AI Video Analytics

Installation and Administration Guide

Installing, updating and operating the platform

Version 2.8.17

7 October 2026

Contents

1. About This Guide

2. System Requirements

- Server
- Software installed by the installer
- Network

3. Installation

- Before you start
- Run the installer
- Installer questions
- What happens next
- Unattended installation

4. First Sign-In

- Open the web console
- Change the initial password
- Activate the AI engines
- License

5. Administration

- Users and roles
- Directory sign-in (LDAP / Active Directory)
- Security
- Audit log
- API keys
- Storage and retention
- Backup and restore
- Certificates
- Changing the branding

6. Updating

- How updates work
- Updating from the web console
- Updating from the command line
- Releases that need the installer

7. Troubleshooting and Support

- Service status and logs
- Support bundle
- Common problems
- Removing the platform

1. About This Guide

This guide is for the system administrator who installs AI Video Analytics on a server, keeps it up to date and looks after it in operation. It covers:

- what the server needs before installation,
- the installation itself and the first sign-in,
- day-to-day administration: users, security, storage, licensing, backup,
- updating the platform and collecting diagnostics for support.

Configuring analytics, reviewing alarms and building automation flows are covered in the **User Guide**. Connecting video management systems (VMS), e-mail and other systems is covered in the **Integration Guide**.

> **Note:** Paths in this guide are written as `<install-dir>` — the directory where you extracted the installation package. The installer shows it as **Install directory** in its summary before it starts.

2. System Requirements

Server

Item	Requirement
Operating system	Ubuntu 22.04 LTS or 24.04 LTS (Server or Desktop edition)
Architecture	x86_64 (amd64)
Memory	16 GB or more recommended
Free disk space	50 GB or more on the system disk for the platform itself
GPU	NVIDIA GPU with a working driver (`nvidia-smi` must list the card)
NVIDIA driver	Version 535 or newer
Access	A root shell (`sudo`) on the server
Internet	Required during installation and updates to download container images

Alarm images and evidence are stored separately from the system disk when a data disk is configured (see **Storage and Retention**). Size that disk from your expected alarm volume and retention period.

> **Important:** Without an NVIDIA GPU the platform can run in **CPU mode**, but detection accuracy and throughput are significantly reduced. CPU mode is intended for evaluation only.

Hardware sizing for a given number of cameras and analyses depends on the analysis types and frame rates in use. Ask your supplier for a sizing estimate before ordering hardware.

GPU memory decides how many analyses can run at the same time. Face recognition and plate recognition each need a sizeable share of it on top of the camera streams. When an analysis does not fit, it is not started half-way: its page shows **Not running** with the reason, and it starts by itself when memory frees up.

Software installed by the installer

You do not need to install these yourself — the installer checks for them and installs or upgrades them when necessary:

- Docker Engine and the Docker Compose plugin,
- NVIDIA Container Toolkit (so that containers can use the GPU),
- the GPU device configuration used by the containers.

The NVIDIA **driver** itself is not installed by the installer. Install it first and confirm it works:

```
nvidia-smi
```

The command must print a table listing your GPU. If it does not, fix the driver before continuing.

Network

Only one service is published on the server's network interface: the web gateway.

Profile	HTTP	HTTPS
Standard	80/TCP	443/TCP
Alternative	8000/TCP	8443/TCP

HTTP only redirects to HTTPS. All other services (database, message broker, cache, analytics) communicate on an internal container network that is not reachable from outside the server. Their connections are encrypted and the identity of each service is verified with an internal certificate authority created at installation.

