

Optic Video Surveillance

Optic Video Surveillance

Table of contents

1. Products	3
2. Optic	4
2.1 Adding a camera	5
2.2 Adding a user	7
2.3 Recording and playing the archive	9
3. Feature reference	13
3.1 Feature reference	13
3.2 Vision	14

1. Products

2. Optic

2.0.1 Optic

Optic is a lightweight foundation for a network video recorder (NVR), built on Sapsan technology. It is intended for building security and operator video surveillance systems, from ingesting video from IP cameras to recording and playing back an archive on your own servers.

Purpose

An NVR has three core tasks:

- **Video ingest** — connecting to IP cameras and receiving live video.
- **Archive recording** — storing video on disks for later playback.
- **Playback** — access to live video and recordings.

Optic combines Sapsan media technology with management through central and its own surveillance console. It provides a foundation for building a product for specific surveillance needs.

Technology

[Sapsan](#) handles video: stream ingest, DVR recording, and delivery of live video and the archive. Central provides shared streamer management. The Optic component defines the product's purpose: video surveillance.

Product family

Optic belongs to the family of products built on Sapsan technology. Each product applies the shared media technology to its own task.

Product	Purpose
Catena	A television service on your own hardware
Agora	Corporate TV and internal broadcasts
Mcaster	Preparing and delivering TV signals
Optic	A lightweight foundation for NVRs and video surveillance systems

Practical guides

- [Camera: RTSP, MBR and AAC audio](#)
- [Console users](#)
- [Archive: disks, recording and playback](#)

2.1 Adding a camera

In Optic, a camera is configured as a stream. Obtain the camera's RTSP URL and video access credentials. The camera must be reachable from the streamer that will ingest its video.

2.1.1 Connect an RTSP source

1. Open **Streams**, enter `entrance` in **New Stream Name**, and click **Create**.
2. Open the new stream and its **Sources** tab. Add an **RTSP** source.
3. Set **URL**, for example `rtsp://camera-user:camera-password@192.168.10.20:554/main`. Replace the address and credentials; obtain the actual path from the camera's settings or documentation.
4. Click **Save**. On **Basic Settings**, check the source status and the picture in the embedded player.

A single RTSP source is enough for one camera. For multiple qualities, configure MBR below.

2.1.2 Add MBR from camera streams

MBR combines existing qualities. First enable the main and substreams on the camera and obtain their RTSP URLs. For example, the main stream may be 1920×1080 and the substream 640×360.

1. On **Sources**, replace the single RTSP source with an **MBR** container.
2. Inside **MBR inputs**, add two **RTSP** sources.
3. Set the first URL to the main stream, for example `rtsp://camera-user:camera-password@192.168.10.20:554/main`.
4. Set the second URL to the substream, for example `rtsp://camera-user:camera-password@192.168.10.20:554/sub`. The `/main` and `/sub` paths are placeholders.
5. Click **Save**. Check that input statistics show both video tracks and that both qualities are available in the player.

Both inputs must be inside the same MBR container. Two separate top-level sources configure failover, rather than a quality ladder. MBR does not create resolutions or transcode video.

If the camera provides only one quality, you can apply **Standard ladder 1080/720/360 + AAC** on **Transcoder** and save. The transcoder generates those qualities and needs encoding resources on the streamer. Use MBR for existing main and substreams.

The screenshot shows the configuration interface for an MBR (Multi-Stream) container. The main container is set to 'MBR' protocol. Inside, there are two 'MBR inputs', each configured with 'RTSP' protocol and a specific URL (main and sub streams). Each input has fields for 'Via agent' and 'Peer timeout (ms)'. A '+ Add Source' button is visible at the bottom.

2.1.3 Enable audio transcoding

If the camera sends G.711 audio, transcode it to AAC for browser playback.

1. Open **Transcoder**. If no transcoder is configured, apply **Fix audio (video as is + AAC)**.
2. Check that video tracks remain **As is** and **Audio** has an **All tracks → AAC** rule. With an existing transcoder, add or change the audio rule while preserving the required video tracks.
3. Set the audio bitrate, for example 128 kbit/s, and click **Save**.
4. Check the output summary for AAC audio and video tracks. Open the player and listen; with MBR, switch between both qualities.

