NxWitness + VisionLabs' FaceStream and LUNA Integration - generic events based

All the required files can be downloaded from the link below:

https://drive.google.com/drive/folders/1rpvdyJfDUIngz3WK0GgjCGGxnteh4PDV?usp=sharing

In order to obtain a trial License required to activate VisionLabs' Software, please contact Anton Nazarkin, VisionLabs sales rep.

Part One: Install

A combination of a face recognition software (**FaceStream**) and a powerful media server (**NxWitness**) brings you an ability not only to improve and to advance your existing security system,

but also allows you to implement such mechanisms as keyless door control and many more with minimum system/hardware requirements.

User friendly **NxWitness UI** allows you to control your security/surveillance objects with high efficiency without high level background knowledge required.

!!! This manual covers current way(using Linux CentOS 7.3 VM with VisionLabs **Luna** pre-installed and configured) to integrate **FaceStream** with **NxWitness** !!!

Possible system setups:

Luna Vm + FaceStream & NxWitness running on the same device	**Luna VM + FaceStream & NxWitness using different hardware**
Pros: easier setup, easier to troubleshoot, less hardware via network interaction.	**Pros**: can be implemented on a pretty basic hardware, independent servers make system more stable, more room for future upgrades/modifications.
Cons: Higher hardware requirements. Higher demand on power management and administration.	**Cons**: More operations have to be done in order to follow relevant network setup.

Minimum Requirements for FaceStream:

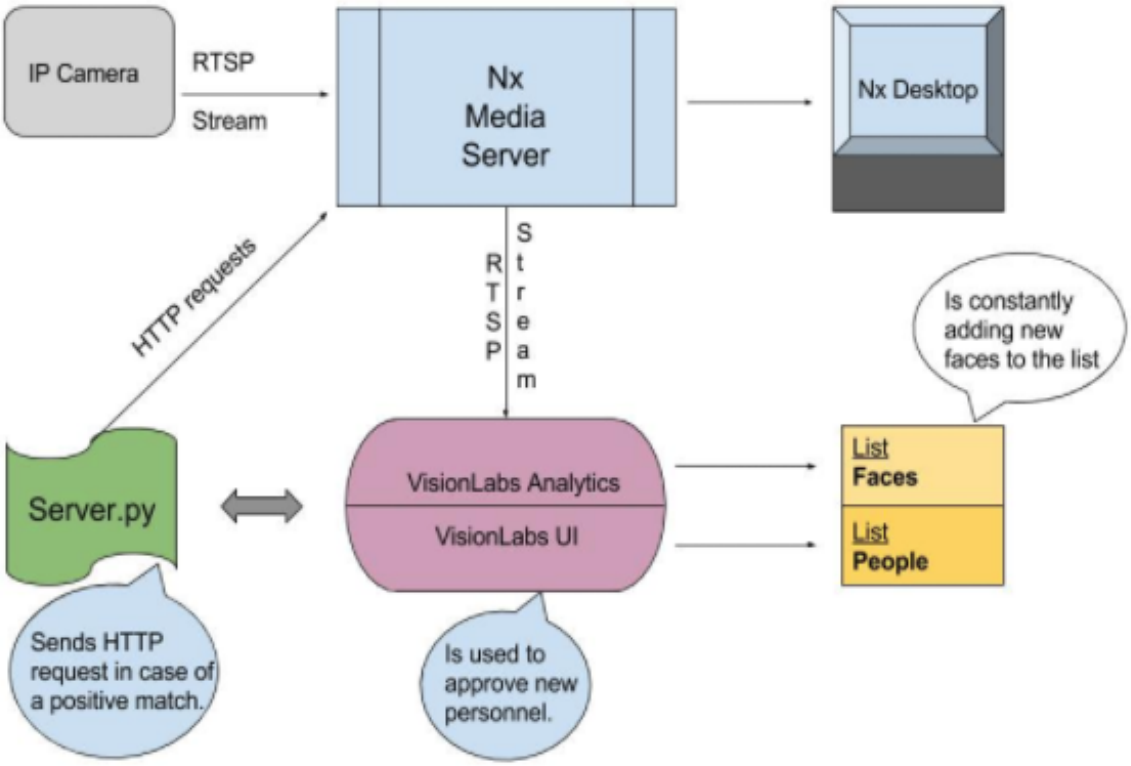
- CPU 2GHz.
- 4Gb RAM.
- HDD free space 400Mb.

- Windows (64-bit) 7 and up or CentOS 7.3 x86_64.

From our experience and in order to decrease the processing time, we strongly recommend to run the VM image (**Luna**) with the following settings:

- 8Gb RAM and up.
- CPU cores 6 and up.
- SSD 10Gb or more.

General diagram:



Comments:

VisionLabs Analytics = FaceStream

VisionLabs UI = LUNA

Requisites to have before we proceed:

For the device with **NxWitness**:

- Camera ID.
- Full access to your current **Nx System**.

For the device with VisionLabs Virtual Machine (**LUNA**):

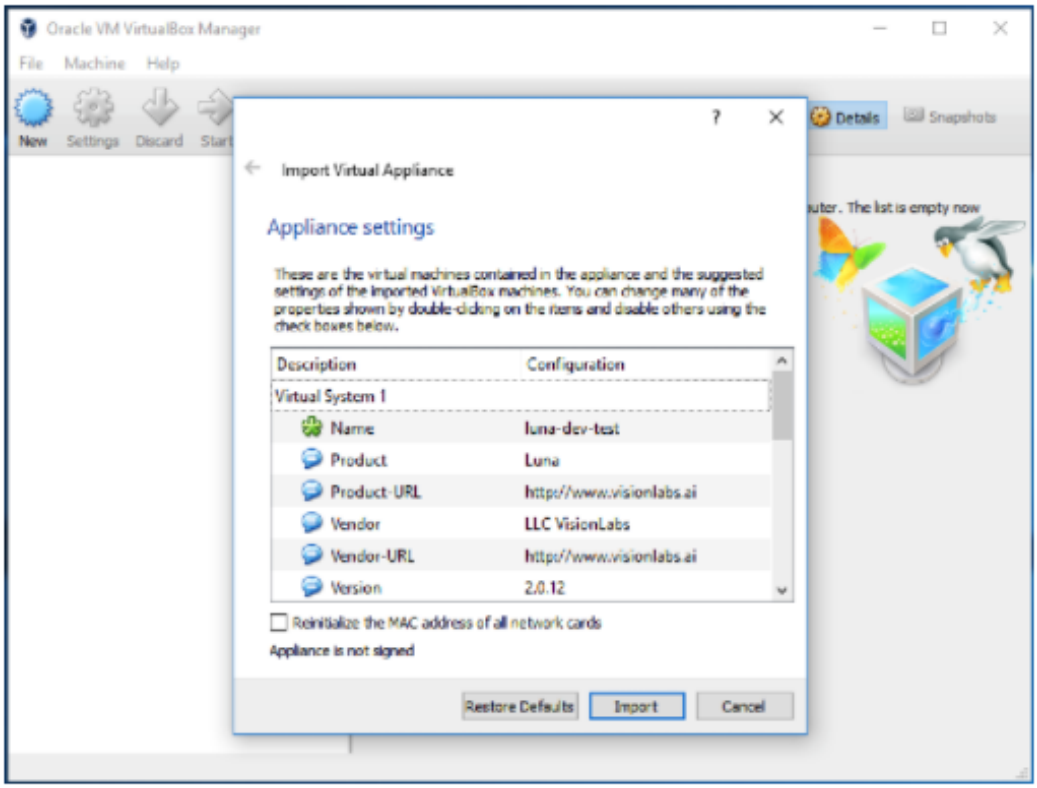
- Hardware virtualization support is enabled in BIOS.
- VisionLabs activation keys (.v2c file).
- VirtualBox or VMWare installed.
- Microsoft C++ Redistributable 2015.
- Sentinel HASP (software protection system, used by VisionLabs LLC. You need to install license file and driver for HASP installation)*****.
- Python 3.4 and up.