The server also needs outbound access to:

Destination	Purpose
Cameras (usually RTSP, TCP 554)	Receiving video streams
the vendor's container registry	Downloading container images during installation and updates
VMS servers (for example Nx, TCP 7001)	Sending alarms and analytics objects, if integrated
SMTP server	Alarm e-mails, if configured
LDAP / Active Directory (TCP 389 or 636)	Directory sign-in, if configured

3. Installation

Before you start

Tick every item before running the installer:

Requirement	Done
Ubuntu 22.04 or 24.04 LTS installed, x86_64	
NVIDIA driver 535 or newer installed	
`nvidia-smi` lists the GPU	
Ports 80/443 (or 8000/8443) are free and allowed by the firewall	
Internet access to the container registry	
Installation package copied to the server	
Optional: data disk attached for alarm images	

Run the installer

1. Copy the installation package to the server and extract it into the directory you want to use as the install directory.
2. Open a root shell and change into that directory:

```
sudo -i
cd <install-dir>
```

3. Start the installer:

```
./install.sh
```

The installer first runs **pre-flight checks** — root access, operating system, architecture, memory, free disk space, Docker, GPU and NVIDIA Container Toolkit. Warnings do not stop the installation; errors do, with an explanation.

Installer questions

The installer asks a few questions. Press **Enter** to accept the default shown in brackets.

GPU not found. If no NVIDIA GPU is detected you can continue in CPU mode or exit and install a GPU and driver first. The default is to exit.

Port profile. Choose **Standard** (80/443) unless those ports are already used on the server, in which case choose **Alternative** (8000/8443).

Language. The language the web console starts in: English, Turkish, Spanish, Arabic or Traditional Chinese. Each user can change it later.

Branding. How the product names itself in the web console and in integrated VMS (analytics engines, object search, menus):

Choice	Result
Unbranded	Neutral names such as "AI Traffic Analytics"; no vendor name in the console
Vendor brand	The supplier's product name and logo
Your company	Your company name, a short name for VMS menus, and optionally your logo, favicon and sign-in background

Branding is read-only in the web console. It can only be changed afterwards with `set_brand.sh` (see **Changing the Branding**).

Data disk. If an unmounted disk is found, the installer offers to mount it persistently for alarm images. You can also do this later (see **Storage and Retention**).

Before it starts, the installer shows a **summary** — host address, ports, language, branding, analytics mode, install directory and log file — and asks for confirmation.

What happens next

After confirmation the installer runs without further questions:

- installs or upgrades Docker, Docker Compose and the NVIDIA Container Toolkit where needed,
- generates all passwords, keys and certificates (they are stored in `<install-dir>/.env` and never shown on screen),

- downloads the container images, showing progress,
- starts the platform and verifies that every service is healthy,
- installs two small host services: the in-app updater and the support bundle collector.

Depending on the internet connection the download takes from several minutes to around an hour. When it finishes the installer prints the web address and the initial sign-in details.

> **Important:** `<install-dir>/.env` contains every secret of the installation. Keep its permissions as the installer set them, include it in your server backups and never send it to anyone — including support.

Unattended installation

For scripted deployments run `./install.sh --unattended` and set the answers as environment variables:

Variable	Values
`INSTALL_PORTS`	`standard` (default) or `alt`
`INSTALL_LANG`	`en` (default), `tr`, `es`, `ar`, `zh-TW`
`INSTALL_BRAND_MODE`	`generic` (default) or `custom`
`INSTALL_BRAND_NAME`	Company name (required for `custom`)
`INSTALL_BRAND_PREFIX`	Short name for VMS menus (`custom`; default: the name)
`INSTALL_BRAND_LOGO` / `_FAVICON` / `_LOGIN_BG`	Image files (`custom`; optional)
`INSTALL_ALLOW_CPU`	`1` to allow CPU mode without a GPU
`INSTALL_DATA_DISK`	Partition for alarm images, for example `/dev/sdb1`
`PULL_RETRIES`	Image download attempts (default 4)

Run `./install.sh --help` for the full list.

4. First Sign-In

Open the web console

Open a browser and go to:

- `https://<server-address>` for the Standard profile, or
- `https://<server-address>:8443` for the Alternative profile.

The installation uses a self-signed certificate, so the browser shows a one-time security warning. To avoid it, replace the certificate with one issued by your organisation (see **Certificates**).

Change the initial password

Sign in with the initial account printed by the installer (user `admin`, password `admin`). You are immediately asked to set a new password; nothing else is available until you do. The new password must satisfy the password policy (see **Security**).

