification.

Tip: You can configure the confidence thresholds individually for each *camera group* and *watch list*.

Important: The default generic confidence threshold is optimal for the majority of recognition cases. We do not recommend changing it on your own. Be sure to consult with our technical experts prior (support@ntechlab.com).

Thumbnail JPEG Quality

Subject to JPEG quality, thumbnails may take up a significant amount of disc volume. Use the *General* tab to configure the parameter.

1.4.3 Role and User Management

In this chapter:

- *Predefined Roles*
- *Create Custom Role*
- *Primary and Additional User Privileges*
- *Create User Account Manually*
- *Integrate with Active Directory for Auto User Creation*
 - *Install and Configure Kerberos*
 - *Generate Keytab File*
 - *Rebuild NGINX on FindFace Server to Support Kerberos*
 - *Finalize FindFace Configuration*
 - *Manage FindFace Users via Active Directory*
- *Deactivate or Delete Users*

Predefined Roles

FindFace provides the following predefined roles:

- Administrator is granted full access to the FindFace functionality, integrative and administrative tools.

Important: Whatever the role, the first administrator (Super Administrator) cannot be deprived of its rights.

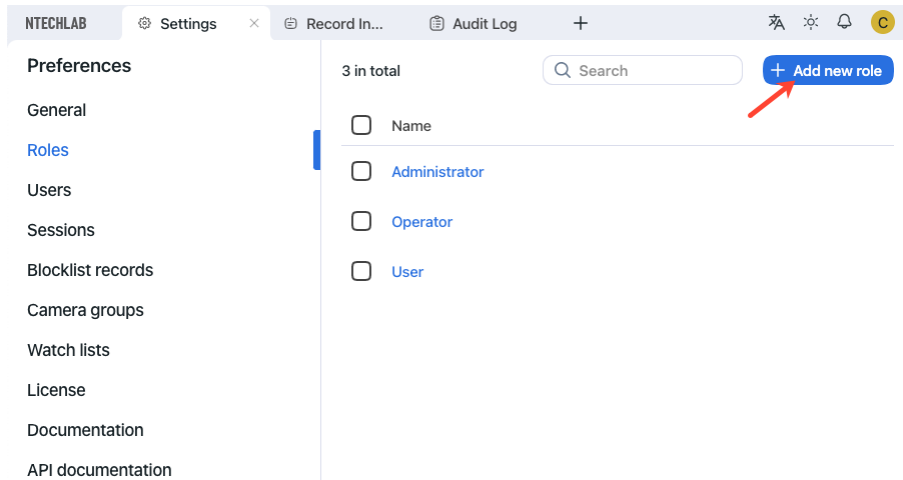
- Operator is granted full access to the FindFace functionality.
- User is granted rights to modify their profile and manage cases. The other functions are available read-only.

You can change the predefined roles privileges, as well as create various custom roles.

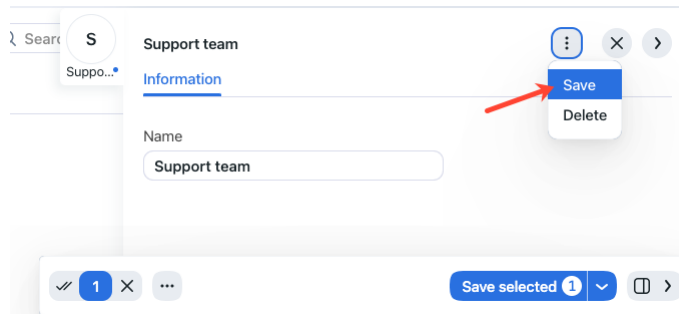
Create Custom Role

To create a custom role, do the following:

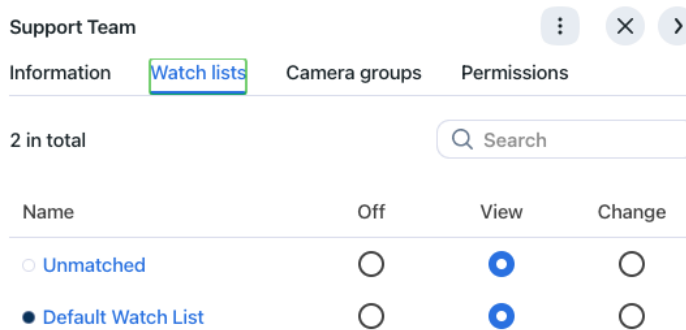
1. Navigate *Settings* -> *Roles*.
2. Click + *Add new role*.



3. On the *Information* tab, specify the role name. Save the role.



4. After saving the role, you will see the following tabs appear next to the *Information* tab:



- *Watch Lists*: role privileges for specific watch lists
- *Camera Groups*: role privileges for specific camera groups
- *Permissions*: role privileges for entire system functions and entities

Set role privileges, subject to your needs. Note that there is a distinction between role privileges for a specific watch list/camera group and a system entity with the name `watchlist/cameragroup`. For example, if you set Off for a certain camera group on the *Camera Groups* tab, users with this role won't be able to work with **this** very group of cameras. Unchecking all checkboxes for the `cameragroup` entity on the *Permissions* tab will prevent users from viewing and working with **all** camera groups.

The full list of the FindFace entities is as follows:

- case: case file
- caseattachment: uploading attachments to a case
- faceobject: face photo in a *record*
- deviceblacklistrecord: *blocklist*
- watchlist: *watch list*
- cameragroup: *camera group*
- uploadlist: list of photos in *bulk upload*
- upload: item (photo) in bulk record upload
- user: *user*
- report: *report*
- all_own_sessions: all *sessions* of the current user on different devices

Note: If relevant permissions for this entity are set, users will be able to view (**view**) and close (**delete**) all their sessions on different devices. Otherwise, users will be only allowed to view and close their session on the current device. Working with sessions takes place on the *Sessions* tab (*Settings*).

- humancard: *record of an individual*

You can also enable and disable rights for the following functionality:

- configure_ntls: configuration of the `findface-ntls` *license server*
- batchupload_cards: *bulk record upload*
- view_runtimeetting: viewing the FindFace *general preferences*
- change_runtimeetting: changing the FindFace general preferences
- view_auditlog: viewing and working with the *audit logs*.

5. Save the changes.

Primary and Additional User Privileges

You assign privileges to a user by using roles:

- *Primary role*: main user role, mandatory for assignment. You can assign only one primary role to a user.
- *Role*: additional user role, optional for assignment. You can assign several roles to one user. The rights associated with the additional roles will be added to the primary privileges.

All users belonging to a particular primary role automatically get access to camera groups (and video archives within the group) and watch lists (and records in the watch list) created by a user with the same primary role, subject to the privileges defined by their additional role(s).

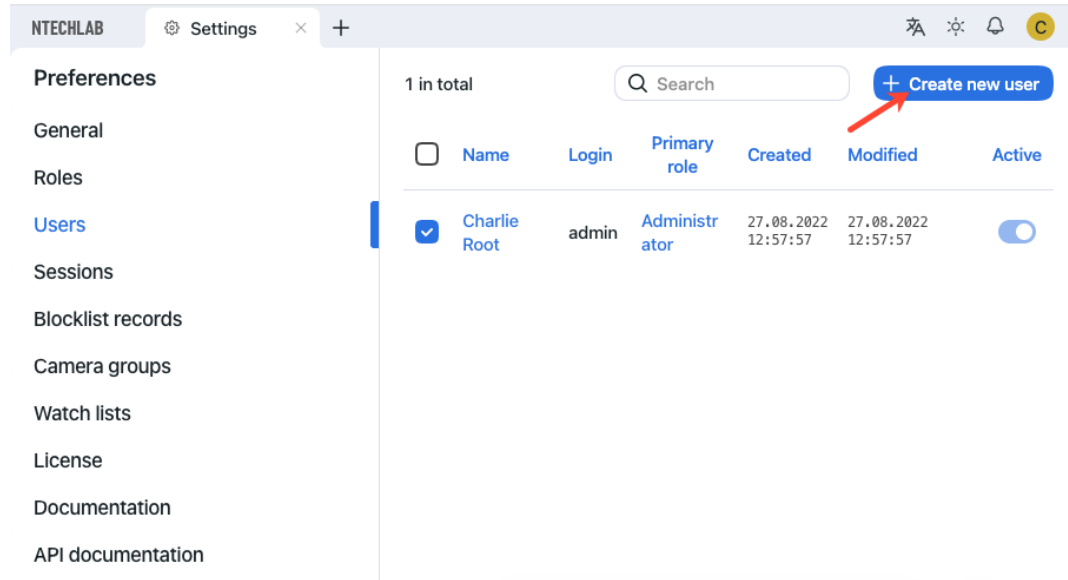
See also:

Create User Account Manually

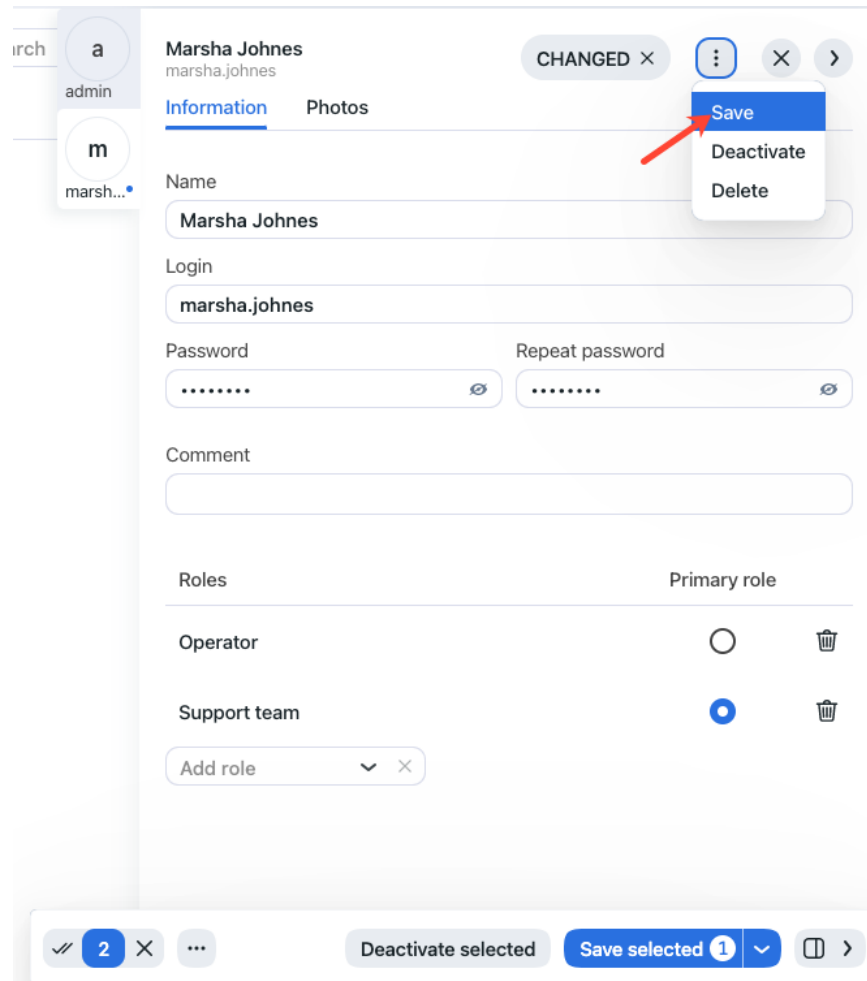
Create User Account Manually

To create a user account manually, do the following:

1. Navigate *Settings* -> *Users*.
2. Click + *Create new user*.



3. On the *Information* tab, specify user data such as name, login, and password. If necessary, add a comment.
4. From the *Roles* drop-down menu, select one or several user roles. Set one of them as the *Primary role*.
5. On the *Photos* tab, attach the user's photo.
6. Save the user account.



Integrate with Active Directory for Auto User Creation

If there are many users in FindFace, it can be inconvenient to create their accounts one by one. One of the ways to facilitate the user creation is to harness the FindFace integration with Active Directory. To do so, follow the step-by-step instructions below, minding the sequence.

Install and Configure Kerberos

First of all, install and configure the Kerberos authentication protocol on the FindFace principal server. Do the following:

1. Install the `krb5-kdc` package.

```
sudo apt-get install krb5-kdc
```

Important: During the installation, you will have to enter the realm name. It must be equal to the Active Directory domain name, but spelled in upper case (`TESTNTL.LOCAL` in the example below). It's ok to skip all other installation windows by pressing `Enter`.

2. Find the realms section in the Kerberos configuration file `/etc/krb5.conf`. Specify the Active Directory domain in it.

```
sudo vi /etc/krb5.conf

[realms]
TESTNTL.LOCAL = {
    kdc = testntl.local
    default_domain = testntl.local
}
...
```

3. Append the following string to the `/etc/hosts` file: `<Active Directory server IP address> <Active Directory domain name>`.

```
sudo vi /etc/hosts

...
192.168.0.5 testntl.local
```

Generate Keytab File

Log-in into the Active Directory server and do the following:

1. Create a new user account in the Active Directory domain to use as a service account.

Do the following:

1. Open Active Directory. Click *Start*, point to *Administrative Tools*, and then click *Active Directory Users and Computers*.
 2. Click the domain name and then expand the contents. Right-click *Users*, point to *New*, and then click *User*. You will see a user creation form.
 3. Fill-in the fields in the form at your discretion. On the second tab, check the *Password never expires* checkbox.
 4. Click *Next*. Review the information that you provided, and if everything is correct, click *Finish*.
 5. Right-click the just created user account, and then navigate *Properties -> Member Of -> Add*.
 6. In the *Select Groups* dialog box, add the *Domain Administrators* and *Domain Users* groups to the list, and then click *OK*.
 7. Click *OK* to finish.
2. Register a Service Principal Name (SPN) for the service account that you created. To do so, open PowerShell as administrator and execute the following command, specifying your actual SERVICE USER NAME and domain:

```
setspn -A HTTP/<SERVICE USER NAME>.testntl.local@TESTNTL.LOCAL <SERVICE USER NAME>
```

3. In the same PowerShell window, generate a Keytab file by executing the command below with your actual SERVICE USER NAME, domain, and desirable KEYTAB FILE NAME.

```
ktpass.exe -princ HTTP/<SERVICE USER NAME>.testntl.local@TESTNTL.LOCAL -mapuser
↵<SERVICE USER NAME> -crypto ALL -ptype KRB5_NT_PRINCIPAL -pass * -out c:\<KEYTAB_
↵FILE NAME>.keytab
```

To check the result, navigate to the root directory of the C drive. You will see a keytab file with the relevant name.

4. Move the keytab file that you created to the FindFace server.
5. Check the keytab file on the FindFace server. To do so, execute the following command on the console, specifying your actual SERVICE USER NAME, domain, and KEYTAB FILE NAME.

```
kinit -5 -V -k -t <path/to/<KEYTAB FILE NAME>.keytab> HTTP/<SERVICE USER NAME>.
↪testntl.local
```

If everything is alright, you will see the message `Authenticated to Kerberos v5`.

Rebuild NGINX on FindFace Server to Support Kerberos

To successfully establish a link between FindFace and Active Directory, you need to enable the Kerberos support in NGINX installed on the FindFace principal server. It can be done by rebuilding NGINX with a third-party module `spnego-http-auth-nginx-module`.

Important: To download `spnego-http-auth-nginx-module`, you will need Git.

Do the following:

1. Download the NGINX source code of the same version as in FindFace. It's currently `nginx-1.14.0`, click [here](#) to download.

2. Unzip the downloaded archive.

```
tar -xf nginx_1.14.0.orig.tar.gz
```

3. Browse to the resulting directory. Clone the `spnego-http-auth-nginx-module` module into it.

```
git clone https://github.com/stnoonan/spnego-http-auth-nginx-module
```

4. Install an auxiliary package `libkrb5-dev`, essential for the `spnego-http-auth-nginx-module` work.

```
sudo apt-get install -y libkrb5-dev
```

5. Install the building toolset.

```
sudo apt-get install build-essential
```

6. Install a set of packages, essential for NGINX rebuilding.

```
sudo apt-get install libpcre3 libpcre3-dev openssl libssl-dev zlib1g zlib1g-dev
↪libxslt-dev libgd-dev libgeoip-dev
```

7. On the console, execute the following command and copy somewhere the argument list that will appear in the output (everything that goes after `configure` arguments).

```
nginx -V

nginx version: nginx/1.14.0 (Ubuntu)
built with OpenSSL 1.1.1 11 Sep 2018
TLS SNI support enabled
configure arguments: --with-cc-opt='-g -O2 -fdebug-prefix-map=/build/nginx-KgqPmI/
```

(continues on next page)

(continued from previous page)

```

↪nginx-1.14.0=. -fstack-protector-strong -Wformat -Werror=format-security -fPIC -
↪Wdate-time -D_FORTIFY_SOURCE=2' --with-ld-opt='-Wl,-Bsymbolic-functions -Wl,-z,
↪relro -Wl,-z,now -fPIC' --prefix=/usr/share/nginx --conf-path=/etc/nginx/nginx.
↪conf --http-log-path=/var/log/nginx/access.log --error-log-path=/var/log/nginx/
↪error.log --lock-path=/var/lock/nginx.lock --pid-path=/run/nginx.pid --modules-
↪path=/usr/lib/nginx/modules --http-client-body-temp-path=/var/lib/nginx/body --
↪http-fastcgi-temp-path=/var/lib/nginx/fastcgi --http-proxy-temp-path=/var/lib/
↪nginx/proxy --http-scgi-temp-path=/var/lib/nginx/scgi --http-uwsgi-temp-path=/var/
↪lib/nginx/uwsgi --with-debug --with-pcre-jit --with-http_ssl_module --with-http_
↪stub_status_module --with-http_realip_module --with-http_auth_request_module --
↪with-http_v2_module --with-http_dav_module --with-http_slice_module --with-
↪threads --with-http_addition_module --with-http_geoip_module=dynamic --with-http_
↪gunzip_module --with-http_gzip_static_module --with-http_image_filter_
↪module=dynamic --with-http_sub_module --with-http_xslt_module=dynamic --with-
↪stream=dynamic --with-stream_ssl_module --with-mail=dynamic --with-mail_ssl_module

```

8. Add the `spnego-http-auth-nginx-module` module to the rebuilding components. To do so, reconfigure NGINX by invoking the `configure` utility with the `--add-dynamic-module=spnego-http-auth-nginx-module` option placed before the argument list.

Briefly:

```
sudo ./configure --add-dynamic-module=spnego-http-auth-nginx-module <argument list>
```

Example:

```

sudo ./configure --add-dynamic-module=spnego-http-auth-nginx-module --with-cc-opt='-
↪g -O2 -fdebug-prefix-map=/build/nginx-KgqPmI/nginx-1.14.0=. -fstack-protector-
↪strong -Wformat -Werror=format-security -fPIC -Wdate-time -D_FORTIFY_SOURCE=2' --
↪with-ld-opt='-Wl,-Bsymbolic-functions -Wl,-z,relro -Wl,-z,now -fPIC' --prefix=/
↪usr/share/nginx --conf-path=/etc/nginx/nginx.conf --http-log-path=/var/log/nginx/
↪access.log --error-log-path=/var/log/nginx/error.log --lock-path=/var/lock/nginx.
↪lock --pid-path=/run/nginx.pid --modules-path=/usr/lib/nginx/modules --http-
↪client-body-temp-path=/var/lib/nginx/body --http-fastcgi-temp-path=/var/lib/nginx/
↪fastcgi --http-proxy-temp-path=/var/lib/nginx/proxy --http-scgi-temp-path=/var/
↪lib/nginx/scgi --http-uwsgi-temp-path=/var/lib/nginx/uwsgi --with-debug --with-
↪pcre-jit --with-http_ssl_module --with-http_stub_status_module --with-http_realip_
↪module --with-http_auth_request_module --with-http_v2_module --with-http_dav_
↪module --with-http_slice_module --with-threads --with-http_addition_module --with-
↪http_geoip_module=dynamic --with-http_gunzip_module --with-http_gzip_static_
↪module --with-http_image_filter_module=dynamic --with-http_sub_module --with-http_
↪xslt_module=dynamic --with-stream=dynamic --with-stream_ssl_module --with-
↪mail=dynamic --with-mail_ssl_module

```

9. Wait until the NGINX reconfiguration is completed and initiate NGINX rebuilding by executing the following commands:

```

sudo make

sudo make install

```

A new file `/usr/lib/nginx/modules/nginx_http_auth_spnego_module.so` will be created as a result.

10. In the `/etc/nginx/modules-enabled/` directory, create a new configuration file `spnego-http-auth-nginx-module.conf` with a string `load_module '/usr/lib/nginx/modules/nginx_http_auth_spnego_module.so'`; inside.

```
sudo vi spnego-http-auth-nginx-module.conf

load_module '/usr/lib/nginx/modules/nginx_http_auth_spnego_module.so';
```

11. Restart NGINX.

```
sudo systemctl reload nginx
```

12. Open the `/etc/nginx/sites-available/ffsecurity-nginx.conf` configuration file. Find the location `/users/me/ad` section and uncomment it. Fill in the section by analogy with the example below, placing your actual variables in the strings with comments (`#`).