---

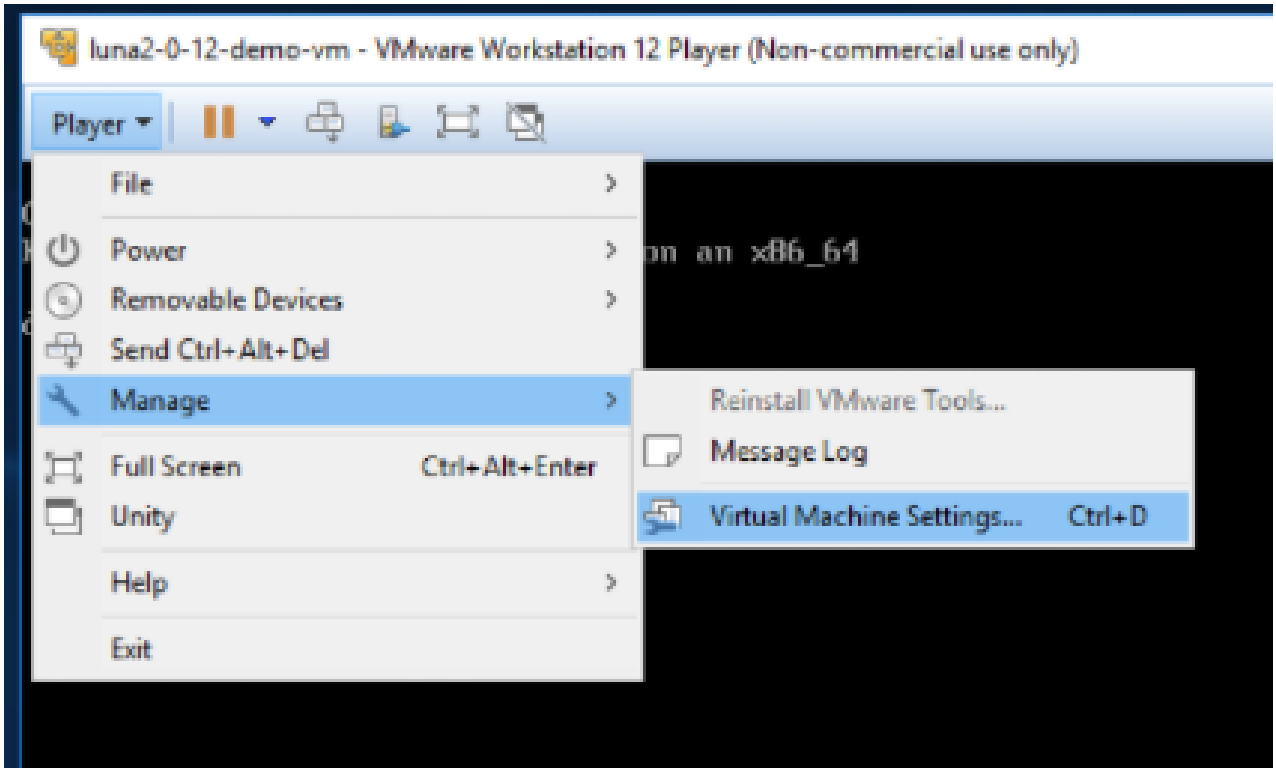
* - will be provided together with VisionLabs VM image(LUNA).

Step 1. Import of the Virtual Machine (LUNA)

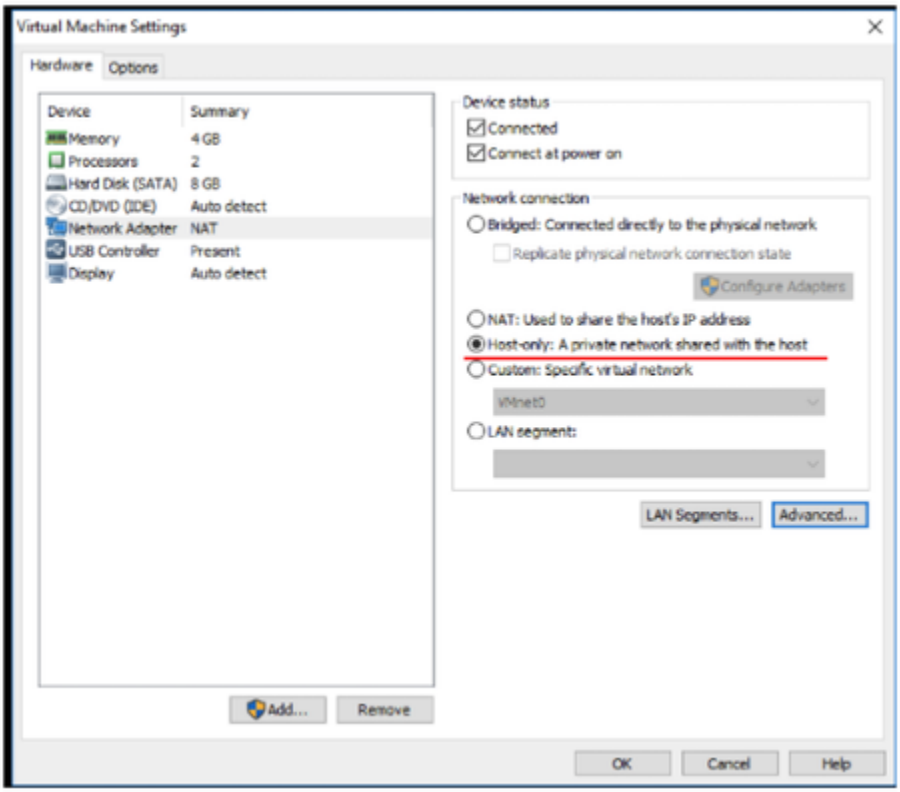
1. Launch the lunaX-X-XX-demo-vm.ova from the delivery package.
2. Import of the virtual machine will start. Press "Import" to continue.



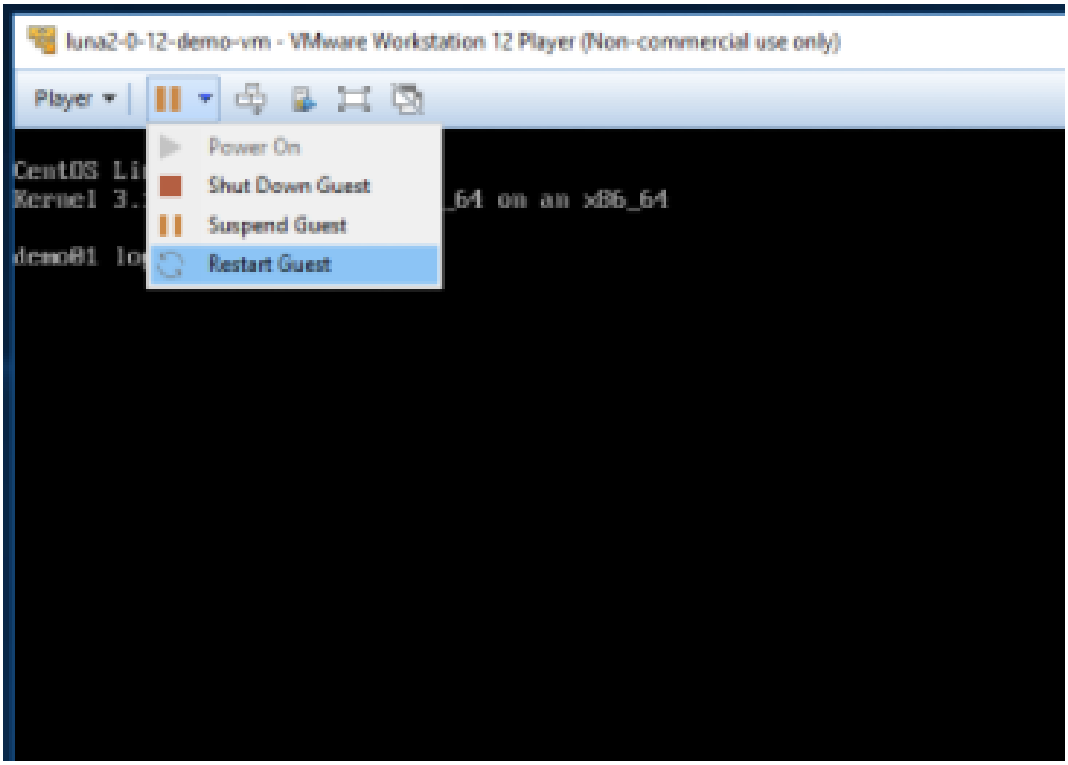
3. Go to **Player** -> **Manage** -> **Virtual Machine Settings**.