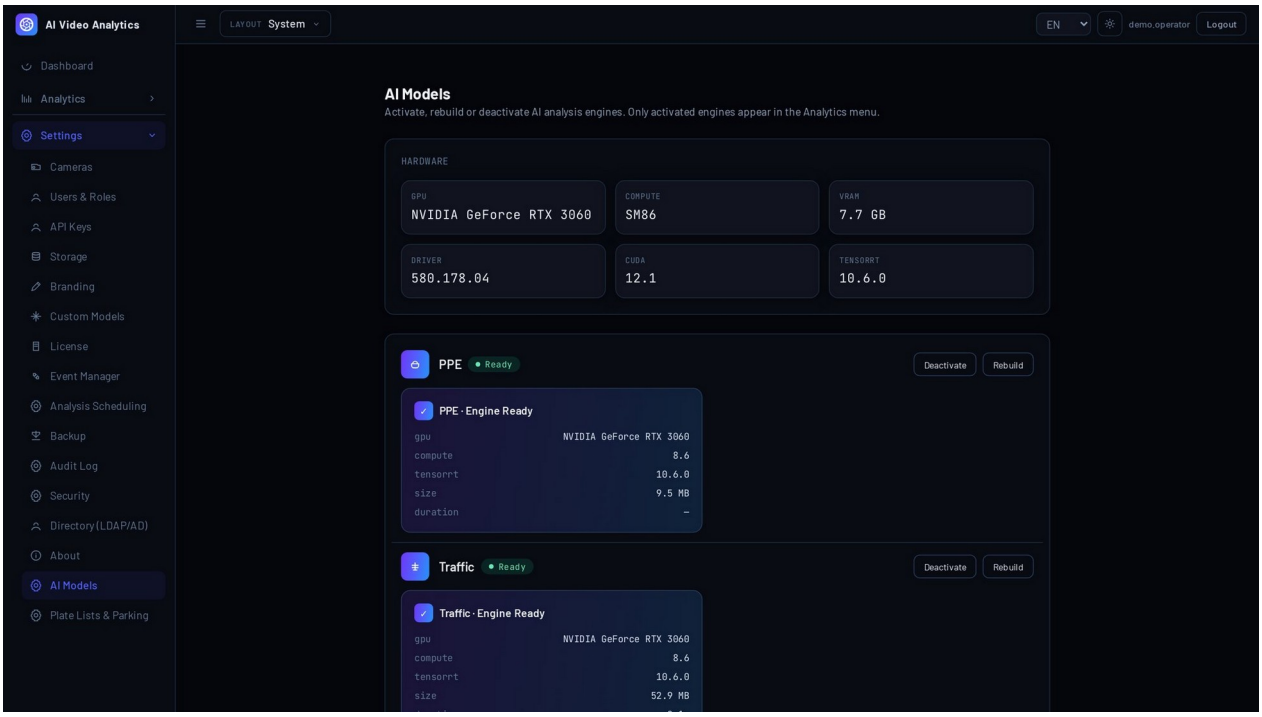
Activate the AI engines

On a new installation the **first-run setup wizard** opens after sign-in:

1. It detects the GPU.
2. You select the analysis types you want to use.

3. It builds an optimised AI engine for each selected type on this GPU, showing each phase as it runs.

Building takes a few minutes per analysis type. An analysis type appears in the menu only after its engine is ready. You can activate, deactivate or rebuild engines at any time under **Settings → AI Models**.



Settings → AI Models: detected hardware and the engine of each analysis type

Face recognition and plate recognition are calibrated rather than built, and calibration needs the GPU completely free. Starting it opens a **maintenance window**: the console lists the analyses and cameras that will stop and the expected duration (about 15–20 minutes). With one click everything is stopped, calibrated and restarted automatically. No alarms are produced during the window, so plan it outside busy hours. Face and plate recognition can be calibrated in the same window.

> **Note:** Engines are built for the exact GPU model and driver. After replacing the GPU or upgrading the driver, rebuild them under **Settings → AI Models**.