The variables to specify are the following:

- `auth_gss_realm`: realm name in Kerberos
- `auth_gss_keytab`: location of the keytab file on the FindFace Server
- `auth_gss_service_name`: full service user name in Active Directory, including the name of the domain it belongs to

```
sudo vi /etc/nginx/sites-available/ffsecurity-nginx.conf

location /users/me/ad {
    proxy_pass http://192.168.0.3/auth/ad_login/; # e.g http://127.0.0.1/auth/ad_
    ↪login/;
    proxy_method POST;

    proxy_set_header    X-Real-IP $remote_addr;
    proxy_set_header    Host $http_host;
    proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header    Authorization $http_authorization;
    proxy_pass_header    Authorization;
    proxy_no_cache 1;
    proxy_cache_bypass 1;

    auth_gss on;
    auth_gss_realm TESTNTL.LOCAL; # Realm name in Kerberos;
    auth_gss_keytab /home/ubuntu/<KEYTAB FILE NAME>.keytab; # e.g. /var/lib/web.
    ↪keytab
    auth_gss_service_name HTTP/<service_user>.testntl.local; # e.g. HTTP/web.
    ↪testntl.local;
    auth_gss_allow_basic_fallback on;
}
```

13. Restart NGINX once again.

```
sudo systemctl reload nginx
```

Finalize FindFace Configuration

To finalize the FindFace integration with Active Directory, perform the following configuration steps on the FindFace side:

1. Open the `/etc/findface-security/config.py` configuration file.

```
sudo vi /etc/findface-security/config.py
```

2. In the `SERVICES` section, set `"active_directory": True`.

```
SERVICES = {  
    ...  
    "active_directory": True,  
    ...  
}  
}
```

3. Fill in the `ACTIVE_DIRECTORY_CONFIG` section as follows:

- `AUTH_LDAP_SERVER_URI`: ldap: <Active Directory server IP address>
- `AUTH_LDAP_BIND_DN`: the name of the service user that you created in Active Directory
- `AUTH_LDAP_BIND_PASSWORD`: the service user password
- `SEARCH_GROUPS`: Active Directory organization units which FindFace will search for user accounts

```
# Specify server credentials  
ACTIVE_DIRECTORY_CONFIG = {  
    'AUTH_LDAP_SERVER_URI': 'ldap://192.168.0.5',  
    # Domain Administrator user  
    'AUTH_LDAP_BIND_DN': '<SERVICE USER NAME IN ACTIVE DIRECTORY>',  
    # Domain Administrator user password  
    'AUTH_LDAP_BIND_PASSWORD': 'SERVICE USER NAME PASSWORD',  
    # Specify organization units where users search will be executed.  
    # Follow pattern (e.g. OU=DEV,DC=domain,DC=com)  
    'SEARCH_GROUPS': 'OU=DEV,DC=testnt1,DC=local',  
}
```

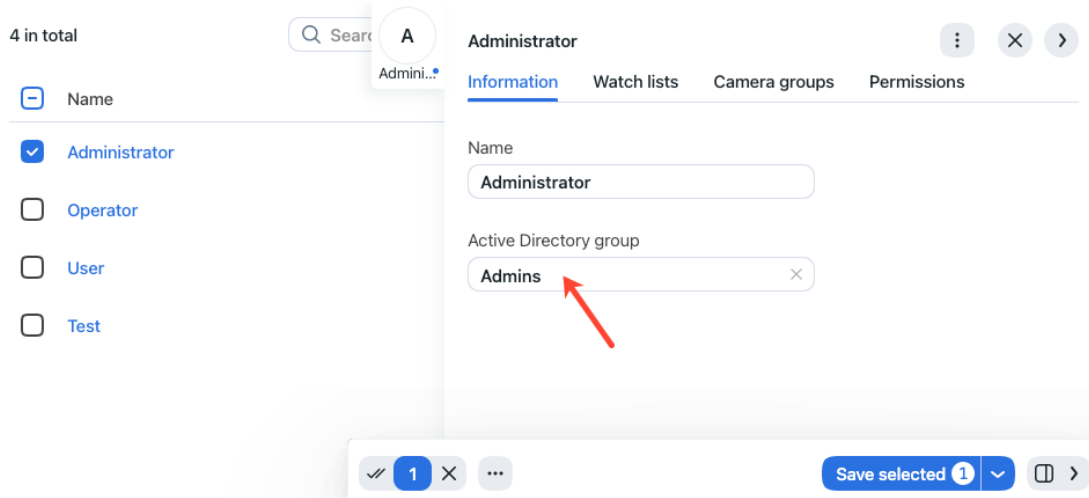
4. Restart the `findface-security` service.

```
sudo systemctl restart findface-security.service
```

Note: Check the output. The list of services should feature the LDAP Server.

Manage FindFace Users via Active Directory

If the FindFace integration with Active Directory is enabled, you will be able to set one of the Active Directory groups for a role you are creating or editing.

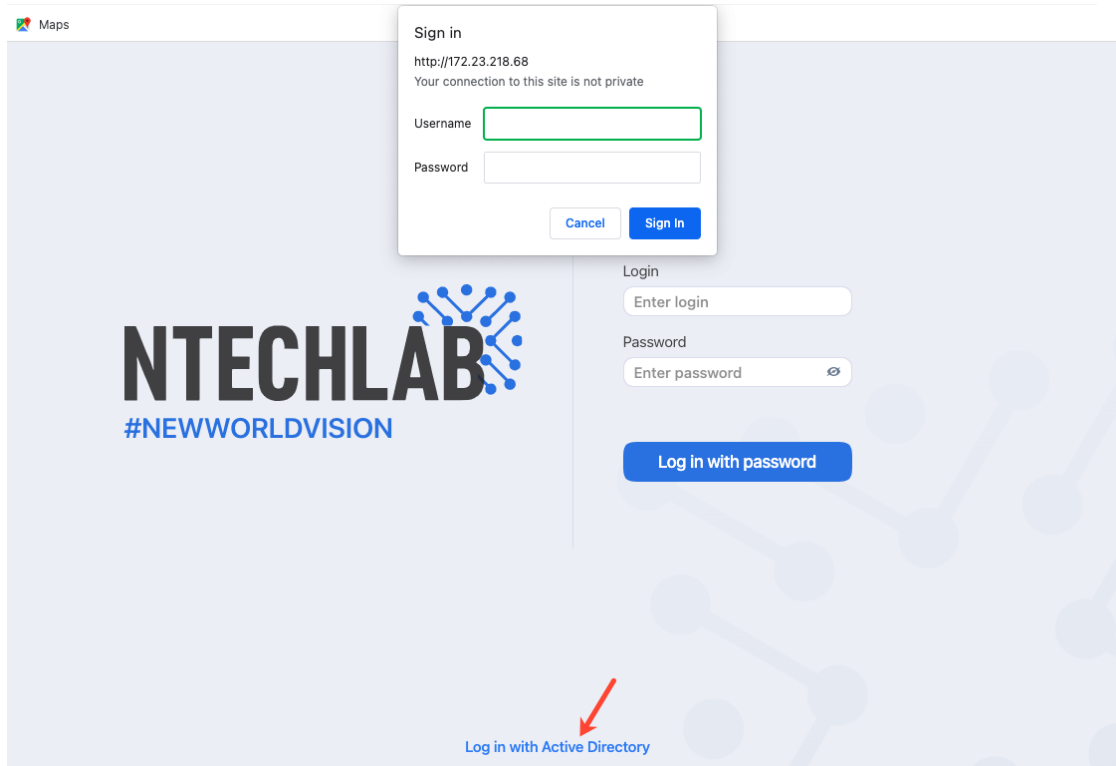


Once a user from the selected Active Directory group logs-in into FindFace for the first time, they will be automatically added to the FindFace user list.

The screenshot shows the FindFace user list with two users. The 'Active Directory' column for the first user is 'Yes', indicated by a red arrow.

Name	Login	Primary role	Active Directory	Created	Modified	Active
[blurred]	[blurred]	User	Yes	01.08.2022 17:38:46	01.08.2022 17:38:46	<input checked="" type="checkbox"/>
Charlie Root	admin	Administrator		07.06.2022 13:47:10	07.06.2022 13:47:10	<input type="checkbox"/>

To log-in with Active Directory, a user must click the *Log in with Active Directory* button in the authentication window, specify their Active Directory credentials, and click *Sign in*.



Deactivate or Delete Users

In order to deactivate a user, unset *Active* on the user list (*Settings* -> *Users*).

If you are going to deactivate multiple users, select them on the user list and then click *Deactivate selected*.

To delete users from FindFace, select them on the user list and then click *Delete selected*.

1.4.4 List of User Sessions. Blocklist

In this chapter:

- *Grant Permissions to Work with Sessions*
- *View User Sessions*
- *Block Device*

FindFace allows you to monitor user sessions and learn associated data, such as the connected device UUID, type of user interface, IP address, last ping time, and so on.

If necessary, you can add a device to the blocklist without deactivating the user account. The device block may come in handy in various situations. For example, if you want users to access the system only from their workplaces. Use the blocklist functionality to take your system safety to the next level.

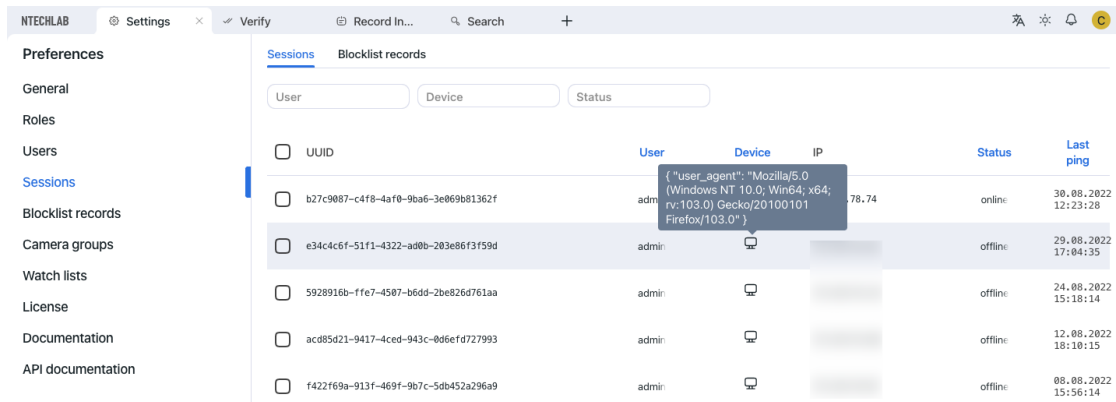
Grant Permissions to Work with Sessions

A user's access to the list of sessions depends on the granted *permissions*:

- Administrator: can view and close sessions of all users
- User with the `all_own_sessions` permissions: can view/close all sessions initiated with their username
- User without the `all_own_sessions` permissions: can only view/close their current session

View User Sessions

To view the list of user sessions, navigate *Settings* -> *Sessions*.



UUID	User	Device	IP	Status	Last ping
<input type="checkbox"/> b27c9087-c4f8-4af0-9ba6-3e0e9b81362f	admin	Desktop	192.168.1.78.74	online	30.08.2022 12:23:28
<input type="checkbox"/> e34c4c6f-51f1-4322-ad0b-203e86f3f59d	admin	Mobile		offline	29.08.2022 17:04:35
<input type="checkbox"/> 5928916b-ffe7-4507-b6d9-2be826d761aa	admin	Mobile		offline	24.08.2022 15:18:14
<input type="checkbox"/> acd85d21-9417-4ced-943c-006ef727993	admin	Mobile		offline	12.08.2022 18:10:15
<input type="checkbox"/> f422f69a-913f-469f-9b7c-5db452a296a9	admin	Mobile		offline	08.08.2022 15:56:14

Each session record provides the following data:

- device UUID
- username
- type of the user interface (mobile/web)
- device information
- IP address
- status (online, offline, blocked)
- last ping time

Use the filter panel above the list of sessions to set up the search conditions.

To close a session, select it in the list and click *x*.

Sessions Blocklist records

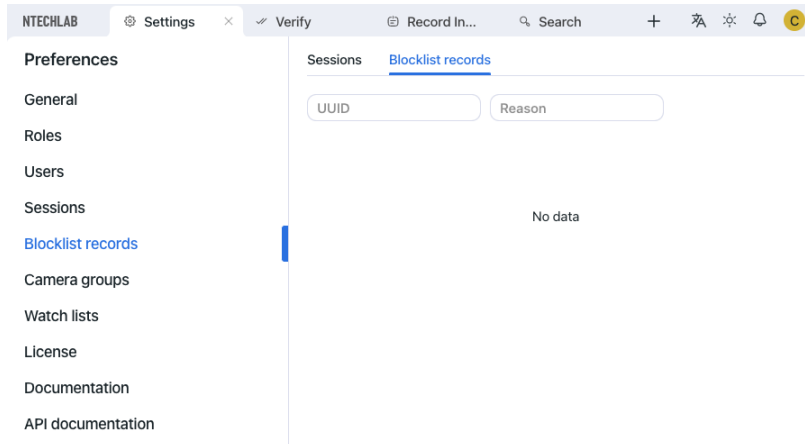
User Device Status

UUID	User	Device	IP	Status	Last ping
<input type="checkbox"/> b27c9087-c4f8-4af0-9ba6-3e069b81362f	admin		192.168.1.1	online	30.08.2022 12:23:28
<input checked="" type="checkbox"/> e34c4c6f-51f1-4322-ad0b-203e86f3f59d	admin		192.168.1.1	offline	29.08.2022 17:04:35
<input checked="" type="checkbox"/> 5928916b-ffe7-4507-b6dd-2be826d761aa	admin		192.168.1.1	offline	24.08.2022 15:18:14
<input checked="" type="checkbox"/> acd85d21-9417-4ced-943c-0d6efd727993	admin		192.168.1.1	offline	12.08.2022 18:10:15
<input checked="" type="checkbox"/> f422f69a-913f-469f-9b7c-5db452a296a9	admin		192.168.1.1	offline	08.08.2022 15:56:14
<input checked="" type="checkbox"/> 93c8c7bf-6aa7-4a8b-98be-afb21e542309	admin		192.168.1.1	offline	05.08.2022 16:26:04
<input checked="" type="checkbox"/> 09b30e09-6e00-4d41-8ec5-cc90be5ea69d	admin		192.168.1.1	offline	03.08.2022 20:27:04
<input checked="" type="checkbox"/> 710b4708-6ac9-4c9e-abe3-96566329efbc	admin		192.168.1.1	offline	03.08.2022 19:35:00
<input type="checkbox"/> 3f5abddd-bd4e-4cbe-9ff5-a9b7868c7707	admin		192.168.1.1	offline	03.08.2022 19:35:00

7 X Block

Block Device

The list of blocked devices is available on the *Blocklist Records* tab.



You can add a device to the blocklist on the *Sessions* tab. Blocking a device leads to the user's automatic log-out.

To block a device, do the following:

1. Select the relevant session record(s).
2. Click *Block*.

Sessions Blocklist records

User Device Status

<input type="checkbox"/>	UUID	User	Device	IP	Status	Last ping
<input type="checkbox"/>	b27c9087-c4f8-4af0-9ba6-3e069b81362f	admin		...	online	30.08.2022 12:23:28
<input checked="" type="checkbox"/>	e34c4c6f-51f1-4322-ad0b-203e86f3f59d	admin		...	offline	29.08.2022 17:04:35
<input checked="" type="checkbox"/>	5928916b-ffe7-4507-b6dd-2be826d761aa	admin		...	offline	24.08.2022 15:18:14
<input checked="" type="checkbox"/>	acd85d21-9417-4ced-943c-0d6efd727993	admin		...	offline	12.08.2022 18:10:15
<input checked="" type="checkbox"/>	f422f69a-913f-469f-9b7c-5db452a296a9	admin		...	offline	08.08.2022 15:56:14
<input checked="" type="checkbox"/>	93c8c7bf-6aa7-4a8b-98be-afb21e542309	admin		...	offline	05.08.2022 18:26:04
<input checked="" type="checkbox"/>	09b30e09-6e00-4d41-8ec5-cc90be5ea69d	admin		...	offline	03.08.2022 20:27:04
<input checked="" type="checkbox"/>	710b4708-6ac9-4c9e-abe3-96566329efbc	admin		...	offline	03.08.2022 19:35:00
<input type="checkbox"/>	3f5abdd-bd4e-4cbe-9ff5-a9b7868c7707	admin		...	offline	03.08.2022

7 X

Block

- Specify the reason for the device to be blocked (mandatory) and the block expiry date (optional). If no date is specified, the block will be permanent.
- Click *Save*.

Create blocklist record

UUID
e34c4c6f-51f1-4322-ad0b-203e86f3f59d, 5928916b-ffe7-4507-b6dd-2be826d761aa, acd85d21-9417-4ced-943c-0d6efd727993, f422f69a-913f-469f-9b7c-5db452a296a9, 93c8c7bf-6aa7-4a8b-98be-afb21e542309, 09b30e09-6e00-4d41-8ec5-cc90be5ea69d, 710b4708-6ac9-4c9e-abe3-96566329efbc

Reason

Expires

Cancel

1.4.5 Camera Groups

Camera groups are entities that are used for video footage classification. After processing a video, the system will attribute the face recognition events obtained from the video to a designated camera group. It makes the further event handling and search a lot easier.

In the current version, all video files uploaded to the system are automatically added to a single group *Video archive default camera group*. If necessary, you can alter its parameters.

Do the following:

- Navigate *Settings* -> *Camera Groups*.

2. Click *Video archive default camera group* on the list.

2 in total [+ Add new camera group](#)

<input type="checkbox"/>	Name	Active
<input type="checkbox"/>	Video archive default Camera Group	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Default Camera Group	<input checked="" type="checkbox"/>

3. On the *Information* tab, modify the group name. Add a comment if needed.

The screenshot shows a configuration window for 'Video archives'. The window title is 'Video archives' with a 'CHANGED' indicator. The 'Information' tab is selected, showing fields for 'Name' (set to 'Video archives'), 'Comment', 'Labels', 'Deduplicate events with interval' (set to 15), and 'Similarity threshold' (checked). The bottom of the window shows a toolbar with 'Deactivate selected' and 'Save selected 1' buttons.

4. By default, video from all camera groups is processed using the *generic confidence threshold*. To set an individual threshold for the camera group, enable *Similarity Threshold* and specify the threshold value.
5. On the *Permissions* tab, assign privileges on the camera group, specifying which user roles are allowed to change/view the camera group settings.

The screenshot shows the 'Video archives' section with the 'Permissions' tab selected. It displays a table with 3 items in total. The table has columns for 'Name', 'View', and 'Change'. The 'View' and 'Change' columns contain checkboxes. The 'Administrator' and 'Operator' rows have both checkboxes checked, while the 'User' row has the 'View' checkbox checked and the 'Change' checkbox unchecked. At the bottom, there is a bar with a checkmark, a '1' in a blue circle, a close button, a 'Deactivate selected' button, a 'Save selected 1' button, and a refresh button.

Name	View	Change
Administrator	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Operator	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
User	<input checked="" type="checkbox"/>	<input type="checkbox"/>

6. Save the changes.

1.4.6 Watch Lists

The appearance of specific individuals in the video is monitored with a set of default and custom watch lists.

Records of individuals are assigned to watch lists. Once a watch list is activated, the system will be looking for each person on it during video processing or *remote monitoring*.

You can create as many custom watch lists as necessary: wanted, suspects, etc. — subject to your needs.

In this section:

- [Monitoring Unmatched Faces](#)
- [Create Watch List](#)

Monitoring Unmatched Faces

FindFace features a special pre-configured watch list used for monitoring only unmatched faces (faces that do not match any record). This watch list cannot be removed from the system. To edit its settings, navigate to the *Settings* tab. Click *Watch Lists* and then click *Unmatched*.

h U Unmat... Information Permissions

Name Unmatched Color #ffffff

Camera groups

Comment Default list for unmatched events

Similarity threshold

Require event acknowledgement

Enable sound alert

Active

✓ 1 × ... Deactivate selected

Create Watch List

To create a custom watch list, do the following:

1. Navigate *Settings* -> *Watch Lists*.
2. Click + *Add new watch list*.

2 in total Q Search + Add new watch list

<input type="checkbox"/>	Name	Active
<input type="checkbox"/>	Unmatched	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Default Watch List	<input checked="" type="checkbox"/>

3. On the *Information* tab, specify the watch list name.
4. From the *Color* palette, select a color which will be shown in notifications for this list.

The screenshot shows the configuration page for a watch list named "Fugitives". The page has two tabs: "Information" (selected) and "Permissions". At the top right, there is a "CHANGED" indicator with a close button, and a menu icon with "X" and ">" options. The "Information" tab contains the following fields and controls:

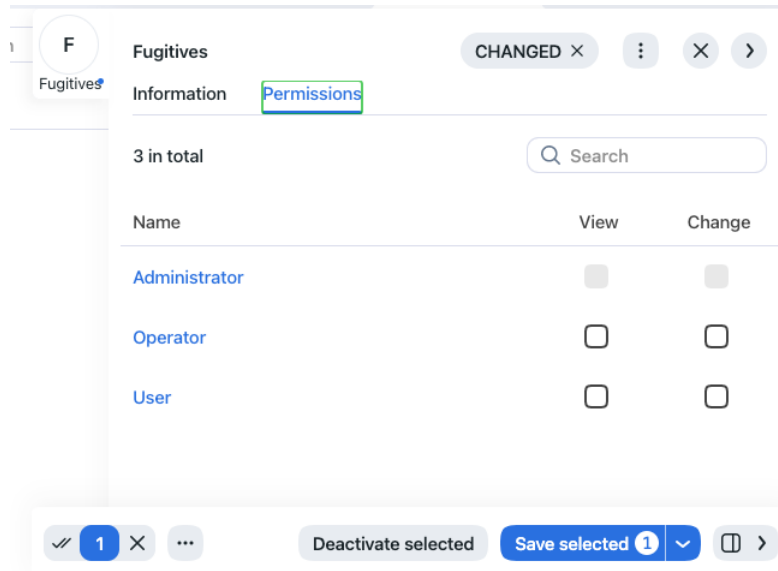
- Name:** A text input field containing "Fugitives".
- Color:** A color picker showing a red square and the hex code "# e92020".
- Camera groups:** A dropdown menu that is currently empty.
- Comment:** A text input field containing "Armed and dangerous".
- Similarity threshold:** A toggle switch that is turned on, followed by a slider and a numeric input field showing "0.616".
- Checkboxes:** Three checked checkboxes: "Require event acknowledgement", "Enable sound alert", and "Active".