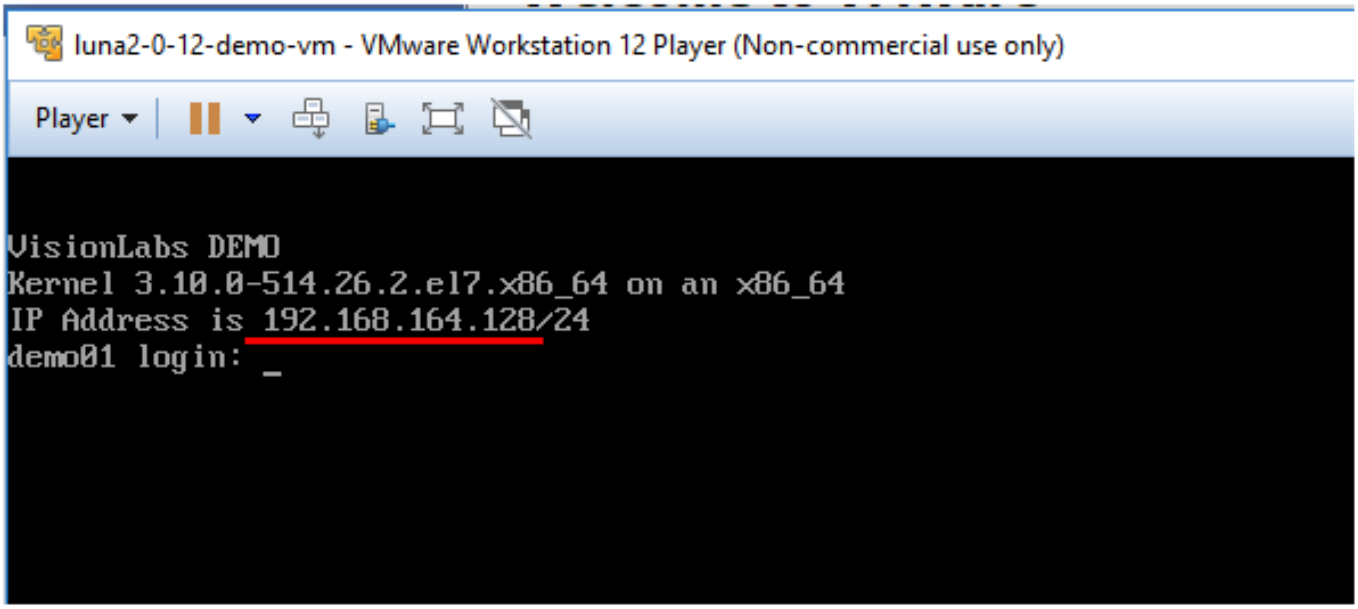
4. In the Hardware tab, go to the **Network Adapter** menu item and set the **Host-only** value. Click **OK** and close the **Settings** window.



5. Reboot the virtual machine and wait for the process to finish.

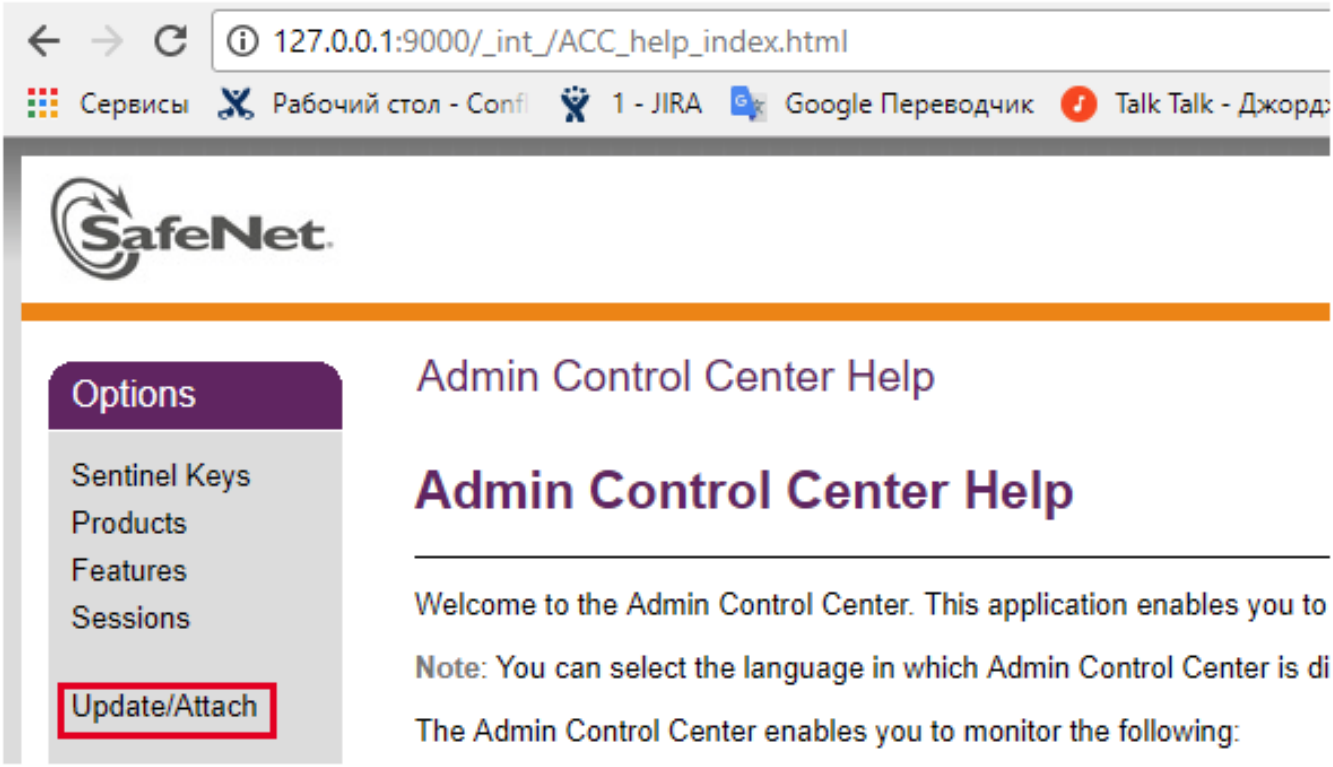


6. The local address value is displayed in the Virtual machine screen. In the example below, the address is *192.168.164.28*, although it may differ according to the installation version.

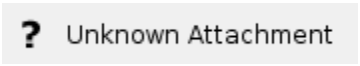


Step 2. License Number 1 Installation

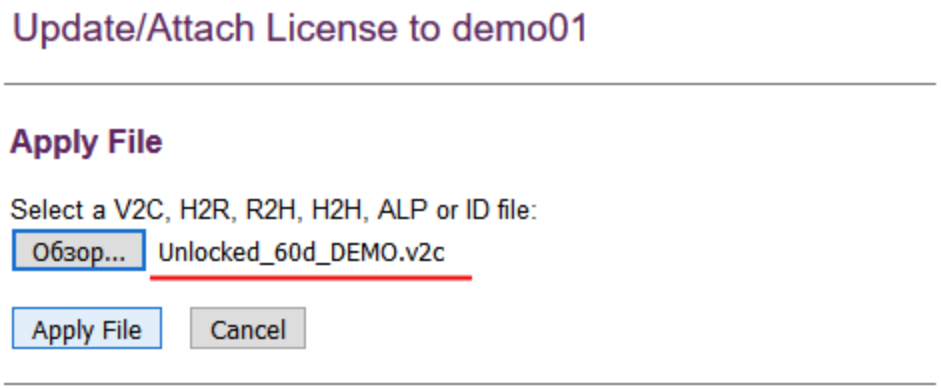
1. Go to *https:// <virtual machine address>: 9000/hasp*. Go to **Update/Attach**.



2. Click **Browse** and select the *. v2c license file from the delivery package. Make sure the file is joined to the session.