The transcoder outputs only tracks declared by its rules: an audio-only rule without video removes the picture. An AAC rule cannot create audio if the camera sends none. If encoding does not start, check the streamer's transcoding plugins and the error shown on the tab.

The screenshot shows the Optic Streams configuration interface. The left sidebar contains navigation links: Dashboard, Streams (selected), Templates, VOD, Auth & DRM, Sessions, Device profiles, Cluster, Streamers, Central, Settings, Cluster DVR, Layouts, Accounts, and Agents. The main content area is titled 'Streams <- entrance' and has tabs for Basic Settings, Sources, Transcoder (selected), Process, DVR, Egress, Layout, Delivery, and Ads. The Transcoder tab is divided into several sections:

- Video (ladder)**: Includes a dropdown for 'Original (no transcoding)' and a 'Codec' dropdown set to 'As is'. There are 'More' and 'X' buttons. Below this is a row of buttons for 'Add rendition': 2160p, 1080p, 720p, 576p, 360p, and a 'Custom...' button.
- Ladder logo**: A text box with instructions: 'Set once for the whole ladder: position, height and offsets are given for the top rung and recomputed proportionally on the others. Individual values live in a rung's advanced settings.' Below is an 'Assign' button.
- Audio**: Includes a 'Which tracks' dropdown set to 'All tracks', a 'Codec' dropdown set to 'aac', and a 'Bitrate (kb...)' input field set to '128'. There are 'More' and 'X' buttons. Below is a '+ Audio rule' link.
- Decoder**: Includes a 'Deinterlace' dropdown set to 'Auto (interlaced frames only)'. Below is a note: 'Interlaced sources show combing after encoding to progressive.'
- Encoding**: Includes a 'Transcode on' dropdown set to 'CPU'. Below is a note: 'CPU is the default; NVIDIA runs decode, scale and encode on the GPU.' Below this is a 'Turn off transcoder' button.

On the right side of the Transcoder tab, there is a 'Preview: input → output' box. It shows:

- Input tracks**: v1 · h264 · 1280×720, a1 · pcma, v2 · h264 · 640×360.
- Output**: All tracks (video) → As is (v1 → h264 · 1280×720, v2 → h264 · 640×360) and All tracks (audio) → aac (a1 → aac · 128 kbps).

At the top right of the Transcoder tab, there are 'Save', 'Cancel', and 'Delete' buttons.

To store video, configure the [archive](#).

2.2 Adding a user

Create a separate Optic console account for each employee. Only a **superadmin** can manage accounts.

2.2.1 Prepare access

Sign in as an existing superadministrator. On a new installation with no named accounts, sign in with the installation's static credentials and follow the console's prompt to create the first user. That user receives **superadmin**. Use this account to create further accounts.

2.2.2 Create an account

1. Open **Accounts** and click **Add account**.
2. Enter a unique employee login, for example `operator1`, and a temporary password of at least 8 characters. Repeat the password in the confirmation field.
3. Choose a role from the table below. Select **admin** for an employee who adds cameras and works with the archive through the console.
4. Enable **Require a password change at first login** and save the account.
5. Give the employee the console address and credentials. At first login, they must set their own password.

Add account

Login

operator1

Password

●●●●●●●●

At least 8 characters

Repeat the password

●●●●●●●●

Role

admin



Require a password change at first login

Cancel

Create

Role	Access
superadmin	Full management, including accounts and the license
admin	Camera and streamer configuration, archive and playback access; no account management or license replacement
viewer	System status and sessions; no access to configuration documents
security	Audit log and own session
player	Playback tickets for integrations; no installation management

The **viewer** role monitors system status; it does not provide access to camera settings or their embedded player. The **player** role is used by a portal or application to obtain playback permissions.

2.2.3 Verify login and revoke access

Ask the employee to sign in in a separate browser window and change the temporary password. For **admin**, check access to a [camera](#) and the [archive](#).

To revoke access, open **Edit** for the account, enable **Disabled**, and save. Existing sessions are closed. To reset a forgotten password, set a new one in the same form; the employee will change it at the next login.