At the bottom of the form, there is a toolbar with a checkmark icon, a "1" in a blue circle, an "X" icon, a menu icon, a "Deactivate selected" button, a "Save selected" button with a "1" in a blue circle and a dropdown arrow, a list icon, and a ">" icon.

- Describe the watch list in a comment if needed.
- By default, all watch lists in the system are applied the *generic confidence threshold*. To set an individual threshold for the watch list, check *Similarity Threshold* and specify the threshold value.

Important: The default generic confidence threshold is optimal for the majority of recognition cases. We do not recommend changing it on your own. Be sure to consult with our technical experts prior (support@ntechlab.com).

- Enable *Require acknowledgment* if it is mandatory that events associated with the list be manually acknowledged.
- Enable *Enable sound alert* to turn on sound notifications for the list if needed.
- On the *Permissions* tab, assign privileges on the watch list, specifying which user roles are allowed to change/view the watch list settings.



10. Activate and save the watch list.

1.4.7 Custom Tabs, Fields, and Filters in Global Records

See also:

To create custom fields in the feature vector database, refer to *Custom Metadata in Tarantool*.

To add custom tabs and fields to the records of individuals, do the following:

1. Prepare the list of custom tabs and fields you want to add to the records.
2. Open the `/etc/findface-security/config.py` configuration file.

```
sudo vi /etc/findface-security/config.py
```

3. Uncomment the `FFSECURITY -> CUSTOM_FIELDS -> human_card` section and modify the exemplary content, considering the following:

- `'items'`: the list of fields in a record. Describe each field with the following parameters:
 - `'name'`: field's internal name, string.
 - `'default'`: field's default value. If a default value exceeds $1e14 - 1$, use a string data type to specify it, for example, `"123123..."` instead of `123123...`
 - `'label'`: field's label in a record, string.
 - `'tab'`: tab that features the field.
 - `'display'`: display format (`form` or `list`), string or array.
 - `'description'`: field's description, string.
 - `'editable'`: field's editability, boolean.
 - `'type'`: field data type, string. Possible values:
 - * `list`: requires `items`, additional parameter for lists (see below), expects objects `{id, name}` in dictionaries;

- * `valuelist`: expects elements of primitive types.
- * `objectlist`: allows for creating arrays of objects of required types.
- * `datetime`: primitive data type displayed as a datetime list.
- * `date`: primitive data type displayed as a date picker.
- * `boolean`: primitive data type displayed as a checkbox.
- * `string`: primitive data type `string`.
- additional parameters for lists (`type=list`, `type=valuelist`):
 - * `multiple`: possibility of selecting several items in the list, boolean.
 - * `items`: dictionary used as a data source for the list.
 - * `allow_create`: possibility of adding new items to the list.
 - * `custom_id`: custom field for id (`type=list`).
- additional parameters for object lists (`type=objectlist`).
 - * `object`: objects used as a data source for the object list.
 - * `simple`: indicator that the field expects data of a primitive type instead of objects, for example, expects strings with phone numbers.
- `'filters'`: the list of search filters associated with the custom fields. Parameters:
 - `'name'`: filter's internal name,
 - `'label'`: filter's label in the web interface,
 - `'field'`: associated field in the format `[field name]`.
- `'tabs'`: the list of tabs in a record.

```
FFSECURITY = {
...
# -- Custom model fields --
# Edit CUSTOM_FIELDS -> `human_card` section to customize human card fields.
# Edit CUSTOM_FIELDS -> `car_card` section to customize car card fields.
...
  'CUSTOM_FIELDS': {
    'human_card': {
      'items': [
        {
          'name': 'personid',
          'default': '',
          'label': 'PersonID',
          'display': ['list', 'form'],
          'description': 'Sigur person ID',
          'editable': False
        },
        {
          'name': 'firstname',
          'default': '',
          'label': 'First Name',
```

(continues on next page)

(continued from previous page)

```
        'display': ['list', 'form'],
        'description': 'Sigur first name',
        'editable': False
    },
    {
        'name': 'lastname',
        'default': '',
        'label': 'Last Name',
        'display': ['list', 'form'],
        'description': 'Sigur last name',
        'editable': False
    },
    {
        'name': 'version',
        'default': '',
        'label': 'Version',
        'display': ['list', 'form'],
        'description': 'Sigur photo version',
        'editable': False
    }
],
'filters': [
    {
        'name': 'personid',
        'label': 'Sigur person ID filter',
        'field': 'personid'
    }
]
},
'car_card': {}, # same fields are available
}
```

4. Restart the findface-security service.

```
sudo systemctl restart findface-security.service
```

You will see the custom content appear in the records.

1.4.8 Custom Metadata in Tarantool

It is often necessary to assign additional metadata to the faces extracted from images uploaded to the record index and now stored in the feature vector database.

In this section:

To assign custom meta fields to the face objects, do the following:

1. Prepare the list of custom meta fields to assign.
2. Open the `/etc/findface-security/config.py` configuration file.

```
sudo vi /etc/findface-security/config.py
```

3. In the `FFSECURITY` section, uncomment the `CUSTOM_FIELDS -> face_object` section and modify the exemplary content, considering the following:
 - `field_name`: field's name;
 - `type`: data type;
 - `default`: field's default value. If a default value exceeds $1e14 - 1$, use a string data type to specify it, for example, `"123123.."` instead of `123123...`

```
FFSECURITY = {
...
    # -- Custom model fields --
    ...
    # Edit CUSTOM_FIELDS -> `face_object` section to customize face object fields.
    ...
    # 'CUSTOM_FIELDS': {
        ...
        'face_object': {
            'items': [
                {
                    "field_name": "tag_name_1",
                    "type": "string",
                    "default": "change_me"
                },
                {
                    "field_name": "tag_name_2",
                    "type": "uint",
                    "default": 123
                },
                {
                    "field_name": "tag_name_3",
                    "type": "bool",
                    "default": True
                },
            ],
        }
    }
}
```

4. *Add the new meta fields* to the feature vector database structure.
5. Restart the `findface-security` service.

```
sudo systemctl restart findface-security.service
```

You can work with the new meta fields through *HTTP API* using the `objects/faces/` methods.

See also:

To create custom tabs, fields, and filters in records, refer to *Custom Tabs, Fields, and Filters in Global Records*.

1.4.9 Enable Data Encryption

To ensure data security, we recommend you enabling SSL encryption. Do the following:

1. Under the nginx configuration directory, create a directory that will be used to hold all of the SSL data:

```
sudo mkdir /etc/nginx/ssl
```

2. Create the SSL key and certificate files:

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/nginx/ssl/my-  
example-domain.key -out /etc/nginx/ssl/my-example-domain.crt
```

You will be asked a few questions about your server in order to embed the information correctly in the certificate. Fill out the prompts appropriately. The most important line is the one that requests the Common Name. You need to enter the domain name or public IP address that you want to be associated with your server. Both of the files you created (`my-example-domain.key` and `my-example-domain.crt`) will be placed in the `/etc/nginx/ssl` directory.

3. Configure nginx to use SSL. Open the nginx configuration file `/etc/nginx/sites-available/ffsecurity-nginx.conf`. Apply the following modifications to the file:

1. Add the new `server { ... }` section that contains the URL replacement rule:

```
server {  
    listen 80;  
    server_name my-example-domain.com www.my-example-domain.com;  
    rewrite ^(.*) https://my-example-domain.com$1 permanent;  
    access_log off;  
}
```

2. Comment out the following lines in the existing `server { ... }` section:

```
# listen 80 default_server;  
# listen [::]:80 default_server;
```

3. Add the following lines, including the paths to the certificate and the key, to the existing `server { ... }` section:

```
listen 443 ssl;  
  
ssl_certificate /etc/nginx/ssl/my-example-domain.crt;  
ssl_certificate_key /etc/nginx/ssl/my-example-domain.key;
```

4. In the generic nginx configuration file `/etc/nginx/nginx.conf`, find the SSL Settings section and append the following lines:

```
ssl_session_cache    shared:SSL:10m;
ssl_session_timeout 1h;
```

The example of the configuration file `/etc/nginx/sites-available/ffsecurity-nginx.conf` with correctly configured SSL settings is shown below:

```
upstream ffsecurity {
    server 127.0.0.1:8002;
}

upstream ffsecurity-ws {
    server 127.0.0.1:8003;
}

map $http_upgrade $ffsec_upstream {
    default "http://ffsecurity-ws";
    "" "http://ffsecurity";
}

server {
    listen 80;
    server_name my-example-domain.com www.my-example-domain.com;
    rewrite ^(.*) https://my-example-domain.com$1 permanent;
    access_log off;
}

server {
    # listen 80 default_server;
    # listen [::]:80 default_server;
    listen 443 ssl;
    ssl_certificate /etc/nginx/ssl/my-example-domain.com.crt;
    ssl_certificate_key /etc/nginx/ssl/my-example-domain.com.key;

    root /var/lib/findface-security;

    autoindex off;

    server_name _;

    location = / {

        alias /usr/share/findface-security-ui/;
        try_files /index.html =404;
    }
    location /static/ {

    }
    location /uploads/ {
        add_header 'Access-Control-Allow-Origin' '*';
        add_header 'Access-Control-Allow-Methods' 'GET';
        add_header 'Access-Control-Allow-Headers' 'DNT,User-Agent,X-
```

(continues on next page)

(continued from previous page)

```

↪Requested-With,If-Modified-Since,Cache-Control,Content-Type,Range,
↪Authorization';
    add_header 'Access-Control-Expose-Headers' 'Content-Length,
↪Content-Range';
    add_header 'Access-Control-Max-Age' 2592000;
}
location /ui-static/ {
    alias /usr/share/findface-security-ui/ui-static/;
}
location /doc/ {
    alias /opt/findface-security/doc/;
}
location ~ /videos/(?<video_id>[0-9]+)/upload/(.*)$ {
    if ($request_method = 'OPTIONS') {
        add_header 'Content-Type' 'text/plain; charset=utf-8';
        add_header 'Content-Length' 0;
        return 204;
    }
    set $auth_request_uri "http://ffsecurity/videos/$video_id/auth-
↪upload/";
    auth_request /video-upload-auth/;

    alias "/var/lib/findface-security/uploads/videos/$video_id.bin";
    client_max_body_size 15g;

    dav_access user:rw group:rw all:rw;
    dav_methods PUT;

    create_full_put_path on;
    autoindex off;
    autoindex_exact_size off;
    autoindex_localtime on;
    charset utf-8;

    add_header 'Access-Control-Allow-Origin' '*';
    add_header 'Access-Control-Allow-Methods' 'PUT, OPTIONS';
    add_header 'Access-Control-Allow-Headers' 'authorization';
}
location = /video-upload-auth/ {
    internal;
    client_max_body_size 15g;
    proxy_set_header Content-Length "";
    proxy_set_header Host $http_host;
    proxy_set_header X-Forwarded-For $remote_addr;
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_pass_request_body off;
    proxy_pass $auth_request_uri;
}

location / {
    client_max_body_size 300m;
    proxy_set_header Host $http_host;

```

(continues on next page)

(continued from previous page)

```

proxy_set_header X-Forwarded-For $remote_addr;
proxy_set_header X-Forwarded-Proto $scheme;
proxy_http_version 1.1;
proxy_set_header Upgrade $http_upgrade;
proxy_set_header Connection "upgrade";
proxy_pass $ffsec_upstream;
proxy_read_timeout 5m;

location ~ ^/(cameras|videos)/([0-9]+)/stream/?$ {
    proxy_set_header Host $http_host;
    proxy_set_header X-Forwarded-For $remote_addr;
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_pass http://ffsecurity;
}

location ~ ^/streams/(.*)$ {
    internal;
    proxy_pass $1;
}
}
}

```

- Restart nginx.

```
sudo systemctl restart nginx.service
```

- Edit the `/etc/findface-security/config.py` configuration file. In the `EXTERNAL_ADDRESS` and `ROUTER_URL` parameters, substitute the `http://` prefix with `https://`.

```

sudo vi /etc/findface-security/config.py

...
EXTERNAL_ADDRESS="https://my-example-domain.com"
...
ROUTER_URL="https://IP_address"

```

- Restart findface-security.

```
sudo systemctl restart findface-security
```

- If there are running `findface-video-worker` services in the system, you need to either recreate cameras in the web interface, or change the `router_url` parameter in relevant video processing jobs, substituting the `http://` prefix with `https://`. This can be done with the following command:

```

curl -s localhost:18810/jobs | jq -r '.[]["id"]' | xargs -I {} curl -X PATCH -d '{
↪ "router_url": "https://my-example-domain.com/video-detector/frame"}' http://
↪ localhost:18810/job/{}

```

1.4.10 Enable Face Attribute Recognition

The face attributes such as age, gender, emotions, etc. present in the filter set for detected face analysis during a *case investigation*.

Face attribute recognition can be automatically enabled and configured during the *FindFace installation*. If you skip this step, you can manually do so afterwards. Face attribute recognition works on both GPU- and CPU-acceleration.

Do the following:

1. Open the `/etc/findface-extraction-api.ini` configuration file.

```
sudo vi /etc/findface-extraction-api.ini
```

2. Specify the relevant recognition models in the `extractors` section, as shown in the example below. Be sure to indicate the right acceleration type for each model, matching the acceleration type of `findface-extraction-api`: CPU or GPU. Be aware that `findface-extraction-api` on CPU can work only with CPU-models, while `findface-extraction-api` on GPU supports both CPU- and GPU-models.

```
extractors:
  face_age: faceattr/age.v2.gpu.fnk
  face_beard: faceattr/beard.v0.gpu.fnk
  face_emotions: faceattr/emotions.v1.gpu.fnk
  face_gender: faceattr/gender.v2.gpu.fnk
  face_glasses3: faceattr/glasses3.v0.gpu.fnk
  face_medmask3: faceattr/medmask3.v2.gpu.fnk
  face_headpose: faceattr/headpose.v2.gpu.fnk
```

The following models are available:

Face attribute	Acceleration	Configure as follows
age	CPU	<code>face_age: faceattr/age.v2.cpu.fnk</code>
	GPU	<code>face_age: faceattr/age.v2.gpu.fnk</code>
gender	CPU	<code>face_gender: faceattr/gender.v2.cpu.fnk</code>
	GPU	<code>face_gender: faceattr/gender.v2.gpu.fnk</code>
emotions	CPU	<code>face_emotions: faceattr/emotions.v1.cpu.fnk</code>
	GPU	<code>face_emotions: faceattr/emotions.v1.gpu.fnk</code>
glasses	CPU	<code>face_glasses3: faceattr/glasses3.v0.cpu.fnk</code>
	GPU	<code>face_glasses3: faceattr/glasses3.v0.gpu.fnk</code>
beard	CPU	<code>face_beard: faceattr/beard.v0.cpu.fnk</code>
	GPU	<code>face_beard: faceattr/beard.v0.gpu.fnk</code>
face mask	CPU	<code>face_medmask3: faceattr/medmask3.v2.cpu.fnk</code>
	GPU	<code>face_medmask3: faceattr/medmask3.v2.gpu.fnk</code>
head pose	CPU	<code>face_headpose: faceattr/headpose.v2.cpu.fnk</code>
	GPU	<code>face_headpose: faceattr/headpose.v2.gpu.fnk</code>

Tip: To leave a recognition model disabled, pass the empty value `""` to the relevant parameter. Do not remove the parameter itself. Otherwise, the system will be searching for the default model.

```
extractors:
  face_age: ""
  face_beard: ""
  face_emotions: ""
```

(continues on next page)

(continued from previous page)

```
face_gender: ""
face_glasses3: ""
face_medmask3: ""
```

Note: You can find face attribute recognition models at `/usr/share/findface-data/models/faceattr/`.

```
ls /usr/share/findface-data/models/faceattr/
age.v2.cpu.fnk age.v2.gpu.fnk beard.v0.cpu.fnk beard.v0.gpu.fnk emotions.v1.cpu.
↪fnk emotions.v1.gpu.fnk gender.v2.cpu.fnk gender.v2.gpu.fnk glasses3.v0.cpu.
↪fnk glasses3.v0.gpu.fnk medmask3.v2.cpu.fnk medmask3.v2.gpu.fnk liveness.colombo.
↪cpu.fnk liveness.colombo.gpu.fnk liveness.pacs.v0.cpu.fnk liveness.pacs.v0.gpu.
↪fnk quality.v1.cpu.fnk quality.v1.gpu.fnk
```

- Restart `findface-extraction-api`.

```
sudo systemctl restart findface-extraction-api
```

- To display the face attribute recognition results in the event list, open the `/etc/findface-security/config.py` configuration file.

```
sudo vi /etc/findface-security/config.py
```

- Specify the required models in the following line of the `FFSECURITY` section, subject to the list of enabled models:

```
FFSECURITY = {
    ...
    'FACE_EVENTS_FEATURES': ['gender', 'age', 'emotions', 'beard', 'glasses',
↪'medmask', 'headpose']
    ...
}
```

- Restart `findface-security`.

```
sudo systemctl restart findface-security
```

1.4.11 Enable Face Liveness Detection

The FindFace face liveness detector tells apart live faces from face images. The face liveness binary result `real/fake` serves as one of the filters for detected face analysis during *case investigation*.

The face liveness detector can be automatically enabled and configured during the *installation*. If you skip this step, you can manually do so afterwards, following the instructions below.

Note: The face liveness detector functions on both GPU- and CPU-acceleration. However, it is much slower on CPU.

In this section:

- *Enable Face Liveness Detector*
- *Configure Liveness Threshold*

Enable Face Liveness Detector

To enable the face liveness detector, do the following:

1. Open the `/etc/findface-video-worker-gpu.ini` (`/etc/findface-video-worker-cpu.ini`) configuration file. In the `liveness` section, specify the path to the neural network model (`fnk`) used in the face liveness detector.

```
sudo vi /etc/findface-video-worker-gpu.ini

#-----
[liveness]
#-----
## path to liveness fnk
## type:string env:CFG_LIVENESS_FNK longopt:--liveness-fnk
fnk = /usr/share/findface-data/models/faceattr/liveness.pacs.v0.gpu.fnk
```

```
sudo vi /etc/findface-video-worker-cpu.ini

#-----
[liveness]
#-----
## path to liveness fnk
## type:string env:CFG_LIVENESS_FNK longopt:--liveness-fnk
fnk = /usr/share/findface-data/models/faceattr/liveness.pacs.v0.cpu.fnk
```

2. Restart `findface-video-worker`.

```
sudo systemctl restart findface-video-worker-gpu
sudo systemctl restart findface-video-worker-cpu
```

Configure Liveness Threshold

If necessary, you can adjust the `liveness threshold` in the `/etc/findface-security/config.py` configuration file. The liveness detector will estimate a face liveness with a certain level of confidence. Depending on the threshold value, it will return a binary result `real` or `fake`.

Note: The default value is optimal. Before changing the threshold, we recommend you to seek advice from our experts by support@ntechlab.com.

```
sudo vi /etc/findface-security/config.py

LIVENESS_THRESHOLD: 0.85,
```

1.4.12 Integration with Remote Facial Recognition Systems

You can integrate your FindFace instance with remote facial recognition systems. In this case, the server known as a puppeteer will be pushing records designated for remote alerting to remote servers known as puppets. In return, it will be receiving recognition events matching with those records.

This functionality has a large scope of possible applications. One course is tracking offenders' location and routes and detecting alleged accomplices. Another one is finding missing people. The results are especially great if applied to Public and Transport Safety systems with thousands of cameras.

The current version supports only integration with facial recognition systems from the FindFace family.

In this section:

- *Sync Schedule*
- *Configure Puppeteer*
- *Configure Puppet*
- *Remote Alerting and Remote Search on Puppeteer*

Sync Schedule

The data between a puppeteer and a puppet are synced in the following way:

- The puppeteer delivers designated records to the puppet with an interval specified in the `REMOTE_MONITORING_SYNC_INTERVAL` parameter (see configuration below).
- The puppet delivers matching recognition events to the puppeteer as soon as they appear.

Configure Puppeteer

To configure your FindFace instance to be a puppeteer, do the following:

1. Open the `/etc/findface-security/config.py` configuration file. Make sure that the `EXTERNAL_ADDRESS` parameter is filled.

```
sudo vi /etc/findface-security/config.py

EXTERNAL_ADDRESS = 'http://192.168.0.4'
```

2. Find the Puppeteer section.