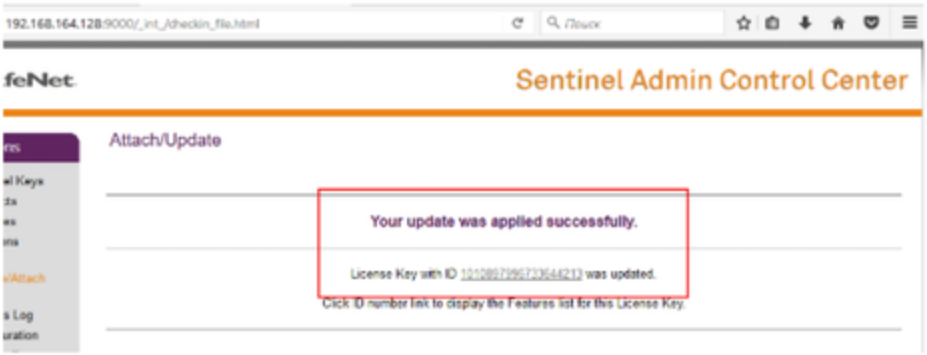


3. Click **Apply** File.



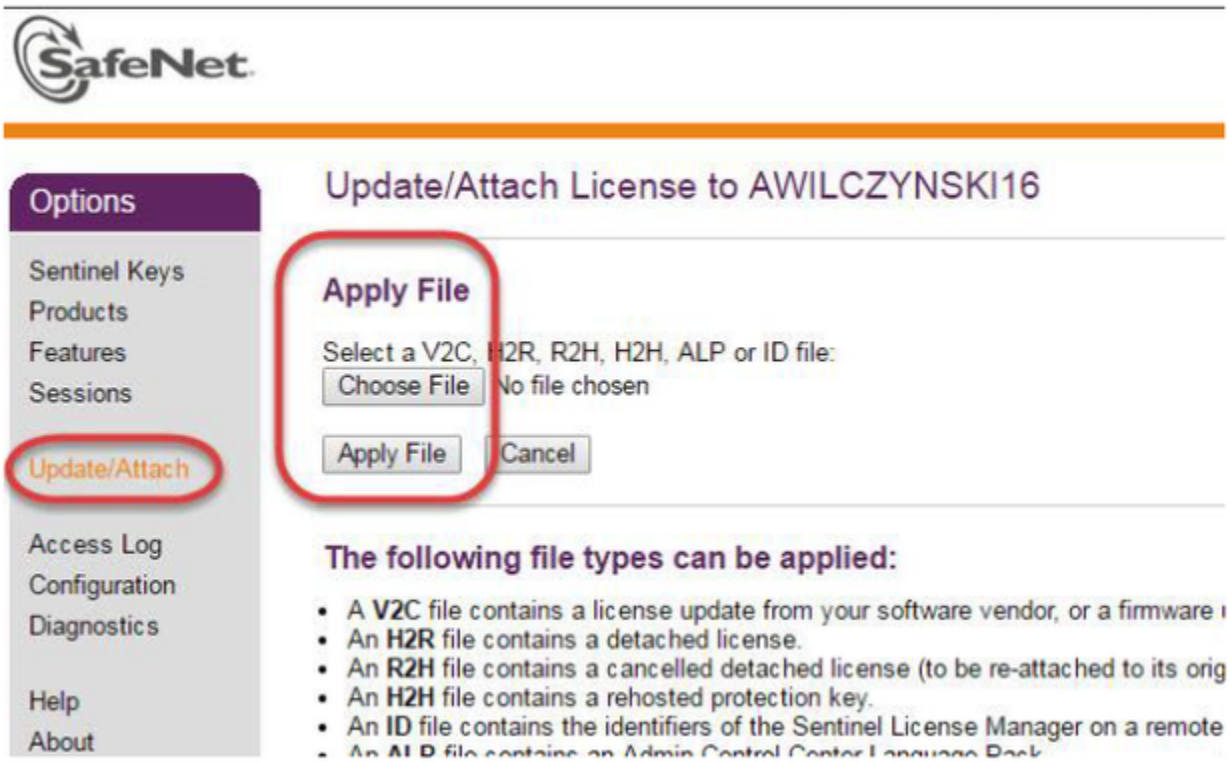
4. If the applying process was successful, the following message appears:

«Your update was applied successfully».



Step 3. License Number 2 Installation.

1. Open **cmd** and for the **Sentinel HASP (local_drive\facestream2_win_v.1.5.2\extras)** driver installation launch *haspdinst.exe* utility with "*-i*" (install) parameter from the command line.
    a. **IMPORTANT!** If you have issues installing HASP driver on Windows 10 (eg. installation never finishes), try to look for solution here: http://sentineldiscussion.gemalto.com/topic/slow-installation-using-command-line-installer-haspdinst-exe. Basically, you need the package not older, than this one: https://supportportal.gemalto.com/csm/?id=kb_article&sys_id=32fbed35db373684d298728dae96199e. After that, do the Step 3.1.
2. After installation is completed, go to the web-page *http://localhost:1947*. Choose the license file *.V2C in the menu section «Update/Attach» by clicking on «Choose file», then activate it by clicking on «Apply file».



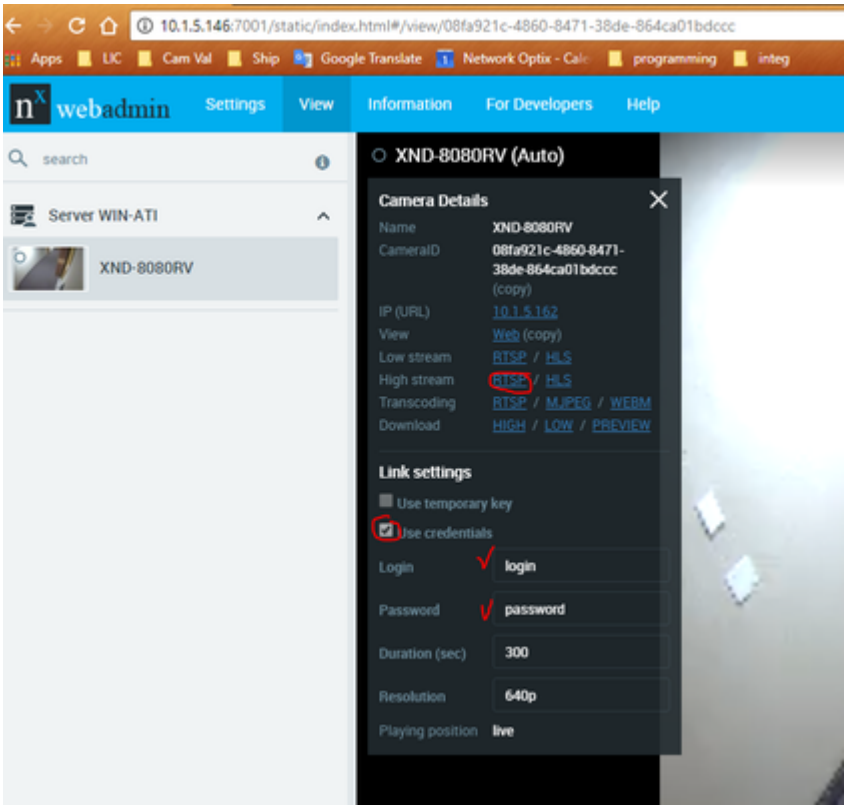
Step 4. Launch FaceStream

1. Get RTSP link for the desired camera from **NxWitness**:

- Open web interface of the NxWitness media server - *http://127.0.0.1:7001/static/index.html#/view?cameraLinks*

!!!If you are using a different device to run **NxWitness**, replace *127.0.0.1* with an actual local IP address!!!

- Enable 'Use credentials' checkbox in Camera Details panel.
- Enter credentials for **NxWitness**.
- Copy link address for High Stream RTSP.



2. Open **cmd** and **CD** to a folder with the FaceStream program/bin.

3. Run the FaceStream with the RTSP link as an argument. Eg.:

FaceStream2.exe -Src rtsp://admin:admin123@10.1.5.197:7001/e3e9a385-7fe0-3ba5-5482-a86cde7faf48?stream=0 -Dst http://192.168.56.101:9000/facestream/receiver

!!! *10.1.5.197* - local IP address of a device with **NxWitness** !!!

!!! *192.168.56.101* - local IP address of the **VisionLabs** VM !!!