2.3 Recording and playing the archive

Prepare the streamer (node) disks first, then enable recording for a [camera](#). Sign in as **admin** or **superadmin** to configure these settings.

2.3.1 Configure RAID on the streamer

Optic uses Sapsan application RAID: recording is distributed across independent disks. Prepare JBOD, with a separate filesystem and mount point for each disk. This setup does not need a separate `mdadm` array or hardware RAID.

This scheme does not mirror the archive: a failed disk loses its recordings, while available disks continue to be used. Plan archive backups separately.

1. Prepare the archive disk filesystems on the server and persistent mounts by UUID in `/etc/fstab`. In this example, two disks are mounted at `/storage/dvr/disk1` and `/storage/dvr/disk2`. The streamer service needs write access to these directories and to `/storage/dvr` for the metadata catalog. Preserve existing filesystems and data when working with existing recordings. For disks with an existing ext4 filesystem, create the mount points:

```
sudo mkdir -p /storage/dvr/disk1 /storage/dvr/disk2
```

Find the UUIDs with `lsblk -f` and add these entries to `/etc/fstab`, replacing `UUID_DISK1` and `UUID_DISK2` with the actual UUIDs. For another filesystem, also replace `ext4`:

```
UUID=UUID_DISK1 /storage/dvr/disk1 ext4 defaults 0 2
UUID=UUID_DISK2 /storage/dvr/disk2 ext4 defaults 0 2
```

Mount the prepared disks:

```
sudo mount /storage/dvr/disk1
sudo mount /storage/dvr/disk2
```

2. Check the actual mount points on the server:

```
lsblk -f
findmnt --mountpoint /storage/dvr/disk1
findmnt --mountpoint /storage/dvr/disk2
```

Each `findmnt` must show the separate filesystem of its archive disk. Check that these mounts also return after a server reboot.

3. In the console, open **Cluster** → **Streamers**, select the streamer and its settings. Fill in **DVR**:

Field	Example value
Archive root	<code>/storage/dvr</code>
Disk 1	<code>disk1</code>
Disk 2	<code>disk2</code>
Check mount before writing	Enabled

Use **Add disk** for the second disk. Disk paths are relative to the root: `disk1`, rather than `/storage/dvr/disk1`. The archive root holds the metadata catalog and does not replace the disk list. An empty **Catalog path** keeps metadata at `/storage/dvr/catalog`.

4. Click **Save changes** and wait for the streamer to apply the configuration. Open this streamer in **Cluster DVR** and check disk availability and capacity.

The procedure is the same for the local streamer in a single-server installation. If a disk is shown as not mounted, fix the mount; keep the check enabled so a missing disk cannot cause recording to fill the system partition.

DVR

 These settings change where the archive is written on the streamer.

Archive root
/storage/dvr

Disk 1
disk1 

Disk 2
disk2 

[+ Add disk](#) Disks are subdirectories inside the archive root: the path is relative to it.

[Move the metadata catalog to different storage...](#) ☒ Check mount before writing

Cache disks

Disks where the node keeps the remote archive it has read. Live recording never picks them; space is freed by least recent read.

[+ Add cache disk](#) Disks are subdirectories inside the archive root: the path is relative to it.

2.3.2 Enable camera recording

1. Open the camera in **Streams** and check that the RTSP source is running.
2. On **DVR**, enable **Record to DVR**.
3. Set **Retention (hours)**, for example **168** for seven days, then click **Save**.
4. Wait for a recorded range to appear and check **Current recording status**. The archive window grows from the time recording starts; seven days requires seven days of continuous recording and enough disk space.

On **Layout**, check which streamer records the archive: that streamer needs storage configured. **Cache archive** controls the read cache and does not enable camera recording.

2.3.3 Play the archive

1. Open the camera in **Streams**, then **Basic Settings** with the embedded player.
2. Find a recorded interval on the player's timeline. Move the timeline to the desired time and click a point inside an available recording.
3. Start playback. Check the selected time and picture; select a moment before the live edge to verify archive playback.
4. Click **Go live** to return to live video.