```
# ===== Puppeteer =====
# INSTALLED_APPS.append('ffsecurity_puppeteer')

# PUPPETEER_CONFIG = {
#     'UNSAVED_RESULTS_DELETION_TIMEOUT': 3600,           # maximum lifetime of search_
#     ↪ results not saved involuntarily
#     'REMOTE_MONITORING_SYNC_INTERVAL': 600,           # monitoring data_
#     ↪ synchronization interval, seconds
#     'REMOTE_MONITORING_EVENTS_MAX_AGE': 30,           # monitoring events older than_
```

(continues on next page)

(continued from previous page)

```

↪this number of days will be
#                                     # automatically deleted
↪(every night at 1:17 am by default)
#   'ENABLE_DAILY_SEARCH': False,     # daily search activation
↪(default False)
#   'DAILY_SEARCH_PUSH_HOUR': 2,     # daily search cards
↪synchronization hour
#   'DAILY_SEARCH_PULL_HOUR': 6,     # hour in which results of
↪daily search will be obtained
#   'puppets': [
#     {
#       'id': 'first_puppet',         # puppet ID
#       'url': 'http://1.1.1.1:8010/', # puppet URL
#       'token': 'first_puppet_token', # use pwgen -s 64 1 (should
↪match the token in puppet)
#       'facen_model': 'mango_320'   # face model in puppet
#     },
#     {
#       'id': 'second_puppet',
#       'url': 'http://1.1.1.1:8010/',
#       'token': 'second_puppet_token',
#
#       # if remote installation has a different face model than the one
↪used in FFSecurity -
#       # you need to specify its name and ExtractionAPI URL where the
↪corresponding face model is specified
#       'facen_model': 'grapefruit_480',
#       'extractor': 'http://127.0.0.1:18667',
#     },
#   ]
# }
#

```

3. Uncomment the section as shown in the example below and specify the following parameters:

- `REMOTE_MONITORING_SYNC_INTERVAL`: interval in seconds with which the puppeteer sends designated records to a puppet.
- `REMOTE_MONITORING_EVENTS_MAX_AGE`: remote alerts older than this number of days will be automatically deleted on the puppeteer (every night at 1:17 a.m. by default).
- `puppets` -> `id`: a puppet ID.
- `puppets` -> `url`: IP address and port of a puppet's principal server.

Specify the port as follows:

- Leave the default port as is if the puppet represents a public or transport safety system (i.e., it has the `public-security` service at its core).
- Switch the default port to `80` or do not specify it at all if the puppet is based on the `findface-security` service (i.e., with FindFace Security or FindFace Multi installed).
- `puppets` -> `token`: token for mutual authentication between the puppeteer and a puppet.

Tip: Use the following command to generate a random token.

```
pwgen -s 64 1
```

- puppets -> facen_model: neural network model used on a puppet for face recognition.
- puppets -> extractor: IP address and port of the biometric data extraction service on a puppet if the neural network model for face recognition on the puppet differs from that on the puppeteer.

Leave other parameters in the section commented out.

```
# ===== Puppeteer =====
INSTALLED_APPS.append('ffsecurity_puppeteer')

PUPPETEER_CONFIG = {
#   'UNSAVED_RESULTS_DELETION_TIMEOUT': 3600,           # maximum lifetime of search
#   →results not saved involuntarily
#   'REMOTE_MONITORING_SYNC_INTERVAL': 600,           # monitoring data
#   →synchronization interval, seconds
#   'REMOTE_MONITORING_EVENTS_MAX_AGE': 30,           # monitoring events older than
#   →this number of days will be
#   #                                                 # automatically deleted
#   →(every night at 1:17 am by default)
#   'ENABLE_DAILY_SEARCH': False,                     # daily search activation
#   →(default False)
#   'DAILY_SEARCH_PUSH_HOUR': 2,                      # daily search cards
#   →synchronization hour
#   'DAILY_SEARCH_PULL_HOUR': 6,                      # hour in which results of
#   →daily search will be obtained
#   'puppets': [
#       {
#           'id': '1',                                 # puppet ID
#           'url': 'http://192.168.0.5:8010/',          # puppet URL
#           'token': '1234567890',                     # use pwgen -s 64 1 (should match the
#   →token in puppet)
#           'facen_model': 'mango_320'                 # face model in puppet
#       },
#       {
#           'id': '2',
#           'url': 'http://192.168.0.6:8010/',
#           'token': '0987654321',
#           'facen_model': 'grapefruit_480',
#           'extractor': 'http://192.168.0.6:18667',
#       },
#   ],
#   #
#   #   # if remote installation has a different face model than the one
#   →used in FFSecurity -
#   #   #   # you need to specify its name and ExtractionAPI URL where the
#   →corresponding face model is specified
#   },
# ]
}
```

4. Restart the findface-security service.

```
sudo systemctl restart findface-security.service
```

5. Perform migration and re-create user groups to have enough permissions for working with data from puppets.

```
sudo findface-security migrate
sudo findface-security create_groups
```

Configure Puppet

To configure a remote FindFace instance to be a puppet, do the following:

1. Open the `/etc/findface-security/config.py` configuration file. Make sure that the `EXTERNAL_ADDRESS` parameter is filled.

```
sudo vi /etc/findface-security/config.py

EXTERNAL_ADDRESS = 'http://192.168.0.5'
```

2. Find the Vns section.

```
# ===== Vns =====
# A plugin for using FindFace Security as a puppeteer server
# INSTALLED_APPS.append('ffsecurity_vns')

# VNS_CONFIG = {
#     'USERS': {
#         'user1': 'token1',
#         'user2': 'token2'
#     },
#     'MONITORING_THRESHOLD': 0.75,
#     'DAILY': {
#         'ENABLED': False,
#         'THRESHOLD': 0.75,
#         'START_TIME': "00:00:00"
#     }
# }
```

3. Uncomment the section as shown in the example below and specify the following parameters:

- `token`: token for mutual authentication between the puppet and a puppeteer. You can specify several users and tokens if the puppet is communicating with several puppeteers. You can leave the user names as is.
- `MONITORING_THRESHOLD`: confidence threshold in face recognition events sent to a puppeteer.

Leave other parameters in the section commented out.

```
# ===== Vns =====
# A plugin for using FindFace Security as a puppeteer server
INSTALLED_APPS.append('ffsecurity_vns')

VNS_CONFIG = {
    'USERS': {
        'user1': '1234567890'
    },
```

(continues on next page)

(continued from previous page)

```
'MONITORING_THRESHOLD': 0.75,
#   'DAILY': {
#       'ENABLED': False,
#       'THRESHOLD': 0.75,
#       'START_TIME': "00:00:00"
#   }
}
```

- Restart the findface-security service.

```
sudo systemctl restart findface-security.service
```

- Perform migration to sync with puppeteers.

```
sudo findface-security migrate
```

Remote Alerting and Remote Search on Puppeteer

See *Remote Alerting and Remote Search*

1.4.13 Multiple Video Cards Usage

Should you have several video cards installed on a physical server, you can create additional findface-extraction-api-gpu or findface-video-worker-gpu instances and distribute them across the video cards, one instance per card.

In this section:

- *Distribute findface-extraction-api-gpu Instances Across Several Video Cards*
- *Allocate findface-video-worker-gpu to Additional Video Card*

Distribute findface-extraction-api-gpu Instances Across Several Video Cards

To distribute the findface-extraction-api-gpu instances across several video cards, do the following:

- Prepare the initial configuration file of findface-extraction-api-gpu for future copying. Open /etc/findface-extraction-api.ini. Limit the number of the findface-extraction-api-gpu instances down to 1 per video card.

```
sudo vi /etc/findface-extraction-api.ini

...
extractors:
    instances: 1
```

- Create several copies of the /etc/findface-extraction-api.ini configuration file, subject to how many video cards you are going to use for feature vector extraction. Append numbers of the GPU devices that will be running the instances to the new names (GPU devices #0 and #7 in the example below).

Note: By default, GPU device numeration in a system starts from #0. To list the video cards in use, execute:

```
nvidia-smi
```

```
cp findface-extraction-api.ini findface-extraction-api-0.ini
cp findface-extraction-api.ini findface-extraction-api-7.ini
```

3. Open the new configuration files. Specify the relevant GPU device numbers and adjust the listening ports. Be sure to associate each instance to a unique port.

```
sudo vi /etc/findface-extraction-api-0.ini
```

```
listen: 127.0.0.1:18667
```

```
...
```

```
gpu_device: 0
```

```
...
```

```
sudo vi /etc/findface-extraction-api-7.ini
```

```
listen: 127.0.0.1:18668
```

```
...
```

```
gpu_device: 7
```

```
...
```

4. For the `findface-extraction-api-gpu` instances to work within one system, bind them via a load balancer, e.g., `nginx`. For simplicity, we recommend enabling `nginx` on the standard `findface-extraction-api` port 18666 (see details below).

To set up load balancing, do the following:

1. Create a new configuration file for `nginx`.

```
sudo vi /etc/nginx/sites-available/lb_extractions
```

2. Insert the following entry into the just created file. Be sure to specify the actual listening ports of the `findface-extraction-api-gpu` instances in the `server` directive (18667, 18668 in our example).

```
upstream extractions {
    server 127.0.0.1:18667 max_fails=3 fail_timeout=30s;
    server 127.0.0.1:18668 max_fails=3 fail_timeout=30s;
}

server {
    listen 18666 default_server;

    server_name _;

    location / {
        client_max_body_size 100m;
        proxy_pass http://extractions;
    }
}
```

(continues on next page)

(continued from previous page)

```
}
}
```

3. Start the load balancer.

```
ln -s /etc/nginx/sites-available/lb_extractions /etc/nginx/sites-enabled/lb_
↪extractions
nginx -t
nginx -s reload
```

5. Stop the initial findface-extraction-api-gpu service and disable its starting on boot.

```
systemctl stop findface-extraction-api.service
systemctl disable findface-extraction-api.service
```

6. Create a new service for the findface-extraction-api-gpu instances.

```
/etc/systemd/system/findface-extraction-api@.service
```

Insert the following entry:

```
[Unit]
Description=Findface Extraction API service %i instance

[Service]
User=ntech
Restart=always
RestartSec=5
Type=notify
ExecStart=/usr/bin/findface-extraction-api -config="/etc/findface-extraction-api-%i.
↪ini"
TimeoutStartSec=60m

[Install]
WantedBy=multi-user.target
```

7. Start the findface-extraction-api-gpu instances. The first start might take up to 10 minutes.

```
systemctl daemon-reload
systemctl enable findface-extraction-api@{0,7}
systemctl start findface-extraction-api@{0,7}
```

Tip: To check the status, use the following command:

```
sudo systemctl list-units 'findface*' -a
```

Allocate findface-video-worker-gpu to Additional Video Card

To create an additional findface-video-worker-gpu instance and allocate it to a different video card, do the following:

1. Display the status of the findface-video-worker-gpu primary service by executing:

```
sudo systemctl status findface-video-worker-gpu.service
```

2. Find the full path to the service in the following line:

```
Loaded: loaded (/usr/lib/systemd/system/findface-video-worker-gpu.service); enabled;  
↪ vendor preset: enabled
```

It is findface-video-worker-gpu.service in our example (name may vary). Create a copy of the service under a new name.

```
sudo cp /usr/lib/systemd/system/findface-video-worker-gpu.service /usr/lib/systemd/  
↪system/findface-video-worker-gpu2.service`
```

3. In the same manner, create a copy of the primary service configuration file under a new name.

```
sudo cp /etc/findface-video-worker-gpu.ini /etc/findface-video-worker-gpu2.ini
```

4. Open the just created configuration file and actualize the GPU device number to use. Modify the streamer port number by the following formula: 18999 (port number for GPU #0) - GPU device number, i.e. for the GPU #1, port = 18998, for the GPU #2, port = 18997, and so on.

```
sudo vi /etc/findface-video-worker-gpu2.ini  
  
## cuda device number  
device_number = 1  
  
...  
  
#-----  
[streamer]  
#-----  
## streamer/shots webservice port, 0=disabled  
## type:number env:CFG_STREAMER_PORT longopt:--streamer-port  
port = 18999  
...  
...
```

5. Open the new service and specify the just created configuration file.

```
sudo vi /usr/lib/systemd/system/findface-video-worker-gpu2.service  
  
ExecStart=/usr/bin/findface-video-worker-gpu --config /etc/findface-video-worker-  
↪gpu2.ini
```

6. Reload the systemd daemon to apply the changes.

```
sudo systemctl daemon-reload
```

7. Enable the new service autostart.

```
sudo systemctl enable findface-video-worker-gpu2.service
```

```
Created symlink from /etc/systemd/system/multi-user.target.wants/findface-video-
worker-gpu2.service to /usr/lib/systemd/system/findface-video-worker-gpu2.service
```

8. Launch the new service.

```
sudo systemctl start findface-video-worker-gpu2.service
```

9. Check the both findface-video-worker-gpu services status.

```
sudo systemctl status findface-video-worker-* | grep -i 'Active:' -B 3

findface-video-worker-gpu2.service - findface-video-worker-gpu daemon
  Loaded: loaded (/usr/lib/systemd/system/findface-video-worker-gpu2.service;
enabled; vendor preset: enabled)
  Active: active (running) since Thu 2019-07-18 10:32:02 MSK; 1min 11s ago
...

findface-video-worker-gpu.service - findface-video-worker-gpu daemon
  Loaded: loaded (/usr/lib/systemd/system/findface-video-worker-gpu.service;
enabled; vendor preset: enabled)
  Active: active (running) since Mon 2019-07-15 15:18:33 MSK; 2 days ago
```

1.4.14 Automatic Tarantool Recovery

Important: We highly recommend that you turn off your Tarantool servers properly in every unstable situation. This measure will prevent data corruption, so there will be no need in the functionality described in this section.

Warning: Be extremely careful with enabling this functionality as it may lead to silent data loss.

If your system architecture doesn't imply uninterrupted availability of Tarantool servers, it is recommended to enable automatic database recovery. In this case, each time an error occurs while reading a snapshot or xlog file, Tarantool will skip invalid records, read as much data as possible, and re-build the file.

To enable automatic database recovery, do the following:

1. For each Tarantool shard, open the configuration file `/etc/tarantool/instances.available/shard-*.lua` and uncomment `force_recovery = true`.

```
sudo vi /etc/tarantool/instances.available/shard-*.lua

box.cfg{
    force_recovery = true,
}
```

2. Restart the shards.

```
systemctl restart tarantool@shard-*
```

1.4.15 Configure Saving Images in Reports

When building *reports*, you will be able to choose to save the report images as links, thumbnails, or full frames. It is possible to configure the image parameters. To do so, open the `/etc/findface-security/config.py` configuration file and alter the default JPEG quality and the maximum height of thumbnails and full frames, subject to your free disc space.

```
sudo vi /etc/findface-security/config.py

# reports image saving options
'REPORT_THUMBNAIL_JPEG_QUALITY': 75,
'REPORT_THUMBNAIL_MAX_HEIGHT': 100,
'REPORT_FULLFRAME_JPEG_QUALITY': 75,
'REPORT_FULLFRAME_MAX_HEIGHT': 250,
```

Restart the `findface-security` service.

```
sudo systemctl restart findface-security.service
```

1.5 Maintenance and Troubleshooting

1.5.1 Back Up and Recover FindFace and Data

This section is all about the backup and recovery of the FindFace data storages and your system as a whole.

In this section:

- *List of Data Storages*
- *Feature Vector Database Backup and Recovery (Tarantool)*
 - *Utilities*
 - *Back Up Database*
 - *Recover Database*
- *Main Database Backup and Recovery (PostgreSQL)*
- *Artifacts Backup and Recovery (findface-upload)*
- *Settings Backup and Recovery*
- *Back Up and Recover Entire System*

List of Data Storages

FindFace uses the following data storages:

- Tarantool-based feature vector database that stores face feature vectors and recognition events.
- PostgreSQL-based main system database `ffsecurity` that stores internal system data, records, and user accounts.
- The `/var/lib/findface-security/uploads` directory that stores photos uploaded to records, video files, full frames of events, and thumbnails.
- The `/var/lib/ffupload/` directory that stores such event artifacts as normalized face images.

Feature Vector Database Backup and Recovery (Tarantool)

There are the following galleries in the Tarantool-based feature vector database:

- `ffsec_face_events`: feature vectors extracted from faces detected in the video.
- `ffsec_face_objects`: feature vectors extracted from face images uploaded to the record index.
- `ffsec_face_clusters`: centroids of face clusters.

The database backup/recovery functionality allows you to fully restore all the galleries when needed.

To avoid data loss, we recommend you back up a feature vector database at least once a week. Overall, the backups' frequency depends on the number of records and face recognition events, and available disk space.

Be sure to back up the database before *migrating* your system to another neural network model.

Utilities

To back up and recover the FindFace feature vector database, the following utilities are needed:

1. backup: `findface-storage-api-dump`,
2. recovery: `findface-storage-api-restore`.

These utilities are automatically installed along with `findface-sf-api`.

Back Up Database

To back up the feature vector database, use the `findface-storage-api-dump` utility as follows:

Important: The following services must be active: `findface-tarantool-server`, `findface-sf-api`.

Note: The backup functionality can be applied to a distributed database. In this case, the `findface-storage-api-dump` utility will back up galleries on all the shards specified in `/etc/findface-sf-api.ini`.

1. On the server with `findface-sf-api`, create a directory to store the backup files (`/etc/findface_dump` in the example below).
2. Launch the `findface-storage-api-dump` utility by executing:

```
sudo findface-storage-api-dump -output-dir=/etc/findface_dump -config /etc/findface-  
↪sf-api.ini
```

The utility will back up at once all the galleries into the files with corresponding names (`ffsec_body_events.json`, `ffsec_face_events`, etc.) and save them into the directory. These files contain all the data needed to restore the entire database.

Recover Database

To recover the feature vector database from the backup, launch the `findface-storage-api-restore` utility for all the files in the backup folder:

```
sudo findface-storage-api-restore -config /etc/findface-sf-api.ini /etc/findface_dump/*.  
↪json
```

The recovery process can be interrupted and resumed whenever necessary. To resume the process after the interruption, launch the `findface-storage-api-restore` utility again.

Main Database Backup and Recovery (PostgreSQL)

To back up the main database `ffsecurity` based on PostgreSQL, execute:

```
sudo -u postgres pg_dump ffsecurity > ffsecurity_postgres_backup.sql
```

To recover the main database, do the following:

1. Stop the `findface-security` service.

```
sudo systemctl stop findface-security.service
```

2. Stop the `pgbouncer` service to delete its active sessions with the `ffsecurity` database.

```
sudo systemctl stop pgbouncer.service
```

3. Open the PostgreSQL interactive terminal.

```
sudo -u postgres psql
```

4. Remove the old `ffsecurity` database.

```
DROP DATABASE ffsecurity;
```

5. Create a new `ffsecurity` database. Leave the PostgreSQL interactive terminal.

```
CREATE DATABASE ffsecurity WITH OWNER ntech ENCODING 'UTF-8' LC_COLLATE='C.UTF-8'  
↪LC_CTYPE='C.UTF-8' TEMPLATE template0;
```

6. Start the `pgbouncer` service.

```
sudo systemctl start pgbouncer.service
```

7. Recover the database content from the backup.


```
sudo -u postgres psql -d ffsecurity -f ffsecurity_postgres_backup.sql
```

8. Migrate the database architecture from FindFace to **PostgreSQL**, re-create user groups with *predefined* rights and the first user with administrator rights.

```
sudo findface-security migrate
sudo findface-security create_groups
sudo findface-security create_default_user
```

9. Start the findface-security service.

```
sudo systemctl start findface-security.service
```

Artifacts Backup and Recovery (findface-upload)

The FindFace artifacts, such as photos uploaded to the record index, video files, and such event artifacts as full frames, thumbnails, and normalized face images, are stored in the following directories:

- /var/lib/findface-security/uploads
- /var/lib/ffupload/

Note: Both directories are operated by the findface-upload component.

To back up the artifacts, execute:

```
sudo tar -cvzf /home/some_directory/var_lib_ffsecurity_uploads.tar.gz /var/lib/findface-
↪security/uploads/
sudo tar -cvzf /home/some_directory/var_lib_ffupload.tar.gz /var/lib/ffupload/
```

To recover the artifacts, execute the following commands from the root directory:

```
cd /
sudo tar -xvf /home/some_directory/var_lib_ffsecurity_uploads.tar.gz
sudo tar -xvf /home/some_directory/var_lib_ffupload.tar.gz
```

Settings Backup and Recovery

The entire set of FindFace configuration files including the Tarantool structural schema is automatically backed up during the instance *removal*. It is saved to the `~/ffmulti_bak_${datetime}/etc/` directory.

When re-installing FindFace, recover the settings after completing the *console installation*, or right after installing services from the *APT repository*.

The entire set of backed up files is the following:

```
ls -R -p
.:
findface-counter.ini.bak      findface-liveness-api.ini.bak  findface-security/      ↵
↪findface-video-manager.conf.bak  findface-video-streamer-cpu.ini.bak  nginx/
findface-extraction-api.ini.bak  findface-ntls.cfg.bak          findface-sf-api.ini.bak ↵
↪findface-video-storage.conf.bak  findface-video-worker-cpu(gpu).ini.bak
```

(continues on next page)

(continued from previous page)

```
./findface-security:  
config.py tnt_schema.lua  
  
./nginx:  
sites-enabled/  
  
./nginx/sites-enabled:  
ffsecurity-nginx.conf
```

To recover the FindFace settings, do the following:

1. Open the `/etc/findface-security/config.py` configuration file of the fresh FindFace instance. Find the `DATABASES -> default -> PASSWORD` parameter that stores the ntech user password from the findface-security database of PostgreSQL. Copy/paste it to the `~/ffmulti_bak_${datetime}/etc/findface-security/config.py` backup.

```
sudo vi /etc/findface-security/config.py  
  
DATABASES = {  
    'default':  
        'PASSWORD': 'some_pass'
```

2. In the `~/ffmulti_bak_${datetime}/etc/` directory, use any method to eliminate the `.bak` extension from the files. For example, you can execute the following command.