Step 4. Configuring and launching Python scripts

1. Open folder itcanfly-luna-stats-c0b659cd813d.
2. Configure Python Integration Script:

Open **settings.py** and update the parameters below -

- NOTIFICATION_URL - use **Nx Server** credentials + IP address

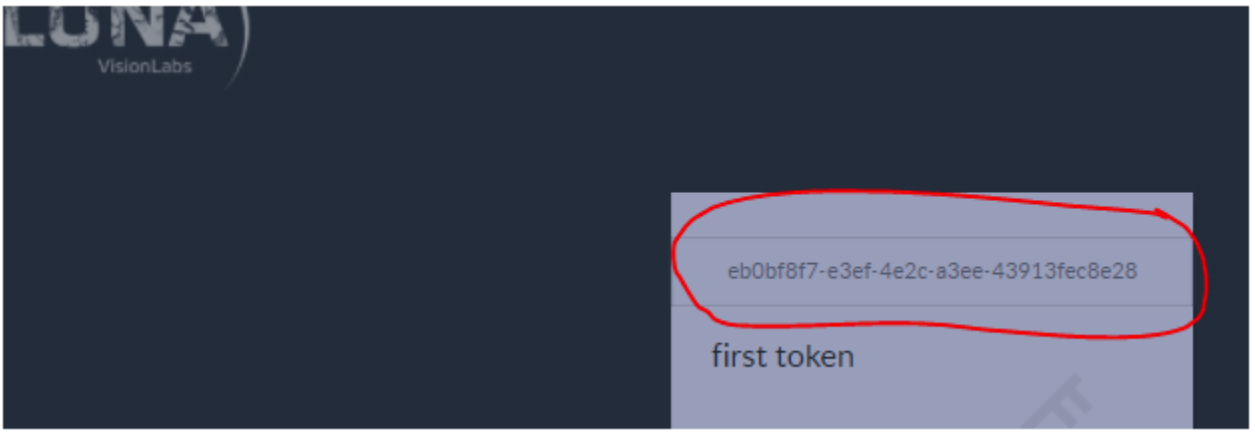
Eg.: *http://admin:NxWitness@10.1.5.146:7001/api/createEvent*

- SUBSCRIPTION_URL - use **Luna** Web Socket address

Eg.: *ws://192.168.56.4:9000/events/api/subscribe*

- CAMERA_ID *https://support.networkoptix.com/hc/en-us/articles/217051248-API-Tip-Finding-a-cameraId*
- LUNA_LIST - can be found at *http://192.168.56.4:9000/lists* where 192.168.56.4 is your VM's IP address.

- LUNA_TOKEN - can be found at *http://192.168.56.4:9000/tokens* where 192.168.56.4 is your VM's IP address.



3. Install all the dependencies by running a command below from **cmd**:

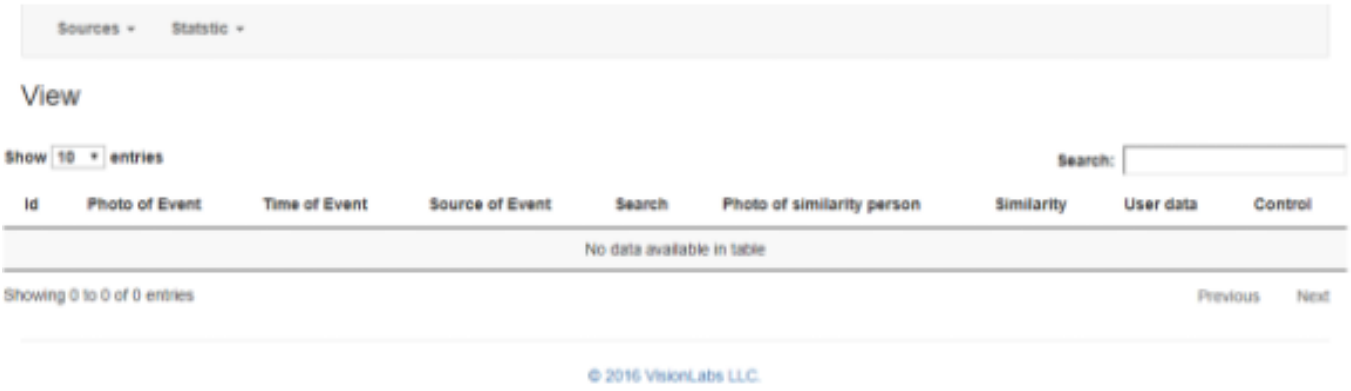
   pip3 install -r requirements.txt

4.Run:

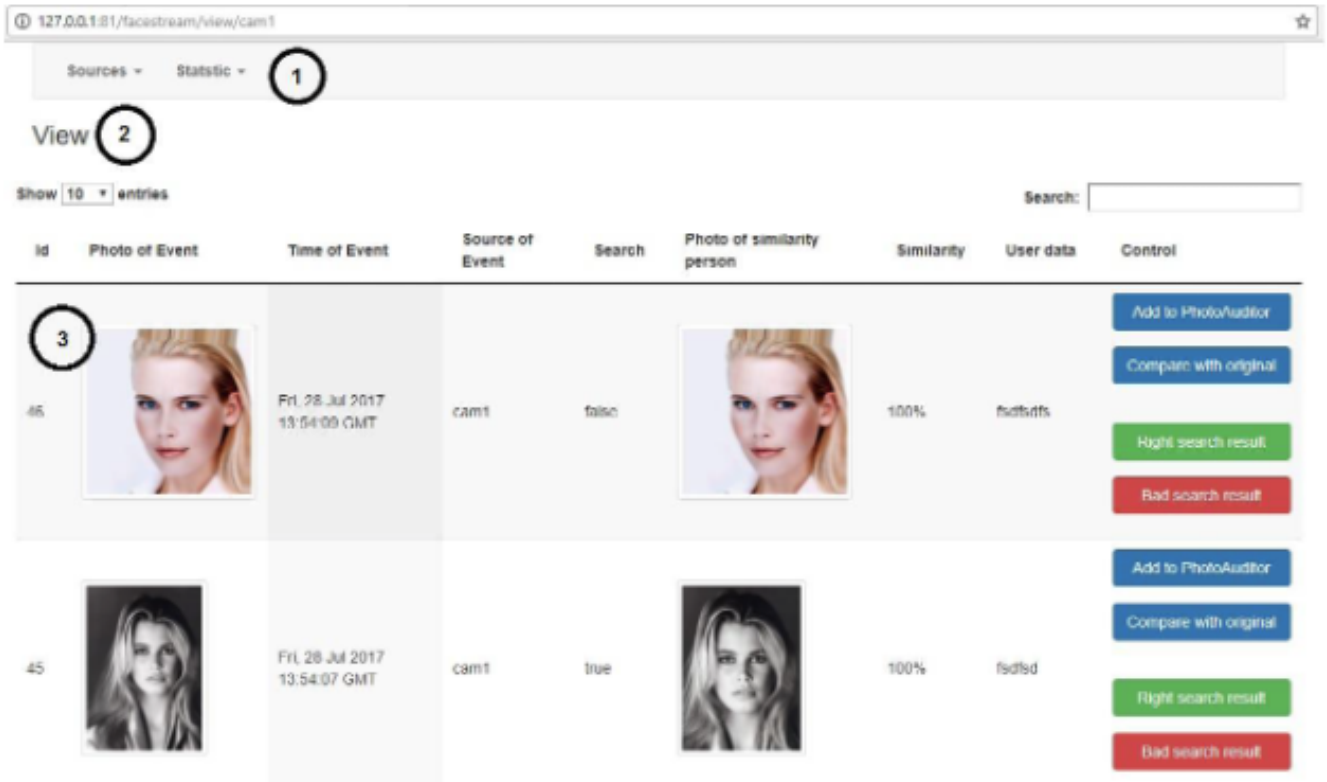
   python3 server.py

Step 5. Check if the System works

Proceed to *http://192.168.56.4:9000/facestream/view* - constant address of the FaceStream Manager, where *192.168.56.4* is your VM's IP address:

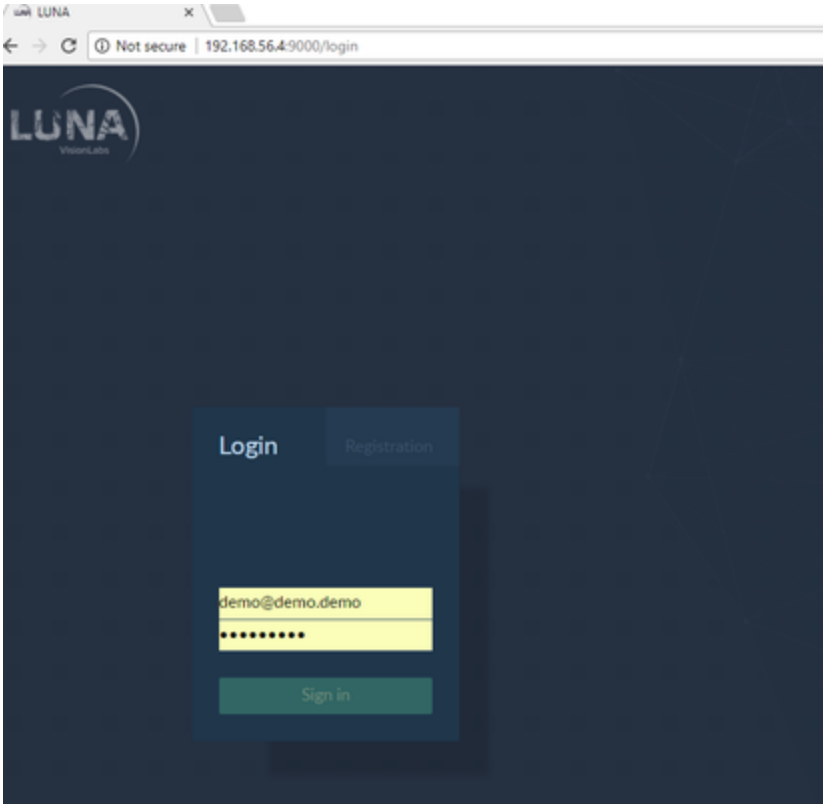


Once objects start to appear in front of the camera, the page will be updated in the following manner:



Part Two: UI

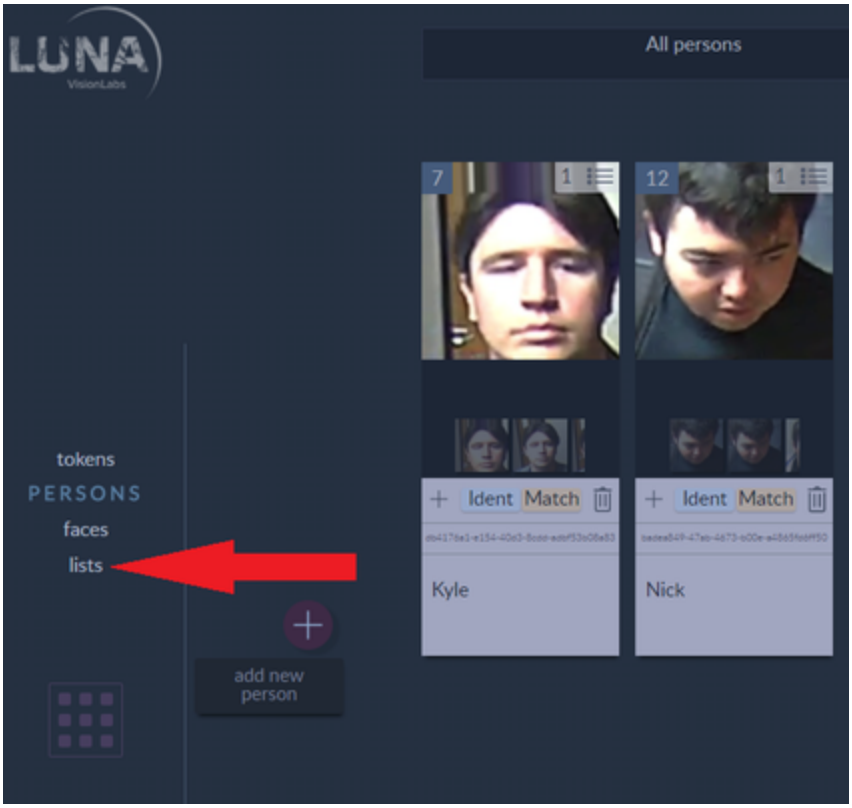
LUNA's Web Interface can be found at <*your virtual machine IP address:9000*>. Once you navigate to it, you'll be promted with a login page as shown below:



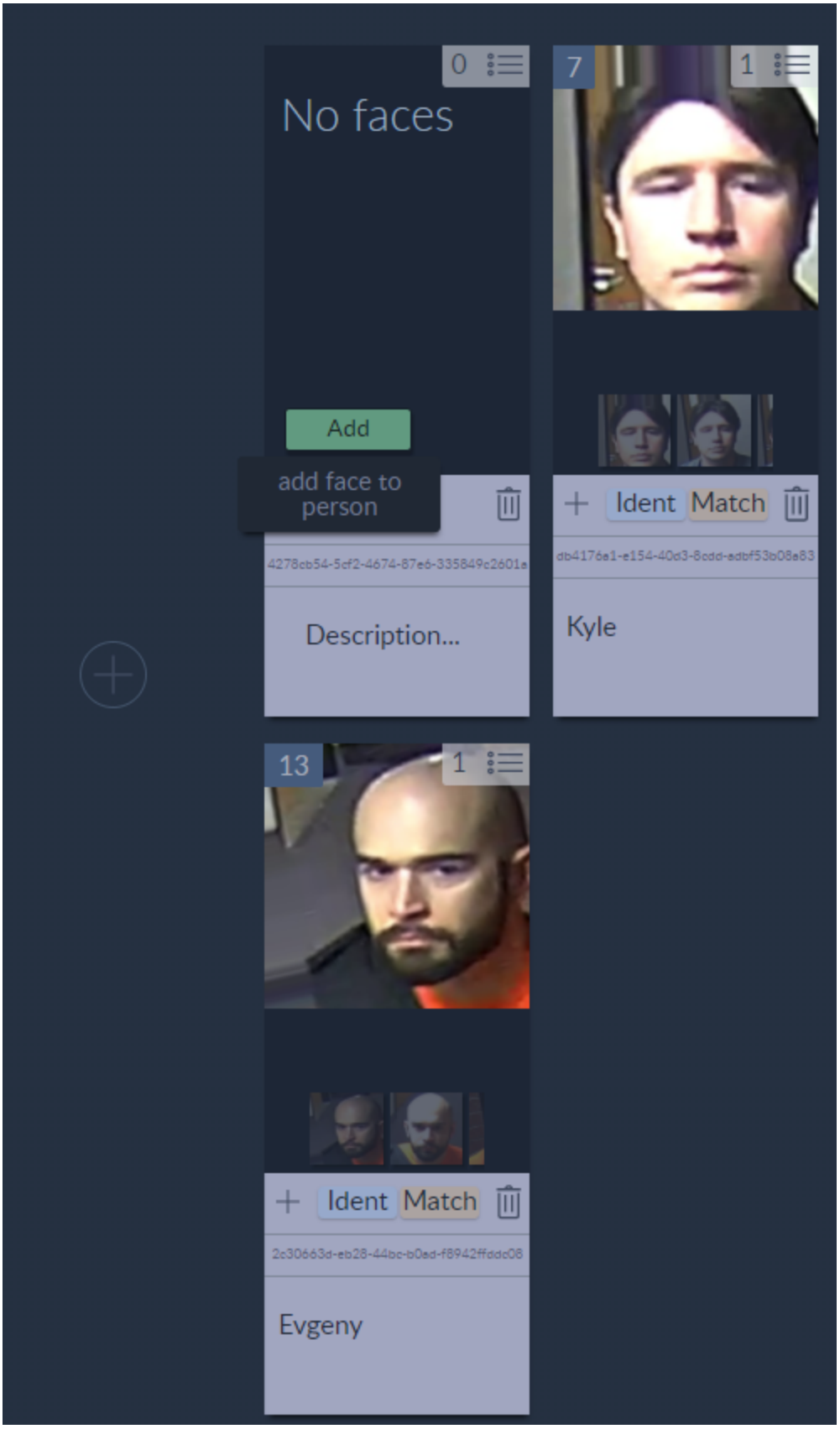
!!! For the login information, please contact Anton Nazarkin, VisionLabs sales rep, on behalf of the NetworkOptix, Inc. !!!