License

Under **Settings → License**:

1. **Generate Request** creates a request file bound to this installation.
2. Send the request file to your supplier.
3. **Upload License** installs the signed license you receive.

License Status shows what the license allows. Starting an analysis beyond the licensed limits is refused with a message.

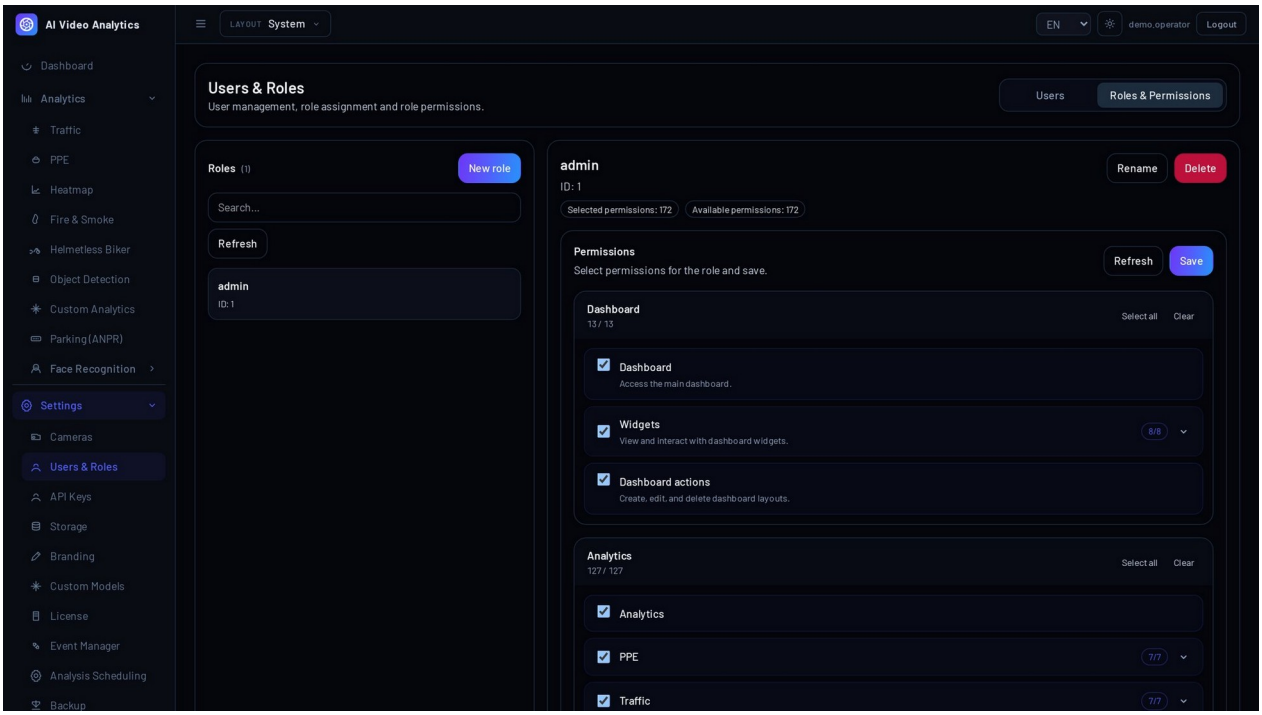
5. Administration

The **Settings** menu contains all administrative functions. What each user sees depends on their role.

Users and roles

Settings → Users & Roles manages local accounts, roles and permissions.

- A **role** is a set of permissions; users receive one or more roles.
- Seeing a menu item and performing actions in it are separate permissions. For example a user can be allowed to view integrations without being allowed to change their credentials.
- Assigning roles and permissions is itself a separate permission. Give it only to administrators — anyone who holds it can extend their own access.
- An administrator can reset a user's password. The user must then set a new one at the next sign-in.