The player can only play intervals that have been recorded. If no recording is available, check the source and recording status on **DVR**, then disk availability and free space in **Cluster DVR**. For write errors, check mounts and the service's directory permissions.

Optic
26.18-374

Dashboard

Streams

Templates

VOD

Auth & DRM

Sessions

Device profiles

Cluster

Streamers

Central

Settings

Cluster DVR

Layouts

Accounts

Agents

Dark mode

Logout

LANGUAGE
English

Streams <- entrance

Save

Cancel

Delete

Basic Settings

Sources

Transcoder

Process

DVR

Egress

Layout

Delivery

Ads

Name
entrance

Template

Enabled

Start mode
As in template

Effective: Always on

Status

Stream time: from input

Input, before the transcoder 2.7Mbit/s

v1

h264 1280×720, 10 fps

a1

pcma 8 kHz, 1 ch

v2

h264 640×360, 10 fps

Output, after the transcoder 2.7Mbit/s

v1

h264 1280×720, 10 fps

2Mbit/s

v2

h264 640×360, 10 fps

701kbit/s

a1

aac 8 kHz, 1 ch

26kbit/s

Closed captions

No closed captions found in this stream.

Node contributions

STREAMER	ROLE	STATUS	BITRATE
optic-nvr	ingest	running	2687

Low latency

 off — the player steps back from the live edge, where the subtitle track reaches

Sound is off — tap to enable sound

22:28:20

22:29:30

22:29:40

22:29:50

22:30:00

22:30:10

Go live

720p

Select

- 12/19 -

© Flussonic 2026

3. Feature reference

3.1 Feature reference

- **Vision**
 - **Detect**
 - [Motion](#)
 - [Faces](#)
 - [Plates](#)
 - **Persons** — Description not yet available
 - **Match**
 - [Faces](#)
 - **List**
 - [Watchlists](#)
 - **Index**
 - [Episodes](#)

3.2 Vision

3.2.1 Detect (vision.detect)

Motion events

`vision.detect.motion`

Sapsan can discover ONVIF cameras on the network, receive their events (motion, human and vehicle detection), and check a camera for compliance.

CAMERA EVENTS

Events from ONVIF metadata become episodes with open and close:

- motion (CellMotion `IsMotion`, MotionAlarm) – `true` opens an episode, `false` closes it;
- human and vehicle detection;
- digital input triggers;
- stuck episodes are closed by an inactivity timeout.

Multiple motion sources on one camera are tracked independently.

Faces

`vision.detect.faces`

The detector locates faces in a frame. By default it analyzes the whole frame; polygonal detection zones restrict the search. Face detection alone does not identify the person.

For subsequent recognition, the face must be discernible in the image. The Watcher manual recommends at least 50–60 pixels between pupils at 720p and uniform lighting of at least 150 lux.

Plates

`vision.detect.plates`

License plate recognition processes camera video and creates events with the detected plate numbers. Events are available in Watcher and through an API for external systems.

The entire frame is searched by default. Detection can be restricted to convex, non-overlapping polygonal zones. Events can be filtered by camera, plate number, organization, and time. Selecting an event opens the corresponding camera recording.

3.2.2 Match (vision.match)

Faces

`vision.match.faces`

Recognition compares a detected face with a reference database in 1:N mode. A person record contains a reference image and attributes such as a name and identifier. People are grouped into lists.

Once camera analytics is enabled, detection works even without lists. Identification requires person records and assigned face lists. Recognition results are available as events alongside camera recordings.

3.2.3 List (vision.list)

Watchlists

`vision.list.watchlists`

Face lists group reference person records by purpose, such as department or site access level. Each record contains a face image and identification attributes.

Watcher uses these lists to match detected faces to known people. Assigning a list defines the reference set for recognition; an integrated access control system makes the decision to open a checkpoint.

3.2.4 Index (vision.index)

Episodes

`vision.index.episodes`

The event index combines analytics results and links them to camera recordings. Watcher provides separate views for motion, face recognition, and license plate events.

Filters select the camera, organization, time interval, and person name or plate number. Auto-update adds new events at the top. Motion and plate events can be exported as CSV. After archive retention expires, the event record may remain without an available preview.