Once logged in, click on **Persons** button to navigate to the **Persons List** web represantation page:

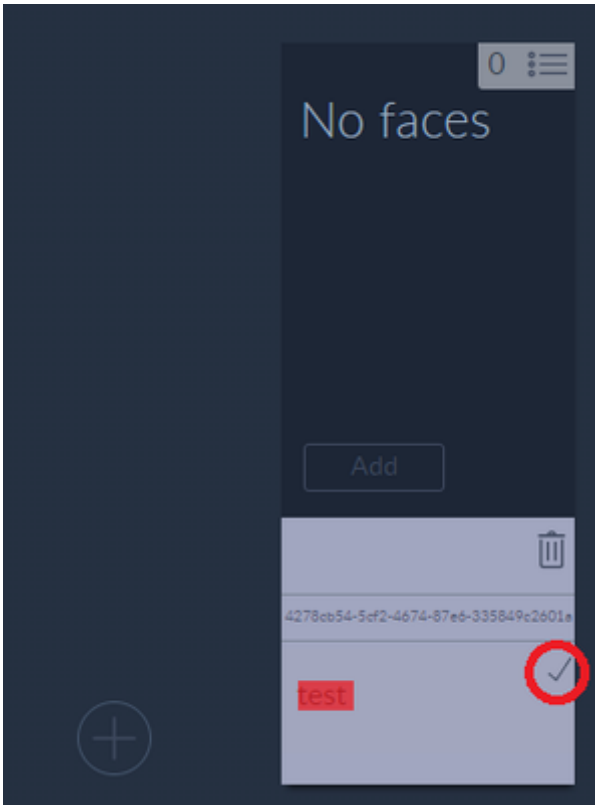
In order to add a new person to a list, click on a plus sign as shown below:



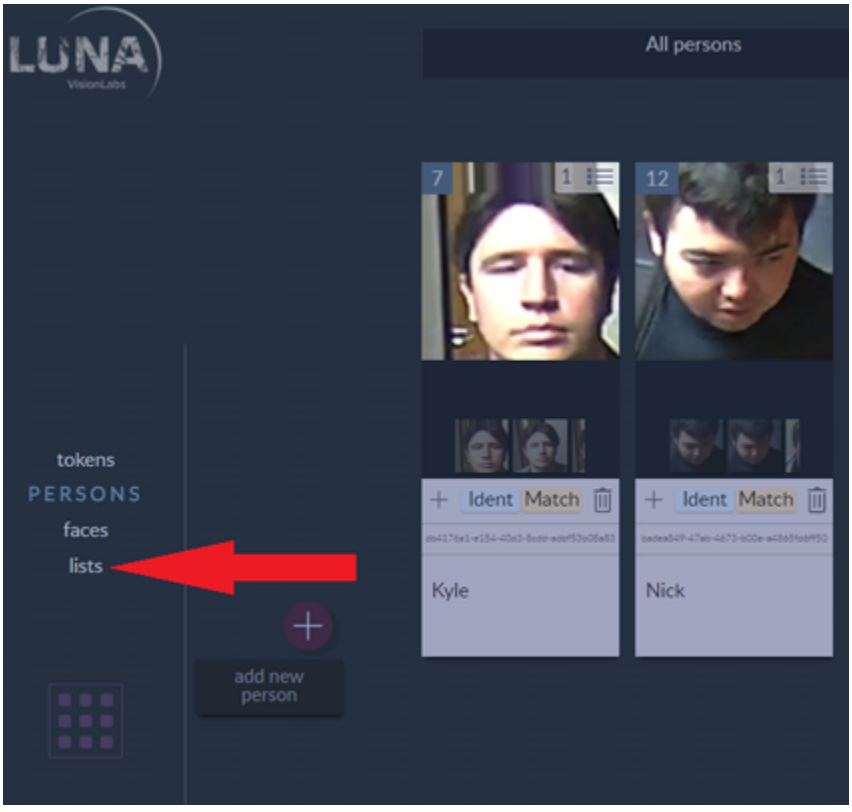
Assign photos, previously captured by the system to a new person:



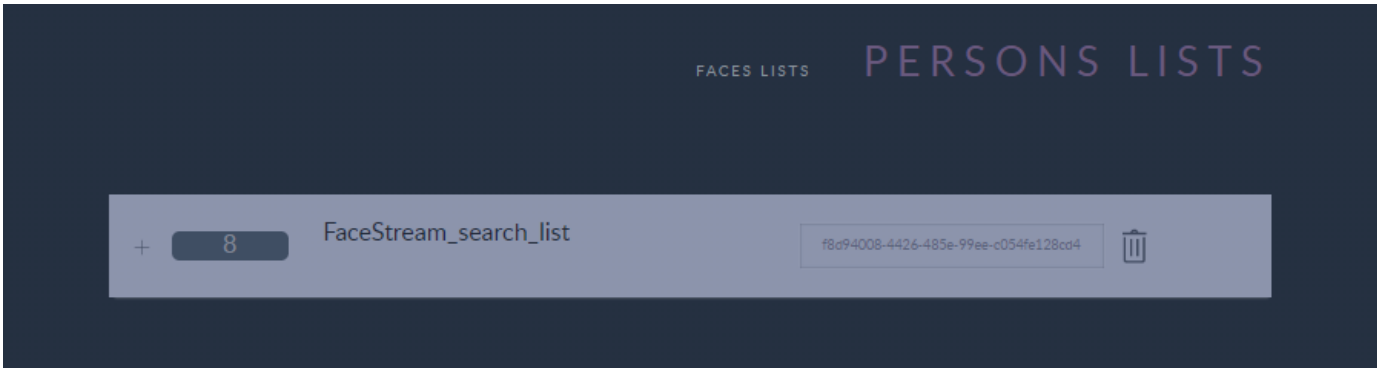
Replace the *description* text by a new person's name and check the **V** to validate:



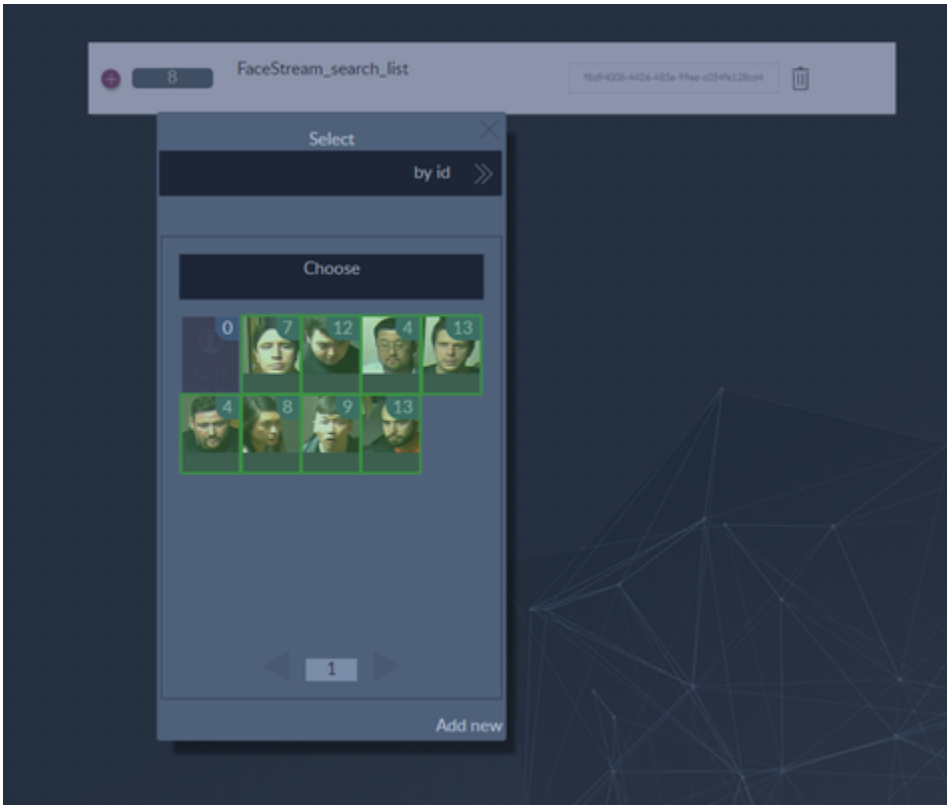
Click on **Lists** button from the right side menu and navigate to Lists:



Navigate to **Persons List** and click on a "+" sign to add a person to the list of the varified personel:



Choose from people you have assigned photos to:



Refresh the page once done.