```
sudo rename 's/.ini.bak/.ini/' * && sudo rename 's/.conf.bak/.conf/' * && sudo ↵  
↵rename 's/.cfg.bak/.cfg/' *
```

Tip: If the `rename` command is absent in your system, you can install it as follows:

```
sudo apt install rename
```

3. Recursively copy the backup files to the `/etc` directory.

```
sudo cp -r * /etc
```

4. Modify the database structure by applying the initial `tnt_schema.lua` file.

```
sudo findface-security make_tnt_schema | sudo tee /etc/findface-security/tnt_schema.  
↵lua
```

5. Restart the services.

On CPU:

```
sudo systemctl restart findface-counter findface-liveness-api findface-video-
↪manager findface-extraction-api findface-ntls findface-sf-api findface-video-
↪worker-cpu findface-security
```

On GPU:

```
sudo systemctl restart findface-counter findface-liveness-api findface-video-
↪manager findface-extraction-api findface-ntls findface-sf-api findface-video-
↪worker-gpu findface-security
```

Back Up and Recover Entire System

If you intend to back up FindFace before uninstalling it, it will be sufficient to follow the step-by-step instructions in the *Remove FindFace Instance* section. The provided `findface_uninstall.sh` script can automatically back up the FindFace configuration files and all data storages to the `~/ffmulti_bak_{{datetime}}/` directory.

To recover FindFace after uninstalling it, use the following algorithm:

1. *Deploy FindFace.*
2. *Recover the settings from the backed up configuration files.*
3. *Recover Tarantool.*
4. *Recover PostgreSQL.*
5. *Recover the artifacts.*

1.5.2 Migrate Feature Vectors to Different Neural Network Model

Tip: Do not hesitate to contact our experts on migration by support@ntechlab.com.

This section is about how to migrate face feature vectors to another neural network model.

Do the following:

1. Create a backup of the Tarantool-based feature vector database in any directory of your choice, for example, `/etc/findface_dump`.

Tip: See *Back Up and Recover FindFace and Data* for details.

```
sudo mkdir -p /etc/findface_dump
sudo cd /etc/findface_dump
sudo findface-storage-api-dump -config /etc/findface-sf-api.ini
```

2. Stop the `findface-sf-api` service.

```
sudo systemctl stop findface-sf-api.service
```

3. Create new shards that will host regenerated feature vectors.

1. Open the `/etc/tarantool/instances.available/` directory and find out the number of shards by counting the number of configuration files `shard-*.lua`.

Note: There are four shards in the example below.

```
cd /etc/tarantool/instances.available/  
  
ls -l  
  
shard-001.lua  
shard-002.lua  
shard-003.lua  
shard-004.lua
```

2. Create the same number of new shards by copying the configuration files `shard-*.lua`.

Note: For convenience, the second digit in the new names is 1: `shard-01*.lua`.

```
sudo cp shard-001.lua shard-011.lua  
sudo cp shard-002.lua shard-012.lua  
sudo cp shard-003.lua shard-013.lua  
sudo cp shard-004.lua shard-014.lua
```

3. Modify the following lines in each new shard's configuration file, subject to its name (`shard-011`, `shard-012`, etc., in our example):

Old value	New value
<code>listen = '127.0.0.1:32001'</code>	<code>Listen = '127.0.0.1:32011'</code>
<code>vinyl_dir = '/opt/ntech/var/lib/tarantool/shard-001'</code>	<code>vinyl_dir = '/opt/ntech/var/lib/tarantool/shard-011'</code>
<code>work_dir = '/opt/ntech/var/lib/tarantool/shard-001'</code>	<code>work_dir = '/opt/ntech/var/lib/tarantool/shard-011'</code>
<code>memtx_dir = '/opt/ntech/var/lib/tarantool/shard-001/snapshots'</code>	<code>memtx_dir = '/opt/ntech/var/lib/tarantool/shard-011/snapshots'</code>
<code>wal_dir = '/opt/ntech/var/lib/tarantool/shard-001/xlogs'</code>	<code>wal_dir = '/opt/ntech/var/lib/tarantool/shard-011/xlogs'</code>
<code>FindFace.start("127.0.0.1", 8101, {</code>	<code>FindFace.start("127.0.0.1", 8111, {</code>

4. Create symbolic links to the new shards.

```
cd /etc/tarantool/instances.enabled/  
  
sudo ln -s /etc/tarantool/instances.available/shard-01*.lua /etc/tarantool/  
instances.enabled/
```

5. Create directories that will host files of the new shards. Assign permissions for the created directories.

```
cd /opt/ntech/var/lib/tarantool/

mkdir -p shard-01{1..4}/{index,snapshots,xlogs}

chown tarantool:tarantool shard-01* shard-01*/*
```

4. Open the `/etc/findface-extraction-api` configuration file and replace the extraction model with the new one in the `face_emben` parameter.

```
sudo vi /etc/findface-extraction-api.ini

extractors:
...
models:
...
face_emben: face/<new_model_face>.gpu.fnk
```

Restart the `findface-extraction-api` service.

```
sudo systemctl restart findface-extraction-api.service
```

5. Start the new shards.

```
for i in {11..14}; do sudo systemctl start tarantool@shard-0$i; done
```

6. Create a configuration file with migration settings `<migration.ini>` based on the example below.

```
extraction-api:
  timeouts:
    connect: 5s
    response_header: 30s
    overall: 35s
    idle_connection: 0s
  extraction-api: http://127.0.0.1:18666
storage-api-from: # current location of the gallery
  timeouts:
    connect: 5s
    response_header: 30s
    overall: 35s
    idle_connection: 10s
  max-idle-conns-per-host: 20
  shards:
    - master: http://127.0.0.1:8101/v2/
      slave: ""
    - master: http://127.0.0.1:8102/v2/
      slave: ""
    - master: http://127.0.0.1:8103/v2/
      slave: ""
    - master: http://127.0.0.1:8104/v2/
      slave: ""
storage-api-to:
  timeouts:
    connect: 5s
```

(continues on next page)

(continued from previous page)

```

response_header: 30s
overall: 35s
idle_connection: 10s
max-idle-conns-per-host: 20
shards:
  - master: http://127.0.0.1:8111/v2/
    slave: ""
  - master: http://127.0.0.1:8112/v2/
    slave: ""
  - master: http://127.0.0.1:8113/v2/
    slave: ""
  - master: http://127.0.0.1:8114/v2/
    slave: ""
workers_num: 3
faces_limit: 100
extraction_batch_size: 8
normalized_storage:
  type: webdav
  enabled: True
  webdav:
    upload-url: http://127.0.0.1:3333/uploads/
  s3:
    endpoint: ""
    bucket-name: ""
    access-key: ""
    secret-access-key: ""
    secure: False
    region: ""
    public-url: ""
    operation-timeout: 30

```

In the `storage-api-from` section, specify the old shards to migrate the data from.

```

storage-api-from: # current location of the gallery
...
shards:
  - master: http://127.0.0.1:8101/v2/
    slave: ""
  - master: http://127.0.0.1:8102/v2/
    slave: ""
  - master: http://127.0.0.1:8103/v2/
    slave: ""
  - master: http://127.0.0.1:8104/v2/
...

```

In the `storage-api-to` section, specify the new shards that will host migrated data.

```

storage-api-to:
...
shards:
  - master: http://127.0.0.1:8111/v2/
    slave: ""

```

(continues on next page)

(continued from previous page)

```

- master: http://127.0.0.1:8112/v2/
  slave: ""
- master: http://127.0.0.1:8113/v2/
  slave: ""
- master: http://127.0.0.1:8114/v2/
  slave: ""
...

```

7. Launch the `findface-sf-api-migrate` utility with the `-config` option and provide the `<migration.ini>` configuration file.

```
findface-sf-api-migrate -config migration.ini
```

Note: The migration process can take up a significant amount of time if there are many events and records in the system.

8. After the migration is complete, stop the old shards and disable their autostart in OS (do not remove them).

```

for i in {01..04}; do sudo systemctl stop tarantool@shard-0$i.service ; done
for i in {01..04}; do sudo systemctl disable tarantool@shard-0$i.service ; done

```

9. Open the `/etc/findface-sf-api.ini` configuration file and adjust the shards ports, subject to the new shards settings. Restart the `findface-sf-api` service.

```

sudo vi /etc/findface-sf-api.ini

shards:
- master: http://127.0.0.1:8111/v2/
  slave: ""
- master: http://127.0.0.1:8112/v2/
  slave: ""
- master: http://127.0.0.1:8113/v2/
  slave: ""
- master: http://127.0.0.1:8114/v2/
  slave: ""

sudo systemctl start findface-sf-api.service

```

10. Import the database structure from the `tnt_schema.lua` file.

```

sudo findface-security make_tnt_schema | sudo tee /etc/findface-security/tnt_schema.
↪lua

```

See also:

Modify Feature Vector Database Structure.

11. Migrate the main database architecture from FindFace to **PostgreSQL**, re-create *predefined* user roles and the first administrator.

```
sudo findface-security migrate
sudo findface-security create_groups
sudo findface-security create_default_user
```

12. Restart the services.

```
sudo systemctl restart findface-security.service
sudo systemctl restart findface-extraction-api findface-video-worker* findface-
↪video-manager findface-sf-api
```

1.5.3 Modify Feature Vector Database Structure

Sometimes it may be necessary to apply a new structural schema to your Tarantool-based feature vector database, for example, when updating to the latest version of the product, or when you want to enhance the default database structure with additional parameters, advanced face metadata, and so on.

In this section:

- [About Database Structure](#)
- [Structure Modification](#)

About Database Structure

In FindFace, the database structure is set via the `/etc/findface-security/tnt_schema.lua` file.

The structure is created as a set of spaces and fields. Each field is described with the following parameters:

- `id`: field id;
- `name`: field name, must be the same as the name of a relevant object parameter;
- `field_type`: data type;
- `default`: field default value. If a default value exceeds `'1e14 - 1'`, use a string data type to specify it, for example, `"123123.."` instead of `123123...`

You can find the default `tnt_schema.lua` file [here](#).

Structure Modification

To modify the database structure, do the following:

1. Stop the `findface-security` service.

```
sudo systemctl stop findface-security.service
```

2. Create a backup of the Tarantool-based feature vector database in any directory of your choice, for example, `/etc/findface_dump`.

Tip: See [Back Up and Recover FindFace and Data](#) for details.

```
mkdir -p /etc/findface_dump
cd /etc/findface_dump
sudo findface-storage-api-dump -config /etc/findface-sf-api.ini
```

3. Prepare the `tnt_schema.lua` file containing the new database structure.
4. Modify the database structure by applying the new `tnt_schema.lua` file.

```
sudo findface-security make_tnt_schema | sudo tee /etc/findface-security/tnt_schema.
↵lua
```

5. Navigate to the directory with Tarantool configuration file(s) `/etc/tarantool/instances.available/`. For each shard, check whether it contains the `dofile` command and the `spaces` definition, as in the example below.

```
sudo vi /etc/tarantool/instances.available/<shard_00N>.lua

dofile("/etc/findface-security/tnt_schema.lua")

-- host:port to bind, HTTP API
FindFace = require("FindFace")
FindFace.start("127.0.0.1", 8101, {
license_ntls_server="127.0.0.1:3133",
replication = replication_master,
spaces = spaces
})
```

6. Stop the `findface-tarantool-server` shards. Purge data from all the directories relevant to active shards.

```
sudo systemctl stop 'tarantool@*'

sudo rm /opt/ntech/var/lib/tarantool/shard-*/{index,snapshots,xlogs}/*
```

7. Restart the `findface-tarantool-server` shards.

```
TNT=$(ls /etc/tarantool/instances.enabled/ | cut -c 7,8,9)
for i in $TNT; do sudo systemctl restart tarantool@shard-$i.service ; done
```

8. Restore the Tarantool database from the backup.

Important: If some fields were removed from the new database structure, you have to first manually delete the corresponding data from the backup copy.

```
cd /etc/findface_dump
for x in *.json; do curl -X POST "http://127.0.0.1:18411/v2/galleries/${x%.json}";
↵done
for x in *.json; do sudo findface-storage-api-restore -config /etc/findface-sf-api.
↵ini < "$x"; done
```

9. Start the `findface-security` service.

```
sudo systemctl start findface-security.service
```

See also:

Custom Metadata in Tarantool

1.5.4 Check Component Status

Check the status of components once you have encountered a system problem.

Component	Command to view service status
findface-extraction-api	sudo systemctl status findface-extraction-api.service
findface-sf-api	sudo systemctl status findface-sf-api.service
findface-tarantool-server	sudo systemctl status tarantool.service
findface-tarantool-server shards	sudo systemctl status tarantool@shard-00*
findface-video-manager	sudo systemctl status findface-video-manager.service
findface-video-worker	sudo systemctl status findface-video-worker*.service
findface-ntls	sudo systemctl status findface-ntls
findface-security	sudo systemctl status findface-security.service
findface-counter	sudo systemctl status findface-counter.service
etcd	sudo systemctl status etcd.service
NginX	sudo systemctl status nginx.service
memcached	sudo systemctl status memcached.service
postgresql	sudo systemctl status postgresql*
nats	sudo systemctl status nats.service
pgbouncer	sudo systemctl status pgbouncer.service

1.5.5 Service Logs

Service logs provide a complete record of each FindFace component activity. Consulting logs is one of the first things you should do to identify a cause for any system problem.

In this section:

- *Configure Logging*
- *Consult Service Logs*

Configure Logging

The FindFace services log a large amount of data, which can eventually lead to disc overload. To prevent this from happening, we advise you to disable `rsyslog` due to its suboptimal log rotation scheme and use the appropriately configured `systemd-journal` service instead.

Do the following:

1. Check whether the `/var/log/journal` directory already exists. If not, create it by executing the following command:

```
sudo mkdir /var/log/journal
sudo chmod 2755 /var/log/journal
```

2. Open the `/etc/systemd/journald.conf` configuration file. Enable saving `journald` logs to your hard drive by uncommenting the `Storage` parameter and changing its value to `persistent`. Disable filtering in `systemd-journal` as well:

```
sudo vi /etc/systemd/journald.conf

[Journal]
...
Storage=persistent
...
RateLimitInterval=0
RateLimitBurst=0
...
```

If necessary, uncomment and edit the `SystemMaxUse` parameter. This parameter determines the maximum volume of log files on your hard drive. Specify its value in bytes or use K, M, G, T, P, E as units for the specified size (equal to 1024 , 1024^2 , ... bytes).

```
...
SystemMaxUse=3G
```

3. Restart the `journald` service.

```
sudo systemctl restart systemd-journald.service
```

4. Stop and disable the `syslog` service.

```
sudo systemctl stop syslog.socket rsyslog.service
sudo systemctl disable syslog.socket rsyslog.service
```

5. If necessary, delete the existing log files created through `syslog`, and the kernel logs.

```
sudo rm /var/log/syslog*
sudo rm /var/log/kern.log*
```

Consult Service Logs

Use the `journalctl -u <component>` command to consult a component log, for example as follows:

```
journalctl -u findface-extraction-api
```

See also:

Audit Log

1.5.6 Troubleshoot Licensing and findface-ntls

When troubleshooting licensing and `findface-ntls` (see *Licensing*), the first step is to retrieve the licensing information and `findface-ntls` status. You can do so by sending an API request to `findface-ntls`. Necessary actions are then to be undertaken, subject to the response content.

Tip: Please do not hesitate to contact our experts on troubleshooting by support@ntechlab.com.

Note: The online licensing is done via the NtechLab Global License Manager `license.ntechlab.com`. Check its availability. A stable internet connection and DNS are required.

To retrieve the FindFace *licensing* information and `findface-ntls` status, execute on the `findface-ntls` host console:

```
curl http://localhost:3185/license.json -s | jq
```

The response will be given in JSON. One of the most significant parameters is `last_updated`. It indicates in seconds how long ago the local license has been checked for the last time.

Interpret the `last_updated` value as follows:

- [0, 5] — everything is alright.
- (5, 30] — there may be some problems with connection, or with the local drive where the license file is stored.
- (30; 120] — almost certainly something bad happened.
- (120; ∞) — the licensing source response has been timed out. Take action.
- "valid" -> "value": false: connection with the licensing source was never established.

```
curl http://localhost:3185/license.json -s | jq
{
  "name": "NTLS",
  "time": 1565186356,
  "type": "online",
  "license_id": "61063ce4b86945e1b70c3bdbedea453b",
  "generated": 1514467939,
  "last_updated": 5,
  "valid": {
    "value": true,
    "description": ""
  },
  "source": "/opt/ntech/license/import_
↪b68d7b7ec9a7310d18832035318cff0c9ddf11e3a9ab0ae962fbe48645e196d1.lic",
  "limits": [
    {
      "type": "time",
      "name": "end",
      "value": 1609161621
    },
    {
      "type": "number",
      "name": "faces",

```

(continues on next page)

(continued from previous page)

```
"value": 9007199254740991,
"current": 0
},
{
  "type": "number",
  "name": "cameras",
  "value": 4294967295,
  "current": 0
},
{
  "type": "number",
  "name": "extraction_api",
  "value": 256,
  "current": 0
},
{
  "type": "boolean",
  "name": "gender",
  "value": true
},
{
  "type": "boolean",
  "name": "age",
  "value": true
},
{
  "type": "boolean",
  "name": "emotions",
  "value": true
},
{
  "type": "boolean",
  "name": "fast-index",
  "value": true
},
{
  "type": "boolean",
  "name": "sec-genetec",
  "value": false
},
{
  "type": "boolean",
  "name": "beard",
  "value": false
},
{
  "type": "boolean",
  "name": "glasses",
  "value": false
},
{
  "type": "boolean",
```

(continues on next page)

```

    "name": "liveness",
    "value": false
  }
],
"services": [
  {
    "name": "video-worker",
    "ip": "127.0.0.1:53276"
  },
  {
    "name": "FindFace-tarantool",
    "ip": "127.0.0.1:53284"
  },
  {
    "name": "FindFace-tarantool",
    "ip": "127.0.0.1:53288"
  }
]
}

```

1.5.7 Manually Purge Old Data from Database

To manually remove old data from the FindFace database, use the `cleanup` utility. You can separately remove the following data:

- matched events,
- unmatched events,
- full frames of matched events,
- full frames of unmatched events,
- audit-logs.

To invoke the `cleanup` help message, execute:

```
sudo findface-security cleanup --help
```

To entirely remove events older than a given number of days, use the `--face-events-max-matched-age/--face-events-max-unmatched-age` options. For example, to remove unmatched events older than 5 days, execute:

```
sudo findface-security cleanup --face-events-max-unmatched-age 5
```

To remove only matched events older than 5 days, execute:

```
sudo findface-security cleanup --face-events-max-matched-age 5
```

The following commands remove only full frames of matched/unmatched events:

```
sudo findface-security cleanup --face-events-max-fullframe-matched-age 5
sudo findface-security cleanup --face-events-max-fullframe-unmatched-age 5
```

To remove only audit logs, execute:

```
sudo findface-security cleanup --audit-logs-max-age 5
```

1.5.8 Reset Password

To reset a user password to the FindFace web interface, execute the following command:

```
findface-security changepassword %username
```

1.5.9 Migrate Data to Another Disk

High disk load may lead to delays in event arrivals. In severe cases, it might result in complete inoperability of FindFace. One of the means for reducing the disk load is to migrate the FindFace data storages to another disk.

In this section:

- *Prepare Disk*
- *Migrate Photo Storage*
- *Migrate Main Database (PostgreSQL)*

Prepare Disk

To prepare a disk for the data migration, do the following:

1. Create a new mount point (/mnt/ffdata in our example).

```
sudo mkdir /mnt/ffdata
sudo chown ntech:ntech /mnt/ffdata
```

2. Create a partition.

```
sudo parted /dev/sdb
mklabel gpt
mkpart primary ext4 1MiB 100%
q
sudo mkfs.ext4 /dev/sdb1
```

3. Learn the UUID of the partition (sdb1 in our example).

```
sudo blkid | grep sdb1
/dev/sdb1: LABEL="data" UUID="0638ebe0-853e-43ea-8f35-bfae305695d1" TYPE="ext4"
↳PARTUUID="8cebaacc-77d7-4757-b4c6-14147e92646c"
```

4. Add the partition to fstab to make it automatically mount on booting.

```
sudo vi /etc/fstab
-----
#DATA mount
```

(continues on next page)

(continued from previous page)

```
UUID=0638ebe0-853e-43ea-8f35-bfae305695d1 /mnt/ffdata/ ext4 auto,user,rw
↪0 2
-----
```

5. Mount all the filesystems.

```
sudo mount -a
```

Migrate Photo Storage

To migrate the FindFace photo storage, do the following:

1. Stop the `findface-security` service to prevent the data loss.

```
sudo systemctl stop findface-security
```

2. By default, the photo data are stored at `/var/lib/`. Migrate the photo storage to the *new disk*.

```
sudo cp -ax /var/lib/findface-security/ -R /mnt/ffdata/
sudo rm -r /var/lib/findface-security/
sudo cp -ax /var/lib/ffupload/ -R /mnt/ffdata/
sudo rm -r /var/lib/ffupload/
```

3. Create symbolic links for the new directories.

```
sudo ln -s /mnt/ffdata/findface-security/ /var/lib/
sudo ln -s /mnt/ffdata/ffupload/ /var/lib/
```

4. Ensure that the rights are correctly assigned.

```
sudo chown ntech:ntech /mnt/ffdata/findface-security/
```

5. Start the `findface-security` service.

```
sudo systemctl start findface-security
```

Migrate Main Database (PostgreSQL)

To migrate the PostgreSQL database, do the following:

1. Stop the `findface-security`, `pgbouncer`, and PostgreSQL services.