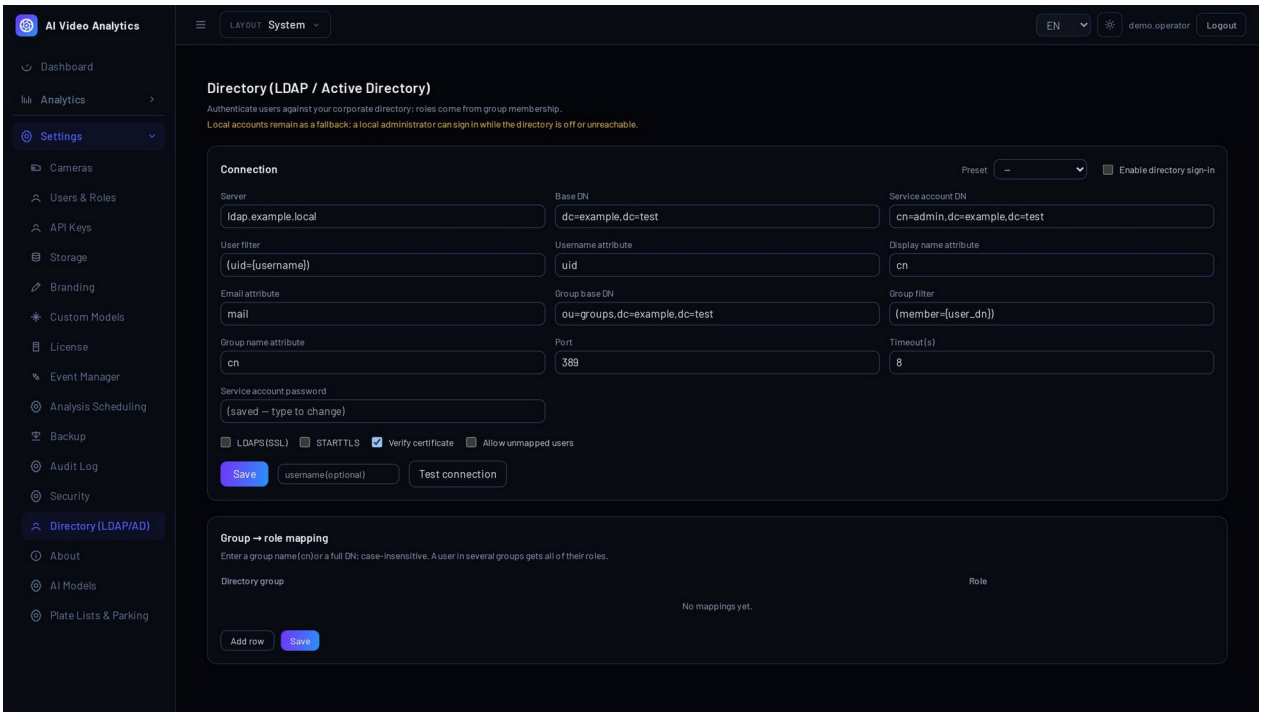


Roles & Permissions: permissions grouped by area, with view and action permissions separate

Directory sign-in (LDAP / Active Directory)

Settings → Directory (LDAP/AD) lets users sign in with their corporate account.

- Passwords are checked by the directory; they are never stored by the platform.
- Roles come from directory **group membership** and are re-synchronised at every sign-in, so removing someone from a group removes their access at their next sign-in.
- Users whose groups are not mapped to any role are refused.
- Local accounts keep working when the directory is unreachable. Keep at least one local administrator account.