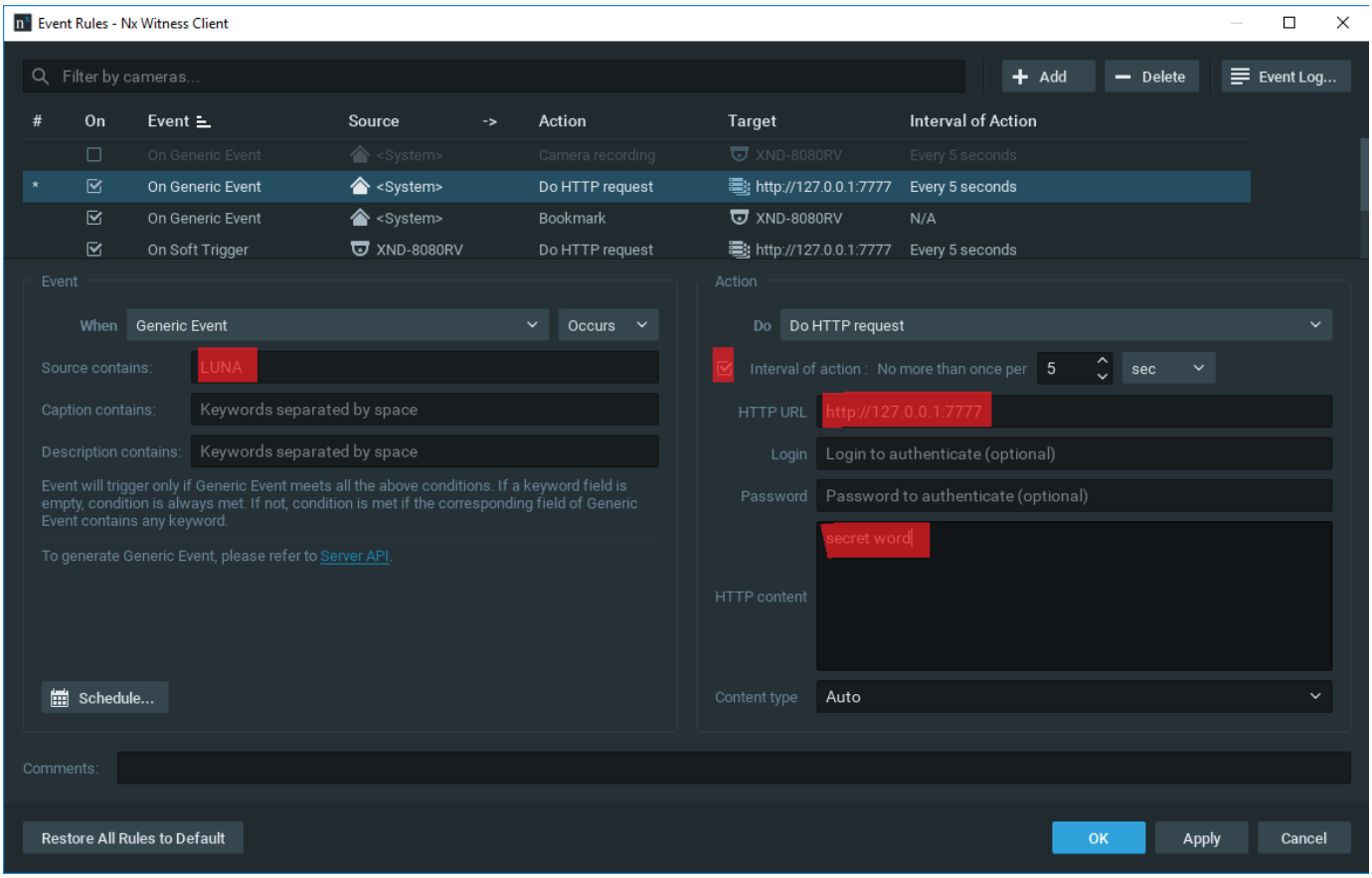
Part Three. NxWitness Desktop Client

Before we move on to the **NxWitness** part, please get familiar with the processes below:

- Installation
- Camera Configuration
- Soft Triggers
- HTTP Events

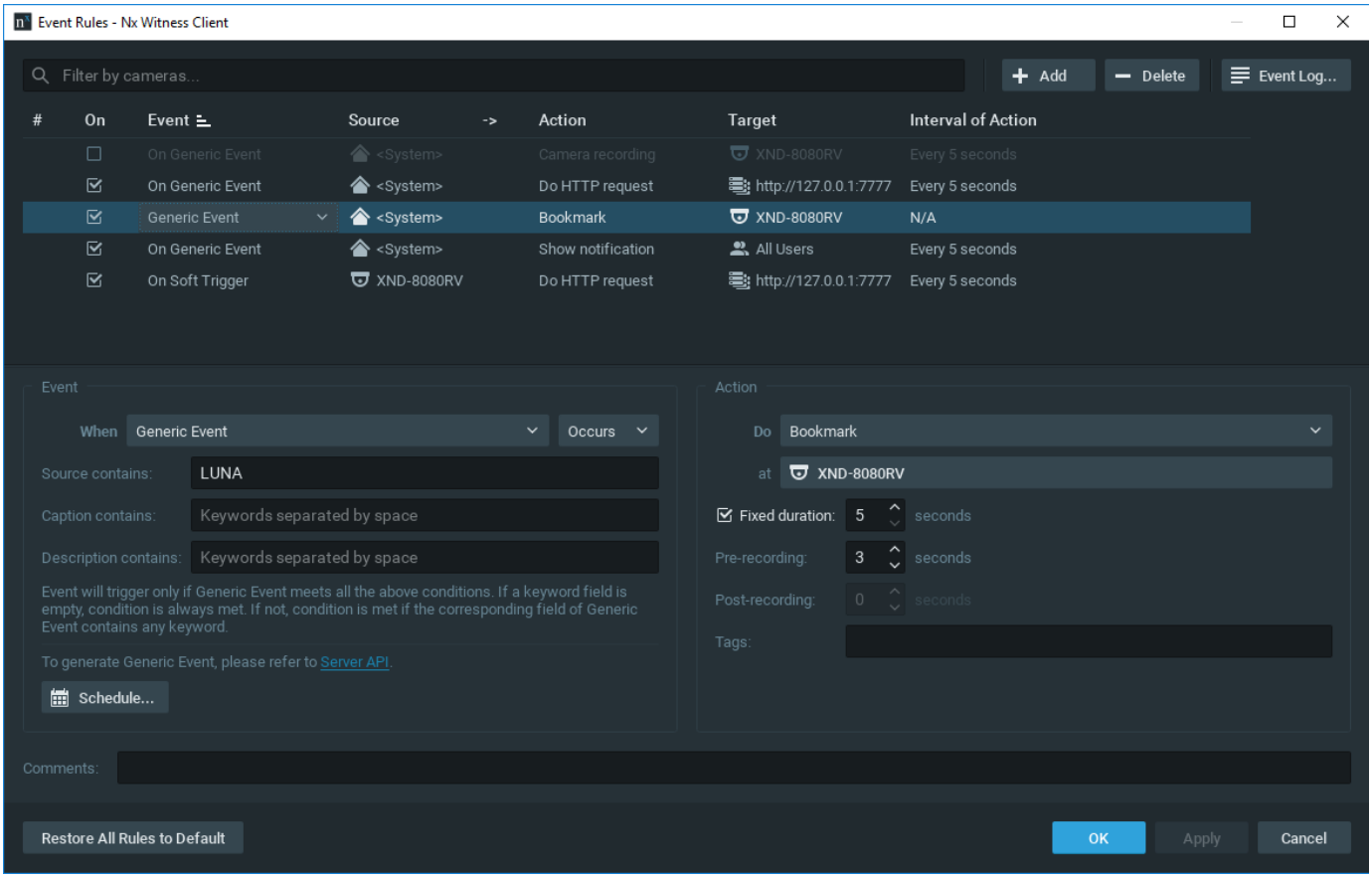
All the http requests received by the Nx Media Server in our case are treated as triggers. This way, by modifing rules within the NxWitness Desktop Client, we can schedule/define actions.

As shown below, we communicate with a third party server every time a request is received:



To make sure only the requests received from the face recognition system are being processed, we fill in the **Source** field accordingly - **LUNA**.

And here is another example of a rule, scheduled within the Desktop Client. Every time a generic event happens, the system creates a bookmark:



To make sure, we register only related events, we fill in the **Source** field accordingly - **LUNA**.