```
sudo systemctl stop findface-security.service pgbouncer.service postgresql.service
↪postgresql@10-main.service
```

2. On the *new disk*, create a directory for the database.

```
sudo mkdir -p /mnt/ffdata/some_directory/db
```

3. Migrate the database to the new directory.

```
sudo mv /var/lib/postgresql /mnt/ffdata/some_directory/db
```


4. Create a symlink to the new directory.

```
sudo ln -s /mnt/ffdata/some_directory/db/postgresql /var/lib/postgresql
```

5. Start the PostgreSQL, pgbouncer, and findface-security services.

```
sudo systemctl start postgresql.service postgresql@10-main.service pgbouncer.service  
sudo systemctl start findface-security.service
```

1.6 Appendices

1.6.1 Installation File

FindFace installation configuration is automatically saved to a file `/tmp/<findface-installer-*.json`. You can edit this file and use it to install FindFace on other hosts without having to answer the installation questions again.

Tip: See *Deploy from Console Installer* to learn more about the FindFace installer.

Important: Be sure to remove fields `*.config`, `exp_ip`, and `int_ip` before installing FindFace on a host with a different IP address.

Here is an example of the installation file.

1.6.2 Neural Network Models

Here you can see a summary for neural network models created by our Lab and used in FindFace.

You can find installed models at `/usr/share/findface-data/models/`.

Important: The default face biometrics model upon a clean install is `mango_320`.

Face detection

```
ls /usr/share/findface-data/models/facedet/  
cheetah.cpu.fnk  cheetah_fast.cpu.fnk  cheetah_fast.gpu.fnk  cheetah.gpu.fnk
```

Face image normalization

```
ls /usr/share/findface-data/models/facenorm/

bee_fast.cpu.fnk  bee.v2.gpu.fnk  crop2x.v2_maxsize400.cpu.fnk  crop2x.v2_
↳no_maxsize.gpu.fnk
bee_fast.gpu.fnk  crop1x.v2_maxsize400.cpu.fnk  crop2x.v2_maxsize400.gpu.fnk  cropbbox.
↳v2.cpu.fnk
bee.v2.cpu.fnk  crop1x.v2_maxsize400.gpu.fnk  crop2x.v2_no_maxsize.cpu.fnk  cropbbox.
↳v2.gpu.fnk
```

Face recognition

```
ls /usr/share/findface-data/models/face/

lime.v2.cpu.fnk  lime.v2.gpu.fnk  mango_320.cpu.fnk  mango_320.gpu.fnk
```

Face attribute recognition

```
ls /usr/share/findface-data/models/faceattr/

age.v2.cpu.fnk  beard.v0.gpu.fnk  gender.v2.cpu.fnk  glasses3.v0.gpu.fnk  ↳
↳liveness.colombo.cpu.fnk  liveness.pacs.v0.gpu.fnk  quality.v1.cpu.fnk
age.v2.gpu.fnk  emotions.v1.cpu.fnk  gender.v2.gpu.fnk  headpose.v2.cpu.fnk  ↳
↳liveness.colombo.gpu.fnk  medmask3.v2.cpu.fnk  quality.v1.gpu.fnk
beard.v0.cpu.fnk  emotions.v1.gpu.fnk  glasses3.v0.cpu.fnk  headpose.v2.gpu.fnk  ↳
↳liveness.pacs.v0.cpu.fnk  medmask3.v2.gpu.fnk
```

1.6.3 FindFace Data Storages

In this section:

- [List of Storages](#)
- [Feature Vector Database Galleries](#)

List of Storages

FindFace uses the following data storages:

- Tarantool-based feature vector database that stores face feature vectors and recognition events.
- PostgreSQL-based main system database `ffsecurity` that stores internal system data, records, and user accounts.
- The `/var/lib/findface-security/uploads` directory that stores photos uploaded to records, video files, full frames of events, and thumbnails.
- The `/var/lib/ffupload/` directory that stores such event artifacts as normalized face images.

Feature Vector Database Galleries

There are the following galleries in the Tarantool-based feature vector database:

- `ffsec_face_events`: feature vectors extracted from faces detected in the video.
- `ffsec_face_objects`: feature vectors extracted from face images uploaded to the card index.
- `ffsec_face_clusters`: centroids of face clusters.

USER'S GUIDE

This chapter describes how to work with the FindFace web interface, including its advanced possibilities, and will be of interest to administrators, operators, and other users.

2.1 Getting Started

Once you have successfully *deployed and configured* FindFace, it's time to open the web interface, and get started. In this chapter, you can find a recommended sequence of steps that will help you harness your system's complete functionality.

Organize Watch Lists and Global Record Index

1. *Create a new watch list* or use the default one. A watch list is an entity that allows you to classify individuals by arbitrary criteria, e.g., wanted, suspects, etc.
2. Upload records of individuals to the global record index and add them to watch lists. See *Record Index*.

Connect FindFace to Remote Systems

Integrate FindFace with *remote facial recognition systems*. In this case, the server known as a puppeteer will be pushing record index data to remote servers known as puppets. In return, it will be pulling recognition events matching with the records.

FindFace in Action

1. *Create case files* and process video footages from crime scenes to define participants.
2. *Search for faces* across the system.
3. *Compare two faces* and verify that they belong to the same individual.
4. *Build* detailed reports on search results and record index.
5. *Harness* the FindFace comprehensive and searchable audit log to enhance your system protection.
6. View alerts from remote facial recognition systems and search remote systems for specific individuals. See *Remote Alerting and Remote Search*.

Basic Maintenance

1. Manually *purge* old data.
2. Regularly *backup* the database.

Go Further

Harness the FindFace functions through *HTTP API*.

2.2 Record Index

Record index stores records of individuals, including their biometric data, related documents, links to relevant cases, and other important data.

To create records in bulk, use the *console bulk record upload* functionality.

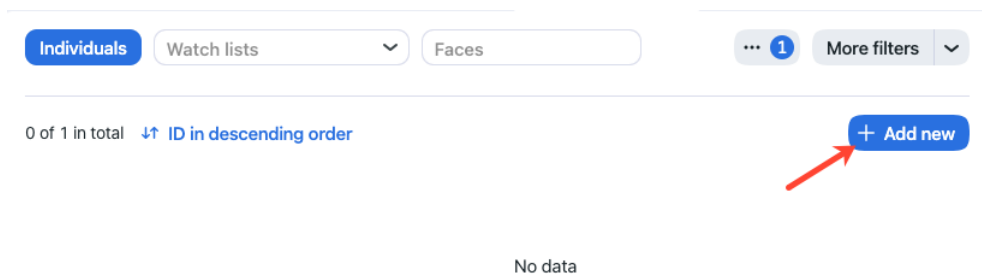
In this section:

- *Create Record*
- *Console Bulk Record Upload*
- *Autopopulation with Criminal Investigation Artifacts*
- *Filter Records*
- *Purge Record Index*

2.2.1 Create Record

To create a record manually, do the following:

1. Navigate to the *Record Index* tab.
2. Click + *Add new*.



3. Specify the individual's name. If necessary, add a comment.
4. From the *Watch lists* drop-down menu, select a watch list for the record (or several watch lists, one by one).
5. Check *Active*. If a record is inactive, it is excluded from the *case analysis, remote alerting and search*.

6. Save the record. You will see additional tabs appear.
7. On the same tab *Info*, attach related files.

The screenshot shows the 'Info' tab for a record titled 'Mrs Smith'. The record is associated with a user 'Mrs S...' (initials M). The 'Info' tab is selected, showing fields for Name (Mrs Smith), Watch lists (Hitmen), Comment, and an 'Active' checkbox. There are no files attached, with a link to 'Attach First One'. The record ID is 1, and it was created on 2022-09-01 at 14:38:11. At the bottom, there is a toolbar with a checkmark, a '1' in a blue circle, a close button, and a menu icon.

8. On the *Faces* tab, attach images of the individual's face. Supported formats: WEBP, JPG, BMP, PNG.

The screenshot shows the 'Faces' tab for the same 'Mrs Smith' record. The 'Faces' tab is selected, displaying a list of faces. The list is sorted by 'ID in descending order'. An 'Upload Photo' button is visible. A single face image is shown, which is blurred, with the ID 4463338839474982766 displayed below it. The bottom toolbar is identical to the previous screenshot.

9. Save the record.

2.2.2 Console Bulk Record Upload

You can bulk-upload records to the record index via the **findface-security-uploader** console utility.

Tip: To view the **findface-security-uploader** help, execute:

```
findface-security-uploader --help

Usage: findface-security-uploader [OPTIONS] COMMAND [ARGS]...

Options:
  --job PATH          Job file (default: enroll-job.db)
  --log-level TEXT    Log level
  --fsync BOOLEAN     Call fsync() to prevent data loss on power failure
  --help             Show this message and exit.

Commands:
  add  Add items from CSV or TSV file to job
  print Print contents of job file as JSON
  run  Run upload job
```

```
findface-security-uploader add --help

Usage: findface-security-uploader add [OPTIONS] FILES...

Options:
  --format [csv|tsv]  Input file format - CSV or TSV
  --delimiter TEXT    Field delimiter - by default it's "\t" for TSV and ","
                    for CSV
  --help             Show this message and exit.
```

```
findface-security-uploader print --help

Usage: findface-security-uploader print [OPTIONS]

Print contents of job file as JSON

Options:
  --failed  Show only failed images
  --noface  Show only images without detection
  --help    Show this message and exit.
```

```
findface-security-uploader run --help

Usage: uploader.py run [OPTIONS]

Run upload job

Options:
  --parallel INTEGER  Number of enroll threads (default: 10)
  --api TEXT          API url (default: http://127.0.0.1:80/) [required]
  --user TEXT         API username [required]
```

(continues on next page)

(continued from previous page)

```

--password TEXT          API password [required]
--watch-lists TEXT      Comma-separated list of card list ids [required]
--inactive              Mark new cards as inactive
--failed               Include failed images
--noface               Include images without detection
--all-faces            Enroll all found faces on each image
--logging-delta INTEGER Logging period delta
--help                Show this message and exit.

```

Do the following:

1. Write the list of photos and metastrings to a CSV or TSV file.

Important: The file used as a metadata source must have the following format: path to photo | metastring.

To prepare a TSV file, use either a script or the `find` command.

Note: Both the script and the command in the examples below create the `images.tsv` file. Each image in the list will be associated with a metastring coinciding with the image file name in the format path to photo | metastring.

To build a TSV file listing photos from a specified directory (`/home/user/25_celeb/` in the example below), run the following command:

```
python3 tsv_builder.py /home/user/25_celeb/
```

The `find` usage example:

```
find photos/ -type f -iname '*g' | while read x; do y="${x%.*}"; printf "%s\t%s\n" "$x" "${y##*/}"; done
```

2. Create a job file out of a CSV or TSV file by using `add`. As a result, a file `enroll-job.db` will be created and saved in a current directory.

```
findface-security-uploader add images.tsv
```

The `add` options:

- `--format`: input file format, `tsv` by default,
- `--delimiter`: field delimiter, by default `"\t"` for TSV, and `","` for CSV.

Note: A job file represents a sqlite database which can be opened on the `sqlite3` console.

3. Process the job file by using `run`.

```
findface-security-uploader run --watch-lists 2 --api http://127.0.0.1:80 --user_
↪admin --password password
```

The important `run` options:

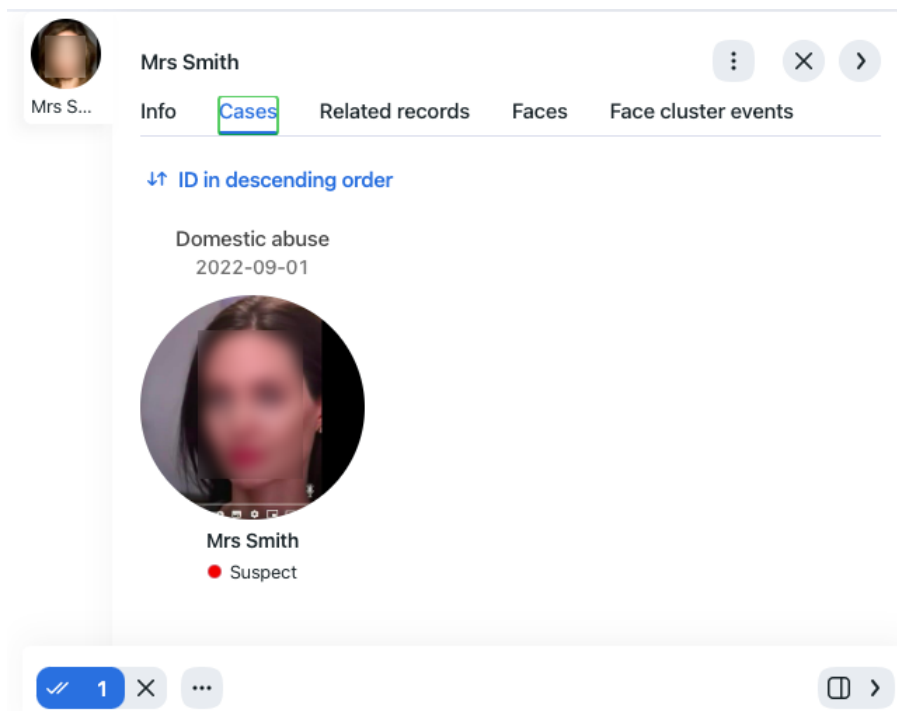
- `--parallel`: the number of photo upload threads, 10 by default. The more threads you use, the faster the bulk upload is completed, however it requires more resources too.
 - `--all-faces`: upload all faces from a photo if it features several faces.
 - `--api`: `findface-security` API URL, `http://127.0.0.1:80/` by default. Mandatory option.
 - `--user`: login. Mandatory option.
 - `--password`: password. Mandatory option.
 - `--watch-lists`: comma-separated list of the watch lists id's. Mandatory option.
 - `--failed`: should an error occur during the job file processing, correct the mistake and try again with this option.
 - `--inactive`: mark new records as inactive.
 - `--noface`: by default, images classified as having no faces will be assigned the `NOFACE` status and automatically excluded from the upload. To attempt re-detecting faces in such images, re-run the job file with this option. If the re-detection gives a negative result again, an image will be skipped and a relevant record will appear in the upload log.
4. (Optional) Print the job processing results as JSON. If necessary, you can print only failed images/ images without detected faces.