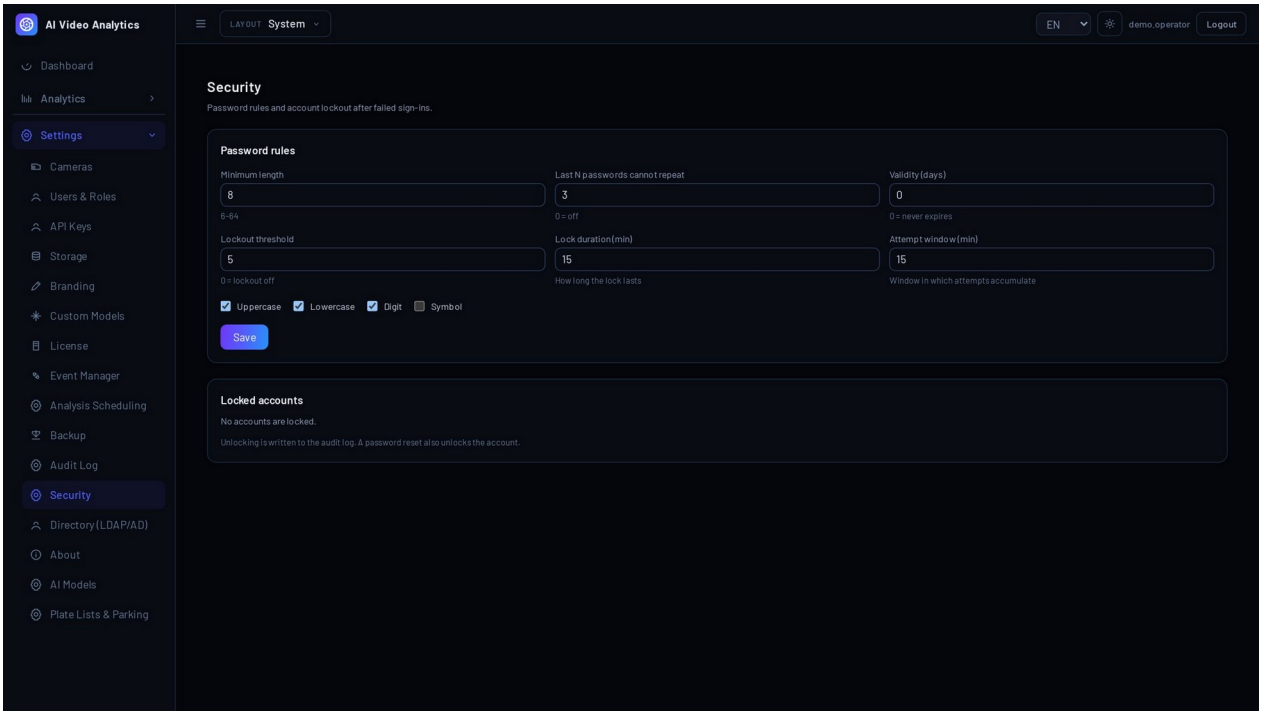
Settings → Directory (LDAP/AD): connection and group-to-role mapping

Security

Settings → **Security** sets the password policy and account lockout:

- minimum length and required character classes,
- password history (reuse of recent passwords is refused),
- password expiry — an expired password must be changed at the next sign-in,
- lockout after repeated failed sign-ins, and a list of locked accounts that an administrator can unlock.