```
findface-security-uploader print --failed  
findface-security-uploader print --noface
```

2.2.3 Autopopulation with Criminal Investigation Artifacts

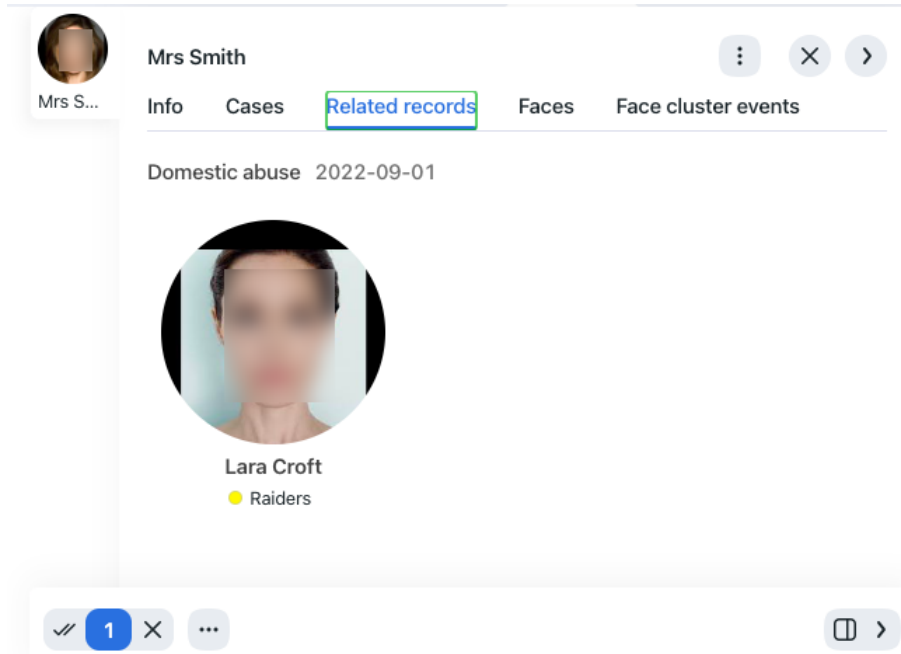
While you are performing your duties using FindFace, investigating cases and analyzing CSI footage, a record is automatically populated with the following data:

- Related cases on the *Cases* tab.



To establish such a link, you need to bind the record to a relevant participant during *a case investigation*.

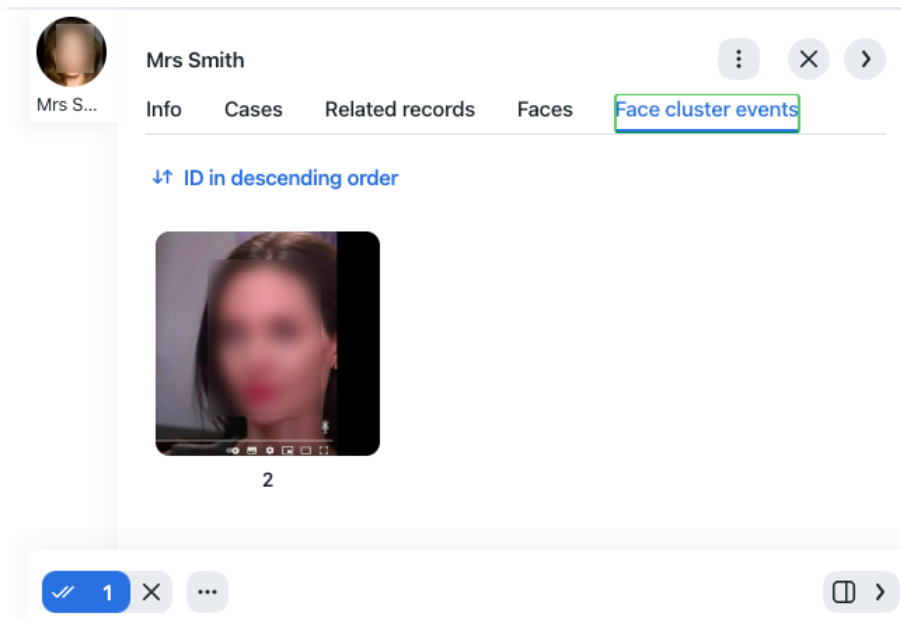
- Related records on the same-name tab.



The first way this tab is populated is when a perpetrator operates under different guises, and you create separate records to address each of them. If you link a case participant to several records, they will automatically get interlinked.

The second way is when a case has several participants linked to records. Each participant's record will store links to the records of the case's other participants.

- Detected faces on the *Face cluster events* tab. All face detection events of an individual from the cases linked to this record will be shown on this tab.

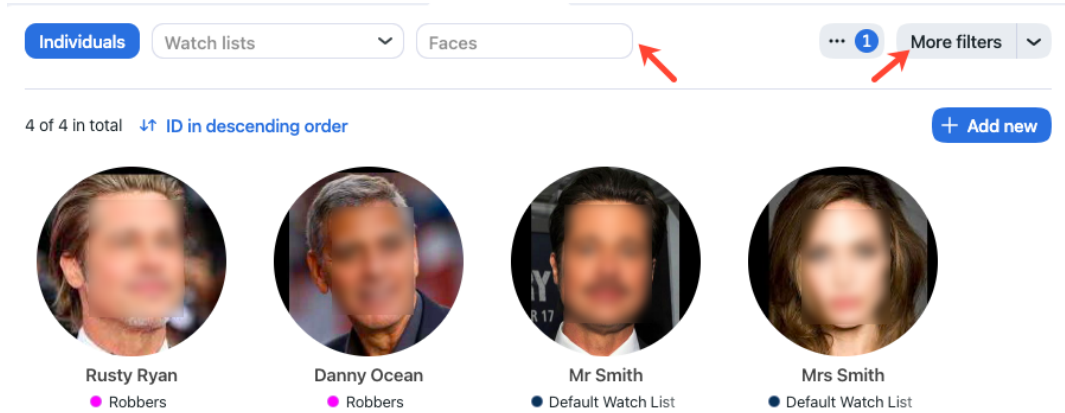


2.2.4 Filter Records

The most frequently used filters for the record index are available in the upper part of the window.

To display the entire set of filters, click the *More filters* button. Here it is:

- *Watch lists*: display records from selected watch lists.
- *Faces*: filter records by presence of a face biometric data.
- *Name*: filter records by name.
- *ID*: display a record with a given ID.



You can sort out records on the list by *Created data*.

2.2.5 Purge Record Index

You can purge the record index entirely or by watch lists in one click. Do the following:

1. Navigate *Settings* -> *Watch Lists*.
2. Select one or several watch lists.
3. Click *Delete records in selected*.

4 in total

Search

+ Add new watch list

Name	Active
<input checked="" type="checkbox"/> Raiders	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> Hitmen	<input checked="" type="checkbox"/>
<input type="checkbox"/> Unmatched	<input type="checkbox"/>
<input type="checkbox"/> Default Watch List	<input type="checkbox"/>

Build version: 5.2.999.2223+1745.gc329f16aca
Build date: 2022-08-31 19:53:54

2

Deactivate selected

2.3 Case Files

FindFace allows for conducting case management and performing investigation by analyzing associated video footage. This functionality is available on the *Case Files* tab.

In this section:

- *Video File Formats*
- *Case Investigation Workflow*
- *Create Case File*
- *Case Access Permissions and Related Documents*
- *Upload and Process Video File*
- *Detected People Analysis*
- *Case Participant Records*
- *Case File Archive*

2.3.1 Video File Formats

Video footage used for case investigations is accepted in a wide variety of formats. Click [here](#) for listing.

2.3.2 Case Investigation Workflow

Video-based case investigation is conducted in the following way:

1. Create a new case file. Specify the incident date, the case ID in a registry, and the registration date.
2. Specify case details and set up access permissions for it.
3. Upload a video footage from the criminal scene. Set up video processing parameters if needed. Process the video. The system will return human faces detected in it.
4. Parse the detected faces. Figure out which individual is likely to be a suspect, a victim, or a witness. Other individuals will be considered not relevant to the case. Link faces to matching records in the [global record index](#) and participants of other cases.
5. Fill in case participant records. Specify their names and attach related documents.
6. Keep returning to the case file to supplement it with new materials, as the investigation progresses.

2.3.3 Create Case File

To create a case file, do the following:

1. Navigate to *Case Files*.
2. Click + *Create case file*.

New case file

You are opening a new case file. Be sure to specify its name and number (ID), and the incident date

+ Create case file



3. Enter the descriptive name of the case. Specify the incident date.
4. Add a comment if required.
5. Specify the case ID and date in the registry.
6. Click *Save*. As a result, the case file will be added to the case list.

New case file

You are opening a new case file. Be sure to specify its name and number (ID), and the incident date

Name	Incident date
<input type="text" value="Domestic abuse"/>	<input type="text" value="01.09.2021"/>
Comment <input type="text"/>	
Case ID in registry	Case date
<input type="text" value="3462"/>	<input type="text" value="01.09.2021"/>
Create >	

2.3.4 Case Access Permissions and Related Documents

After you create a case, specify its details and set up access permissions. Do the following:

1. Click the case on the case list to open it.

All	Open	Archived	<input type="text" value="Search"/>	+ New case file			
<input type="checkbox"/>	Name	Creator	Created	Updated	Videos	Participants	Status
<input type="checkbox"/>	Domestic abuse	Charlie Root	2022-09-03 20:38:27	2022-09-03 20:38:27	0	0	open

2. Click *Set access permissions* to modify the default access permissions.
3. Attach one or several files relevant to the case.
4. Click *Save*.

Name Incident date

Comment

Case ID in registry Case date

3 in total

Name	View	Change	Delete
Administrator	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Operator	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
User	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

No Files [Attach First One](#)

ID 2
Created 2022-09-03 20:38:27 by Charlie Root

2.3.5 Upload and Process Video File


To upload and process a CSI video footage, do the following:

1. Open the case file.
2. Navigate to the *Sources* tab.
3. Click the + *Upload first video* button.
4. Specify a URL or select a file. Click *Upload*.

File URLs

Enter File URLs

Use Enter to add multiple URLs



Drag & drop Files to upload or
[Select files](#)

pittjolie.mp4 2.26Mb X

Upload

The video will be uploaded and shown in the source list.

- Click the video on the list to open the processing configuration wizard. Set up the video processing parameters on the following tabs:

- General.*

If necessary, change the video file name. Specify the video start time.

d
domestic_abuse.mp4
CHANGED X
▶
⋮
X
▶

domestic...

General
Advanced
Zones
Faces

Name

File size

2 Mb

Info

Camera group

Video archive default Came X

Camera (Optional)

▼

Detectors

Faces

Start time

01.09.202.

12:00:00

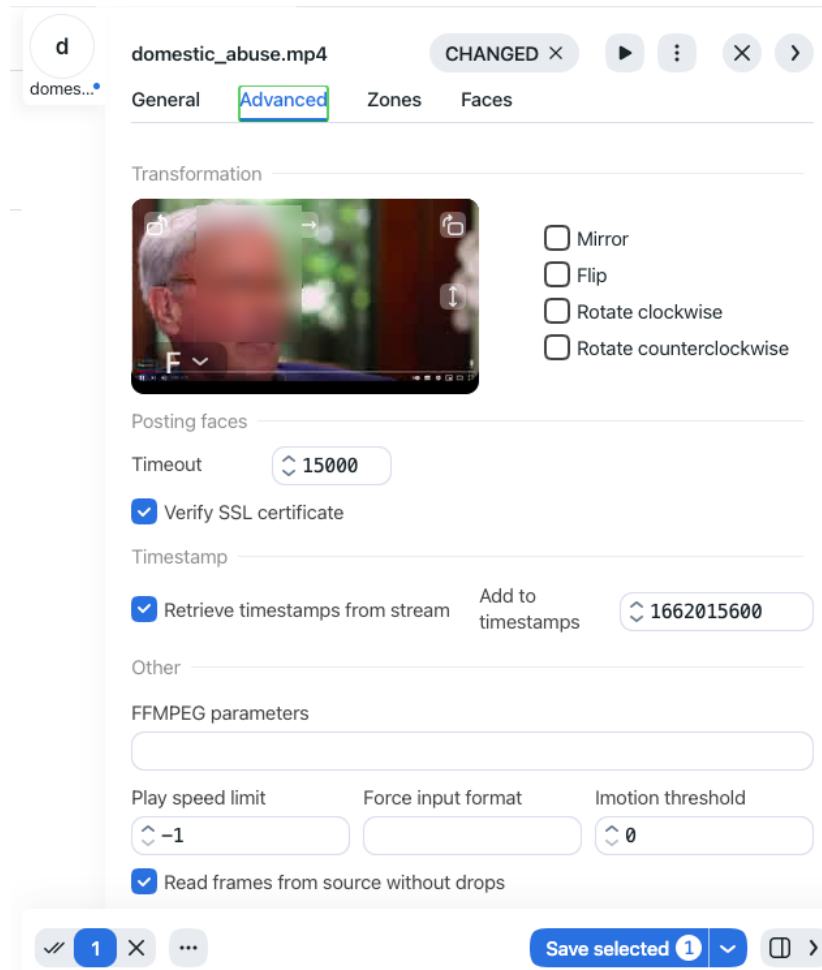
Now

Clear

✓ 1 X ...
Save selected 1
□ ▶

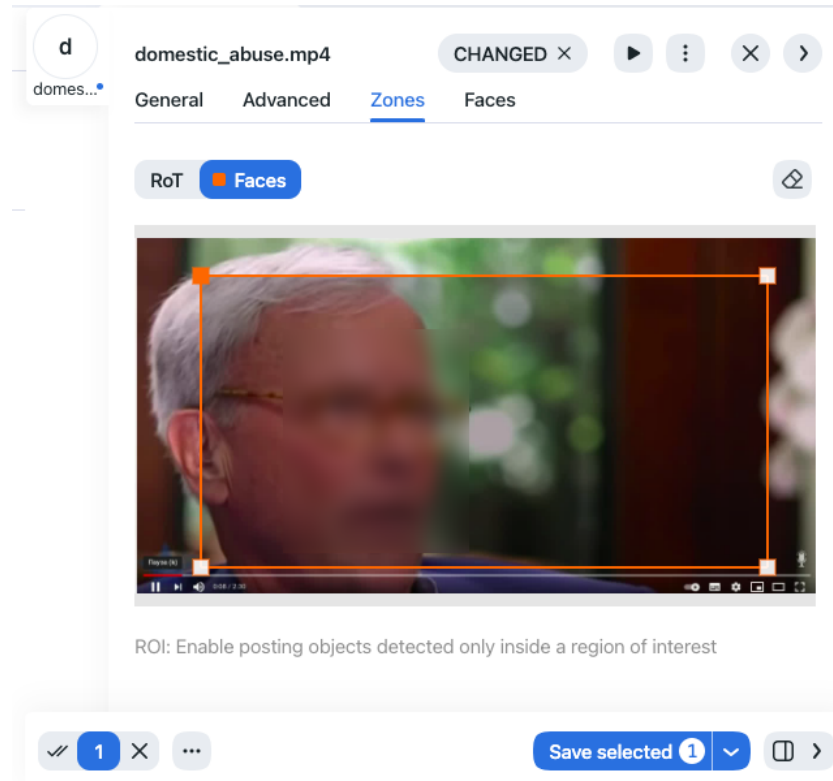
- Advanced.*

Fine-tune the video processing using the following parameters:



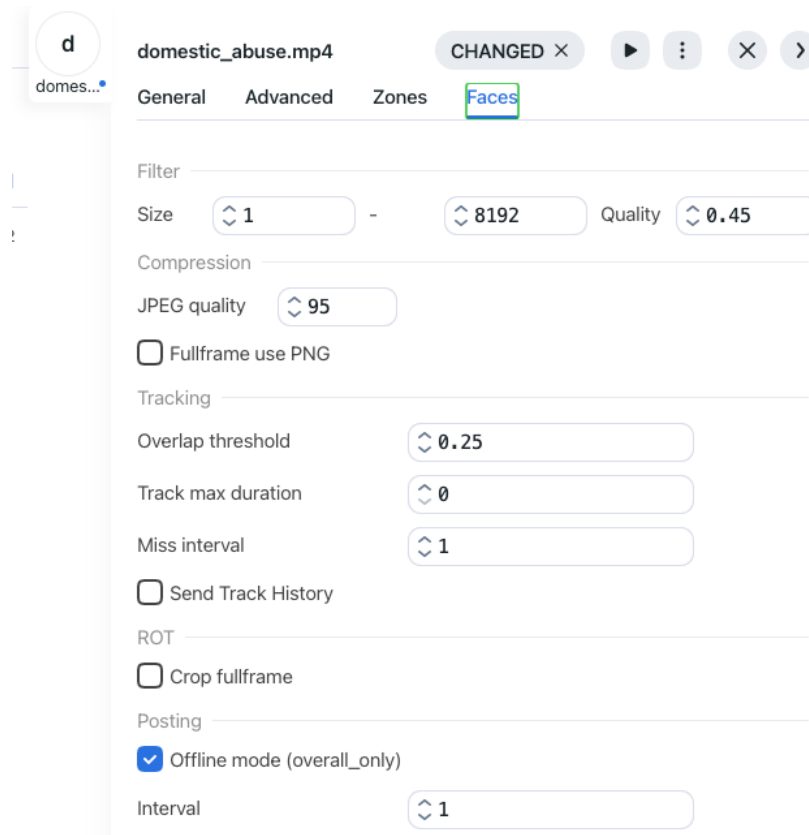
- If needed, change the video orientation.
 - *Timeout*: Specify the timeout in milliseconds for posting detected faces.
 - *Verify SSL*: Check to enable verification of the server SSL certificate when the video face detector posts faces to the server over https. Uncheck the option if you use a self-signed certificate.
 - *Retrieve timestamps from stream*: Enable to retrieve and post timestamps from a video stream. Disable to post the actual date and time.
 - *Add to timestamps*: Add the specified number of seconds to timestamps from a stream.
 - *FFmpeg parameters*: FFmpeg options for the video in the key-value format, for example, `rtsp_transport=tcp ss=00:20:00`.
 - *Play speed limit*: If less than zero, the speed is not limited. In other cases, the stream is read with the given speed.
 - *Force input format*: Pass FFmpeg format (mxg, flv, etc.) if it cannot be detected automatically.
 - *Imotion threshold*: Minimum motion intensity to be detected by the motion detector.
 - *Read frames from source without drops*: Enables posting all appropriate faces without drops. By default, if the video face detector does not have enough resources to process all frames with faces, it drops some of them. If this option is active, it puts odd frames on the waiting list to process them later.
- *Zones*.

Specify the region of tracking within the camera field and region of interest. The region of tracking enables detecting and tracking faces only inside a clipping rectangle. You can use this option to reduce the video face detector load. The region of interest enables posting faces detected only within its boundaries.



- *Faces.*

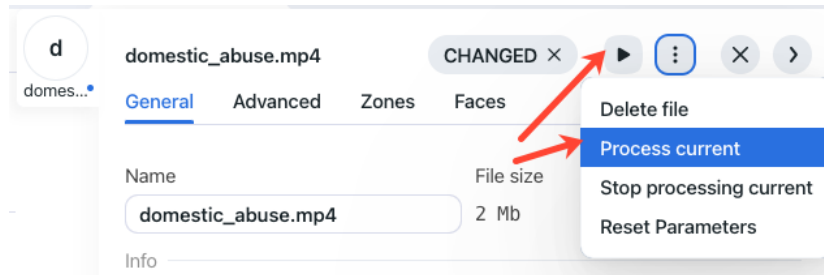
Specify settings for face detection.



- *Size*: Face size range to post, in pixels.
- *Quality*: Minimum quality of a face snapshot to post. Do not change the default value without consulting with our technical experts (support@ntechlab.com).
- *Compression quality*: Full frame compression quality.
- *Full frame in PNG*: Send full frames in PNG and not in JPEG as set by default. Do not enable this parameter without supervision from our team as it can affect the entire system functioning.
- *Track max duration*: The maximum approximate number of frames in a track after which the track is forcefully completed. Enable it to forcefully complete “eternal tracks,” for example, tracks with faces from advertisement media.
- *Miss interval*: The system closes a track if there has been no new face in the track within the specified time (seconds).
- *Send track history*: Send the history of a track.
- *Crop full frame*: Crop posted full frames by ROT.
- *Offline mode*: By default, the system uses the offline mode to process the video, i.e., it posts one snapshot of the best quality per track to save the disk space. Disable it to receive more face snapshots if needed. If the offline mode is on, parameters of the real-time mode are off.
- *Interval (real-time mode)*: Interval in seconds (integer or decimal) within which the face tracker picks up the best snapshot in the real-time mode.
- *Post first object immediately (real-time mode)*: Check to post the first face from a track immediately after it passes through the quality, size, and ROI filters, without waiting for the first Interval to complete. Uncheck the option only to post the first face after the first Interval completes.

- *Post every interval* (real-time mode): Check to post the best snapshot obtained within each Interval in the real-time mode, regardless of its quality. Uncheck the option to post the best snapshot only if its quality has improved compared to the previously posted snapshot.
- *Post first frame of track* (real-time mode): Post the first frame of a track.
- *Post last frame of track* (real-time mode): Post the last frame of a track.
- *Post best normalized image of track*: Send best normalized images for detected faces.
- *Post best full frame of track*: Send best full frames of detected faces.

6. Process the video footage.



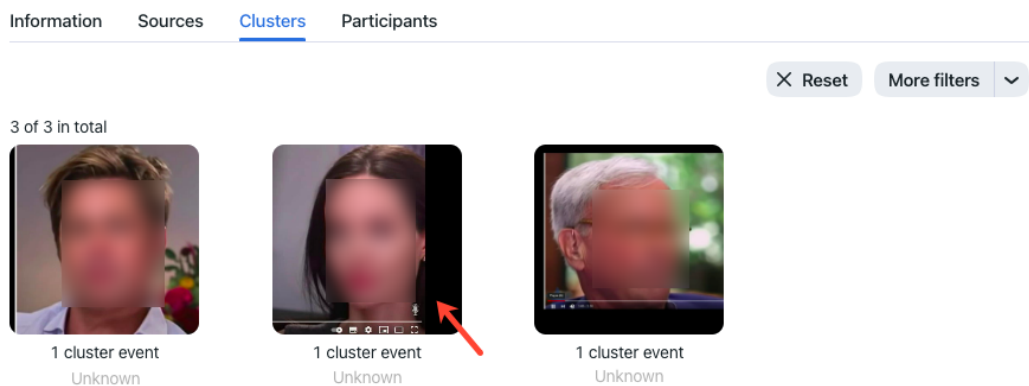
The processing results will be available on the *Clusters* tab.

2.3.6 Detected People Analysis

The clusters of faces, once detected in the video, are shown on the *Clusters* tab. Here you need to parse them subject to the role a person plays in the incident and establish links to relevant global records, other case files, participants of the same case, and other clusters.

Do the following:

1. Click on a face on the list.



Tip: With the large number of clusters, use filters. Click the *More filters* button in the upper-right corner to display them.

The screenshot displays a filter configuration interface for 'Faces'. It includes several sections for filtering results:

- Object:** A blue button labeled 'Faces' and a close icon (X).
- Matches:** Three buttons: 'Any', 'Only with matches', and 'Only without matches'.
- Watch lists:** Two buttons: 'Any' and 'Default Watch List' (selected).
- Camera groups:** A dropdown menu labeled 'Select camera group'.
- Cameras:** A dropdown menu labeled 'Select camera'.
- Card Name:** A dropdown menu labeled 'Card Name'.
- Cluster ID:** A text input field labeled 'Enter cluster ID'.
- Date & time:** Two date-time pickers: '01.01.1970 05:00' and '06.09.2022 15:09', separated by a minus sign and a dropdown arrow.
- Participant:** Three buttons: 'Any', 'Only With Participant', and 'Only Without Participant'.

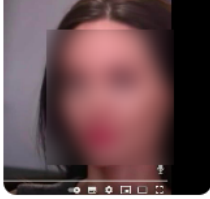
Attributes

- Age:** Two dropdown menus: '1' and '100', separated by a minus sign.
- Gender:** Three buttons: 'Any', 'Male', and 'Female'.
- Beard:** Three buttons: 'Any', 'No beard', and 'Beard'.
- Glasses:** Four buttons: 'Any', 'No glasses', 'Eyeglasses', and 'Sunglasses'.
- Emotions:** Eight buttons: 'Any', 'Anger', 'Disgust', 'Fear', 'Happiness', 'Sadness', 'Surprise', and 'Neutral expression'.
- Face mask:** Four buttons: 'Any', 'No mask', 'Improperly worn', and 'On'.
- Liveness:** Three buttons: 'Any', 'Real', and 'Fake'.

At the bottom right, there are two buttons: 'Reset filters' and 'Apply filters'.

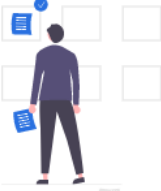
2. Link the face to a matching record in the *global record index* and matching participants of other case files. Do so by clicking the *Add* button. You can also link the face to participants of the same case and other clusters.
3. Click *Mark as participant*.

Detected people



1 cluster event
Unknown


This is the individual who was automatically recognized while processing case file footage. If they are relevant to the case, mark them as a participant. If you are sure that there is a match with the other recognized individuals, or with the index records, or with the participants of other cases, establish a link by clicking the Add button



Mark as participant

Detected people


0.519



1 cluster event
Unknown


Record index

0.822



Mrs Smith
● Hitmen

0.503



Mr Smith
● Hitmen

Add

- Specify the participant name and type, and add a comment if necessary. This will create a new case participant record that you will be able to view on the *Participants* tab.

Create New Participant ✕

Name Type ● Suspect ▾

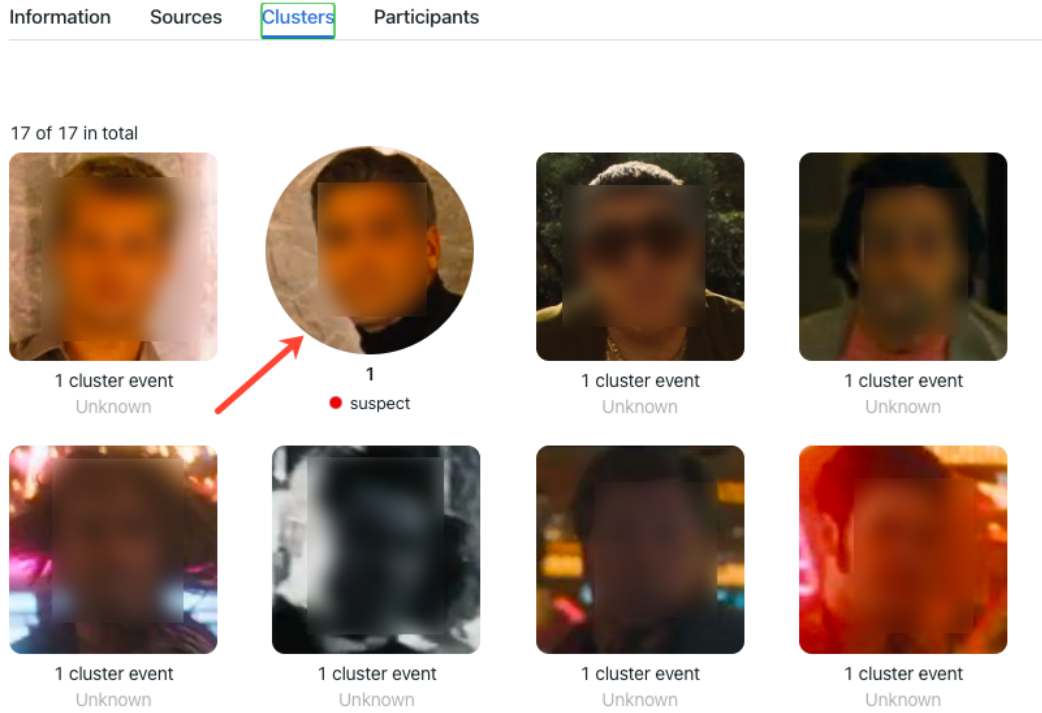
Comment

Save & Close

A participant can be one of the following types:

- suspect
- victim
- witness

- Click *SaveClose*.
- You can reopen the connections wizard on the *Clusters* tab by clicking on the participant's face again.

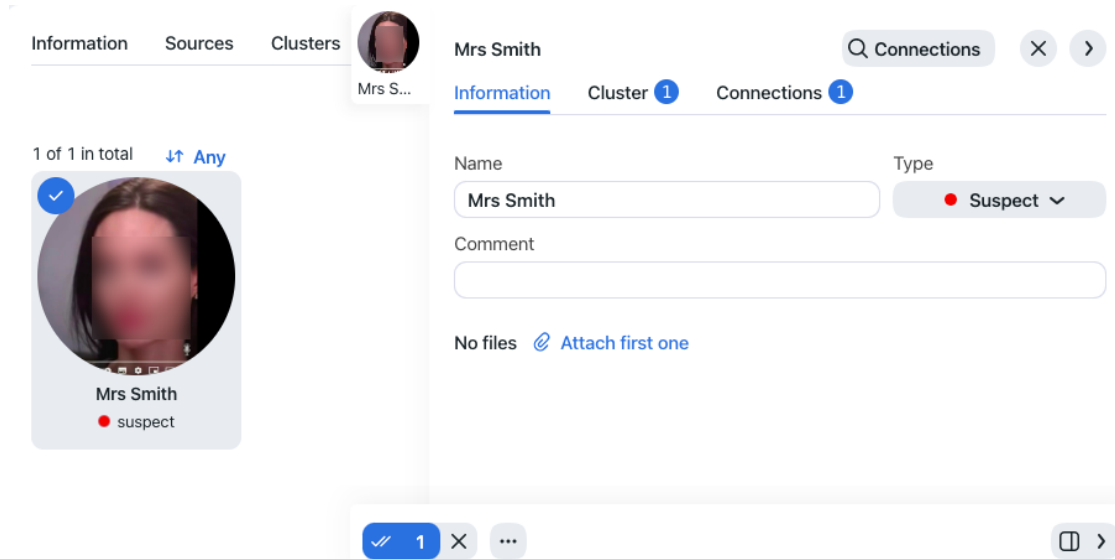


To access the participant record, navigate to the *Participants* tab.

2.3.7 Case Participant Records

The *Participants* tab provides access to all participant data collected so far.

To view a case participant record, click the relevant face on the list. In the record, you can attach files, view detected face images of the participant, re-establish links with matching global records and other case files, etc.

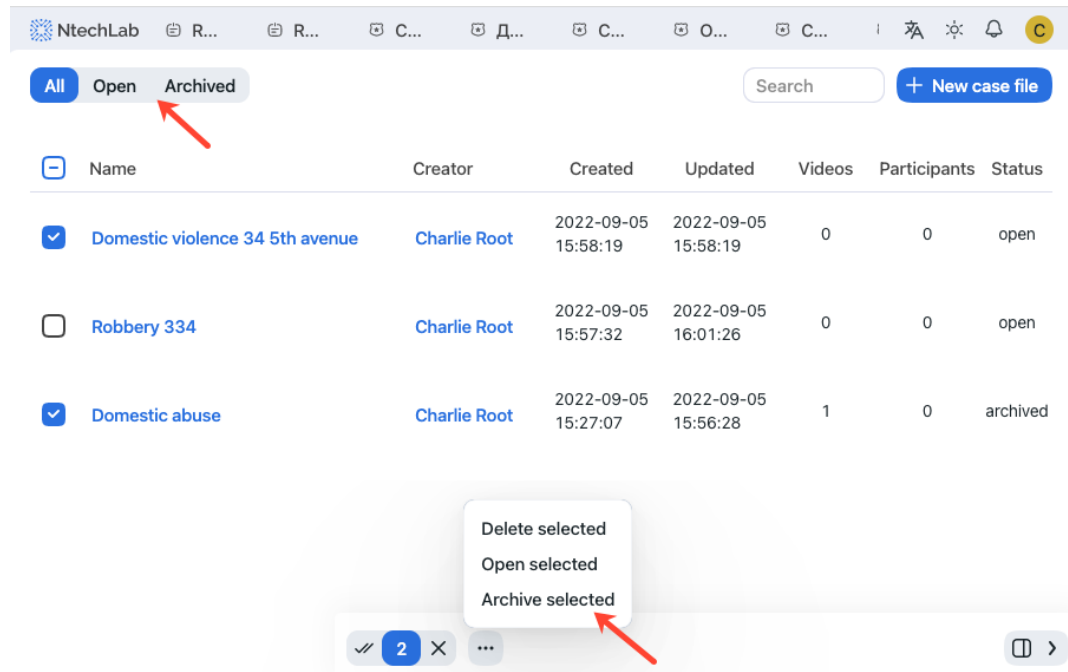


2.3.8 Case File Archive

It's possible to label a case file as archived to indicate that the case is closed, or for another reason.

To archive/unarchive case files, do the following:

1. On the *Case Files* tab, select one or several case files.
2. Click *Archive selected* to archive the case files. Click *Unarchive selected* to reopen them.



You can filter the case file list by archived status: *All*, *Opened*, *Archived*.

2.4 Search Faces in System

FindFace allows you to search for individuals throughout the entire system.

To find an individual, do the following:

1. Navigate to the *Search* tab.
2. Specify a face to search for in one of the following ways:
 - by record URL
 - by uploading a photo

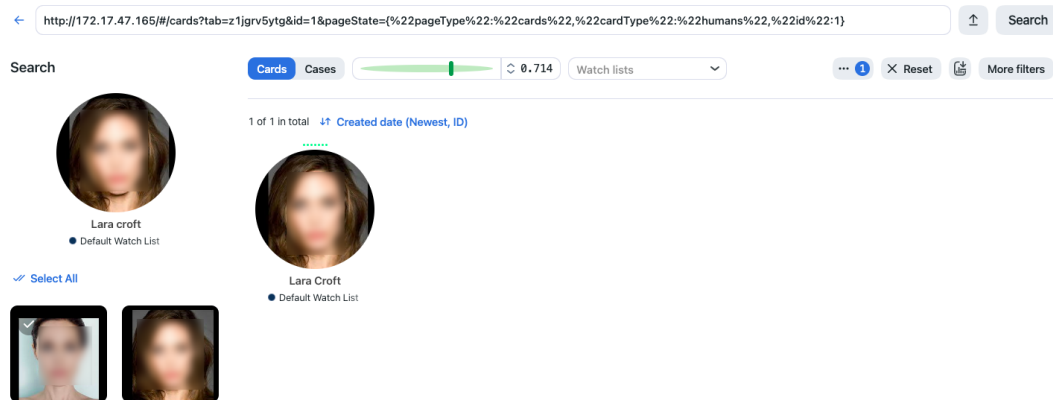
Enter event or record URL, or upload a file

3. Click *Upload*.

If you specified a record URL, select a photo from it. If there are multiple photos, you can select some or all of them. Click the *Apply* button.

If you uploaded a photo, it will be displayed in the new window. If there are multiple faces in the image, select the one of your interest. Click *Search*.

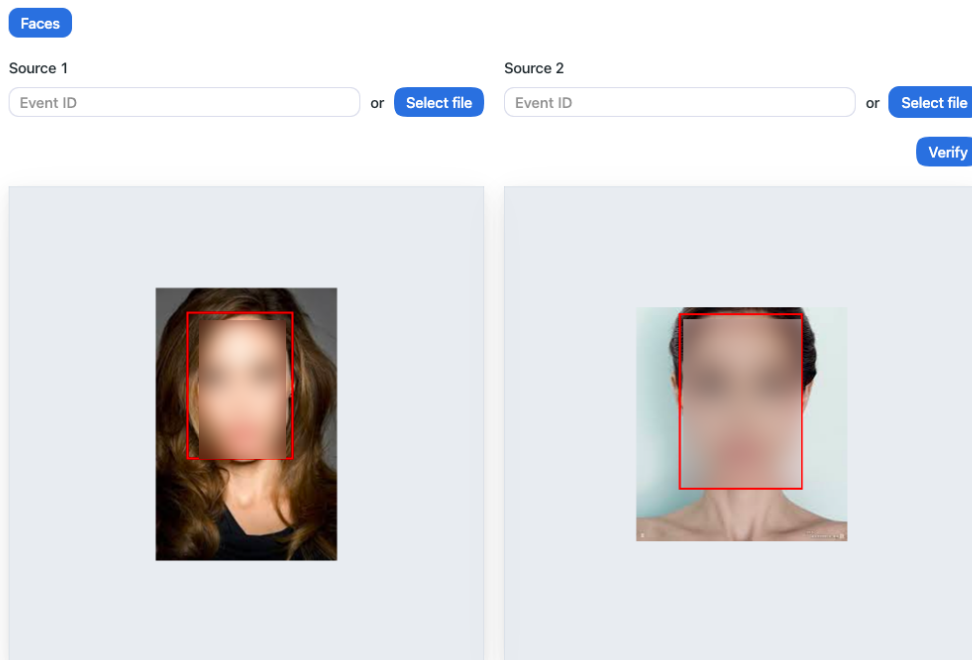
4. You will see the search results appear. If necessary, you can narrow down your search by specifying a watch list, similarity threshold, etc.



2.5 Compare Two Faces

FindFace allows you to compare two faces and verify that they belong to the same individual. Do the following:

1. Navigate to the *Verify* tab.
2. Upload two photos with faces to compare.



3. Click *Verify*. You will see the probability that the faces match.

2.6 Reports

FindFace provides a possibility of building reports on the following system entities:

- *search results*
- *global records*

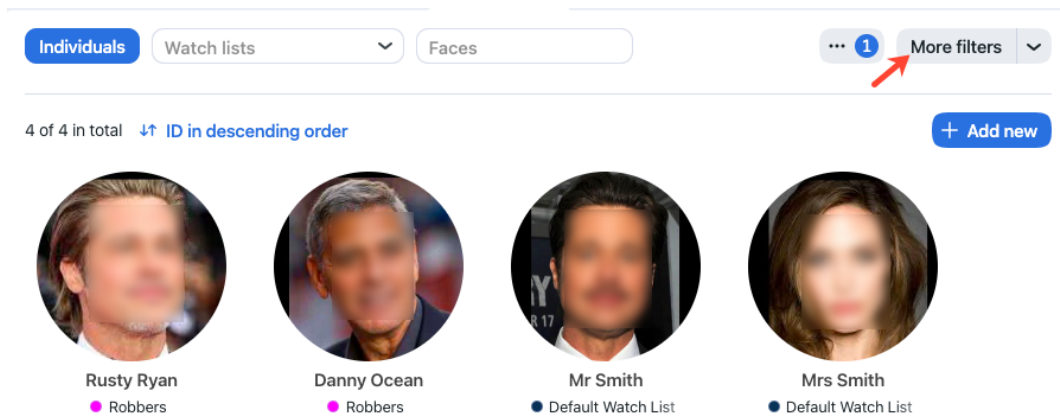
In this section:

- *Build Report*
- *Work with Reports*

2.6.1 Build Report

To build a report on a system entity, do the following:

1. Navigate to the tab associated with the required entity: *Search, Record Index*.
2. Perform the search if you are on the *Search* tab.
3. Click the *More filters* button.



4. Set filters for the report.
5. Click *Create report*.

6. Specify the report name.
7. Check one or several report formats: XLS, JSON, CSV.
8. Choose whether to save the report images as links, thumbnails, or full frames.

9. Click *Create*. The report will be available for download on the *Reports* tab.

2.6.2 Work with Reports

You can access reports previously created in the system on the *Reports* tab. The following operations are available:

- Download selected reported.
- Update selected reports.
- Delete selected reports.

<input checked="" type="checkbox"/>	ID	Name	Type	Modified	Records	Size	Status
<input checked="" type="checkbox"/>	1	Individuals records report	Individuals cards	06.09.2022 17:33:23	9	JSON: 6.38KB XLS: 7.93KB	Completed ↓

✓ 1 ×
Download
Update
Delete
🗑️ ⏪

See also:

Configure Saving Images in Reports

2.7 Audit Log

The FindFace comprehensive and searchable audit log is an excellent complementary tool for user management that provides you with a thorough audit of the user actions and strengthens your system protection. You can access this functionality on the *Audit Logs* tab.

User	IP	Device ID	Action	Object	Object ID	Time
admin	172.20.78.6	702fb88f-c8ec-4469-90fa-bdb8b25bd4e9	Create	Role	4	27.08.2022 14:09:30
admin	172.20.78.6	702fb88f-c8ec-4469-90fa-bdb8b25bd4e9	Delete	User	2	27.08.2022 13:34:02
admin	172.20.78.6	702fb88f-c8ec-4469-90fa-bdb8b25bd4e9	Create	User	2	27.08.2022 13:32:55
admin	172.20.78.6	702fb88f-c8ec-4469-90fa-bdb8b25bd4e9	Upload	License		27.08.2022 13:05:09
admin	172.20.78.6	702fb88f-c8ec-4469-90fa-bdb8b25bd4e9	Update	Settings		27.08.2022 12:59:25
admin	172.20.78.6	702fb88f-c8ec-4469-90fa-bdb8b25bd4e9	Authorization	User	1	27.08.2022 12:59:18

Each record provides the following data:

- username of the user who performed the action
- IP address where the request came from
- device id: the unique identifier of the client device
- action type such as authorization, search, object modification, restart, and so on
- object type to which the action applies, for example, a record or a case
- object identifier
- details, subject to the action type
- timestamp

Use the filter panel above to set up the search conditions.

2.8 Remote Alerting and Remote Search

This section covers an additional but very useful functionality which is a possibility of pulling face recognition events, matching with records on the local server, from remote facial recognition systems. This functionality has a large scope of possible applications. One course is tracking offenders' location and routes and detecting alleged accomplices. Another one is finding missing people. The results are especially great if applied to Public and Transport Safety systems with thousands of cameras.

The remote alerting is disabled by default, so if you haven't configured it yet, click [here](#) for instructions.

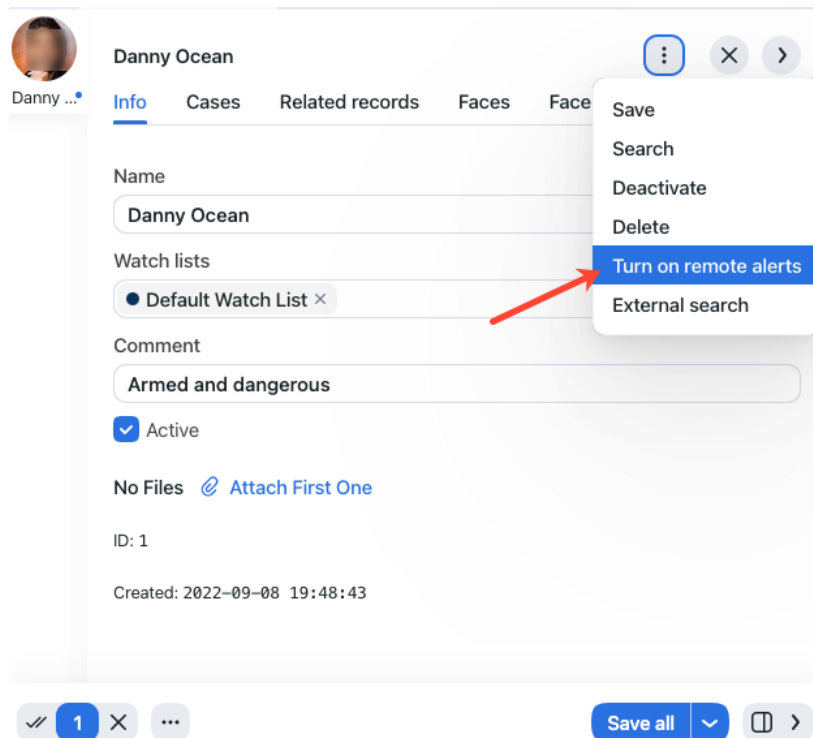
In this section:

- *Turn On/Off Remote Alerts for Individuals*
- *View Remote Alerts*
- *Search Individuals in Remote Systems*

2.8.1 Turn On/Off Remote Alerts for Individuals


To turn on/off remote alerts for a specific individual, do the following:


1. Navigate to *Record Index*.
2. Open the individual's record.
3. Click *Turn on remote alerts* to enable remote alerting.



Click *Turn off remote alerts* to disable remote alerting.

4. In the case you are turning on remote alerts, specify the reason for that. Click *Turn on remote alerts*.





Danny Ocean
● Default Watch List

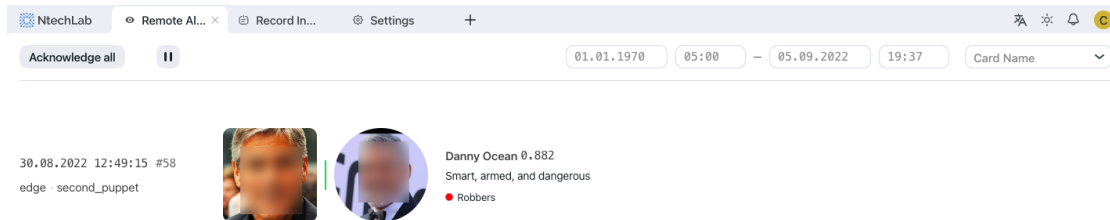
Reason for remote alerting

Specify the reason for remote alerting

Turn on remote alerts

2.8.2 View Remote Alerts

You can view alerts from remote facial recognition systems on the *Remote Alerting* tab. You can filter the alerts by date and time, and name of an individual.



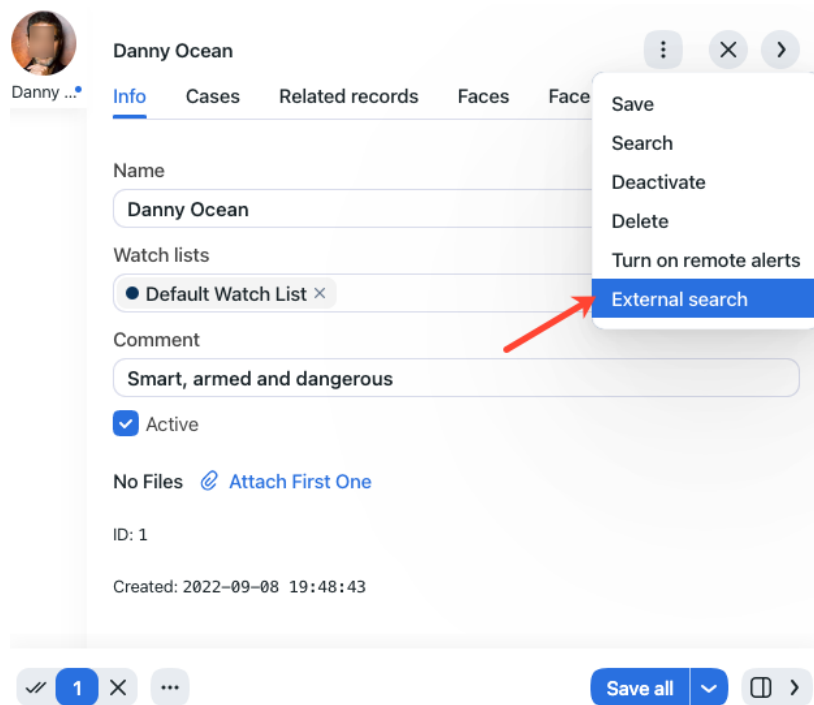
The screenshot shows a web interface with a browser tab titled "NtechLab" and several sub-tabs: "Remote Al...", "Record In...", and "Settings". Below the tabs, there is a filter bar with "Acknowledge all" and a pause icon, followed by date and time filters: "01.01.1970" to "05:00" and "05.09.2022" to "19:37". A dropdown menu is labeled "Card Name".

The main content area displays an alert for "Danny Ocean 0.882". To the left of the alert, there is a timestamp: "30.08.2022 12:49:15 #58" and the text "edge - second_puppet". To the right of the timestamp are two circular profile pictures. To the right of the profile pictures, the name "Danny Ocean 0.882" is displayed, followed by the description "Smart, armed, and dangerous" and a red dot next to the label "Robbers".

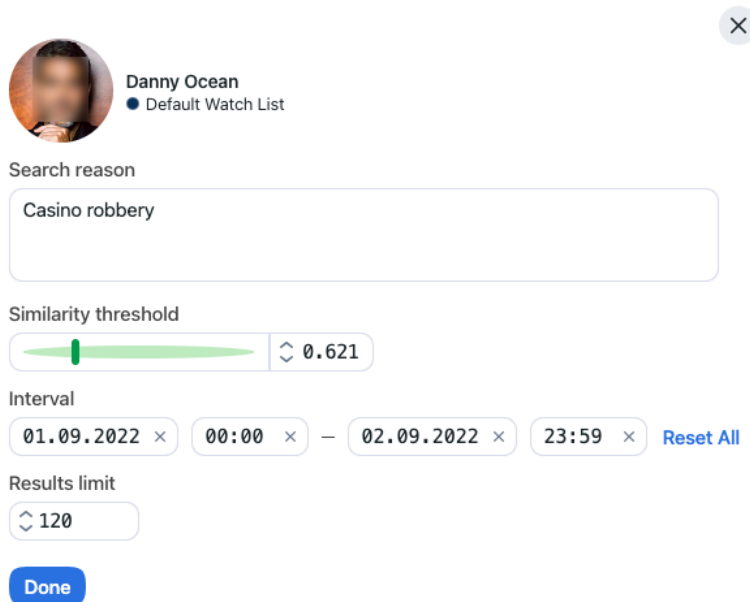
2.8.3 Search Individuals in Remote Systems

To search an individual in remote systems, do the following:

1. Navigate to *Record Index*.
2. Open the individual's record.
3. Click *External search*.



4. Specify the search conditions, such as the similarity threshold, date and time of the individual's appearance, and the maximum number of search results. Click *Done*.



The search results will be shown on the *Remote Search* tab. You can filter them by date and time, name of the individual, and user who initiated the search.

INTEGRATIONS

This chapter is all about integration with FindFace. In the current version of FindFace, you can only integrate your system via HTTP API.

3.1 HTTP API

Detailed interactive documentation on the FindFace HTTP API is available after installation at <http://<findface-ip:port>/api-docs>. Learn and try it out.

Tip: You can also find it by navigating to *Settings* -> *API Documentation* in the web interface.