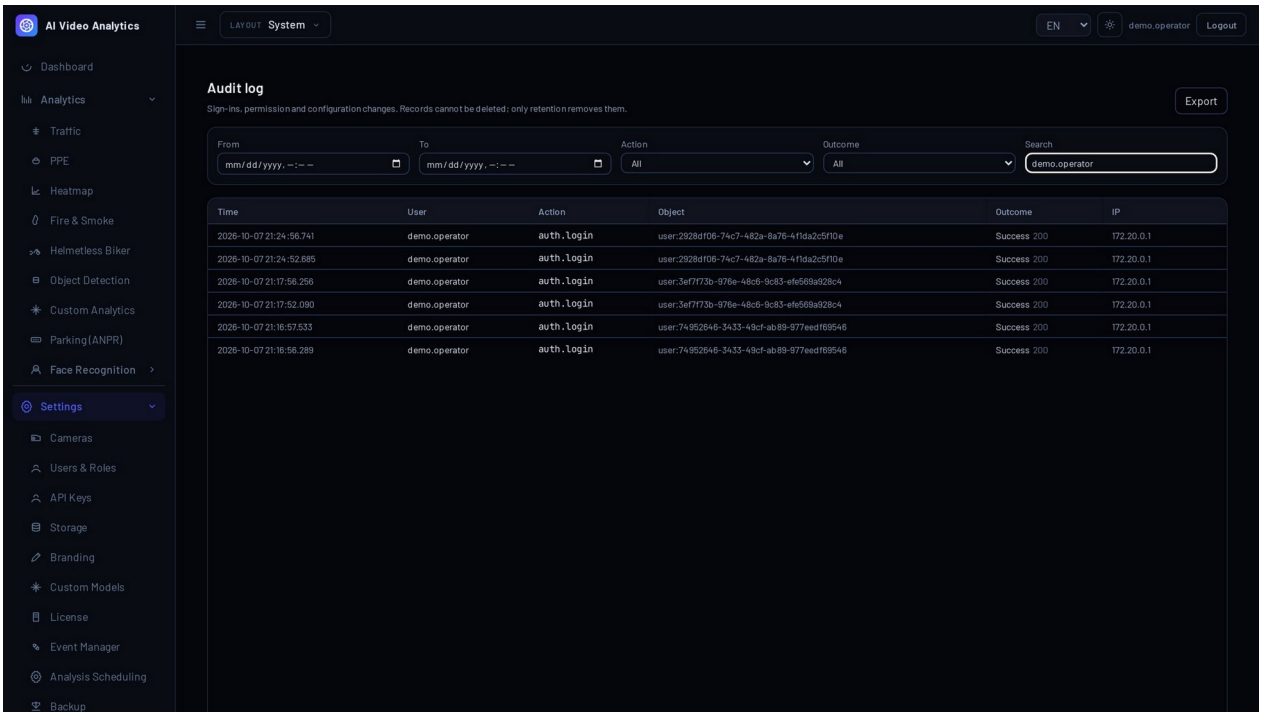
A locked account cannot sign in even with the correct password until the lockout period ends or an administrator unlocks it.



Settings → Security: password rules and locked accounts

Audit log

Settings → **Audit Log** records every configuration change made through the console or the API — who, what, when and from where — plus sign-ins, failed sign-ins and lockouts. Each entry names the signed-in user who made the change. Secrets are masked before they are written. The log can be filtered and exported to CSV. It is kept for the same period as other metadata (see **Storage and Retention**).



Settings → Audit Log

API keys

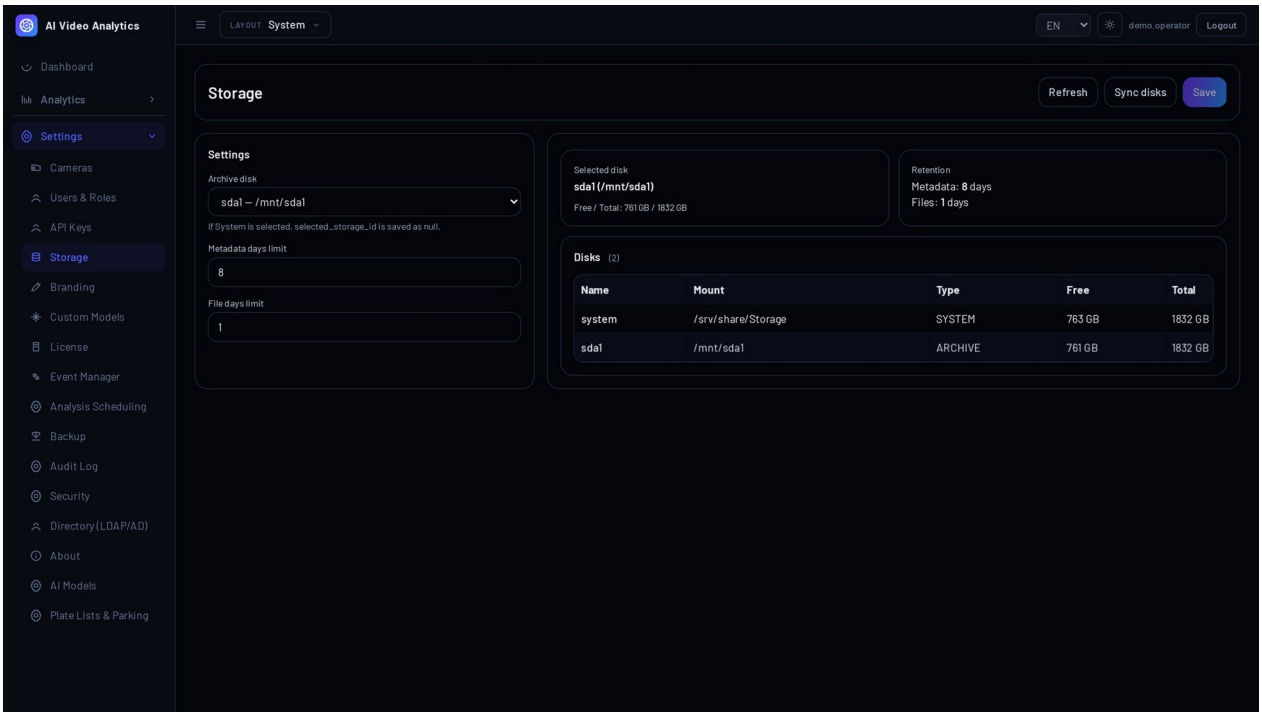
Settings → **API Keys** creates keys for integrators and scripts. A key can be limited to specific permissions and given an expiry date. The API reference is available from the same page to users who hold the API documentation permission.

Storage and retention

Settings → **Storage** selects where alarm images are stored and how long data is kept:

- **File retention** — alarm images older than this are deleted.
- **Metadata retention** — alarm records and the audit log older than this are deleted.

Clean-up runs automatically every hour.



Settings → Storage: archive disk, retention and the disks found on the server

Platform files (AI engines, custom models, license, branding) are kept in `/srv/ai-analytics-share`. Alarm images are kept on the selected storage path.

> **Note:** An installation upgraded from an earlier version keeps its original directory, file and service names. The names in this guide apply to new installations.

> **Warning:** If a disk is attached but not mounted, the storage page shows it under **Unmounted disks** and warns when the selected path actually resides on the system disk. Writing alarm images to the system disk can fill it and stop the server. Mount the disk persistently first — the easiest way is the installer:

```
cd <install-dir>
sudo ./install.sh --mount-disk
```

Backup and restore

Settings → **Backup** exports cameras, analysis configurations, users, roles, integrations and all system settings to a single JSON file, and restores such a file on this or another installation.

- A restore **replaces** the current configuration and cannot be undone.
- The account you are signed in with keeps its password, so a restore cannot lock you out.
- Alarm images, AI engines and the license are **not** part of this backup. Back up the storage path and `<install-dir>/.env` with your usual server backup.
- After a restore, refresh the page; running streams and analyses may need to be restarted.

Certificates

The web console uses a self-signed certificate generated at installation. It is valid for 397 days — the longest period current browsers accept.

> **Important:** When the self-signed certificate expires, browsers refuse the connection. Re-running `./install.sh` in `<install-dir>` renews it (existing settings and data are kept). Better still, install a certificate issued by your organisation.

To use your own certificate, place it in `<install-dir>/certs/`, replacing the two existing files and keeping their names:

- `<install-dir>/certs/gateway.crt` — the certificate, followed by any intermediate certificates,
- `<install-dir>/certs/gateway.key` — the private key, not password-protected.

Then restart the gateway:

```
cd <install-dir>
sudo docker compose restart gateway
```

The installer never replaces a certificate it did not create itself, so your certificate survives re-installation and updates. Renewing it before it expires is your responsibility.

Changing the branding

Branding can only be changed on the server:

```
cd <install-dir>
sudo ./set_brand.sh --mode generic
sudo ./set_brand.sh --mode custom --name "Your Company" --prefix YOURCO \
     --logo logo.png --favicon favicon.ico --login-bg background.png
```

Only the web and application services are restarted; cameras and analyses keep running. VMS integrations pick up the new names automatically within a few minutes.

6. Updating

How updates work

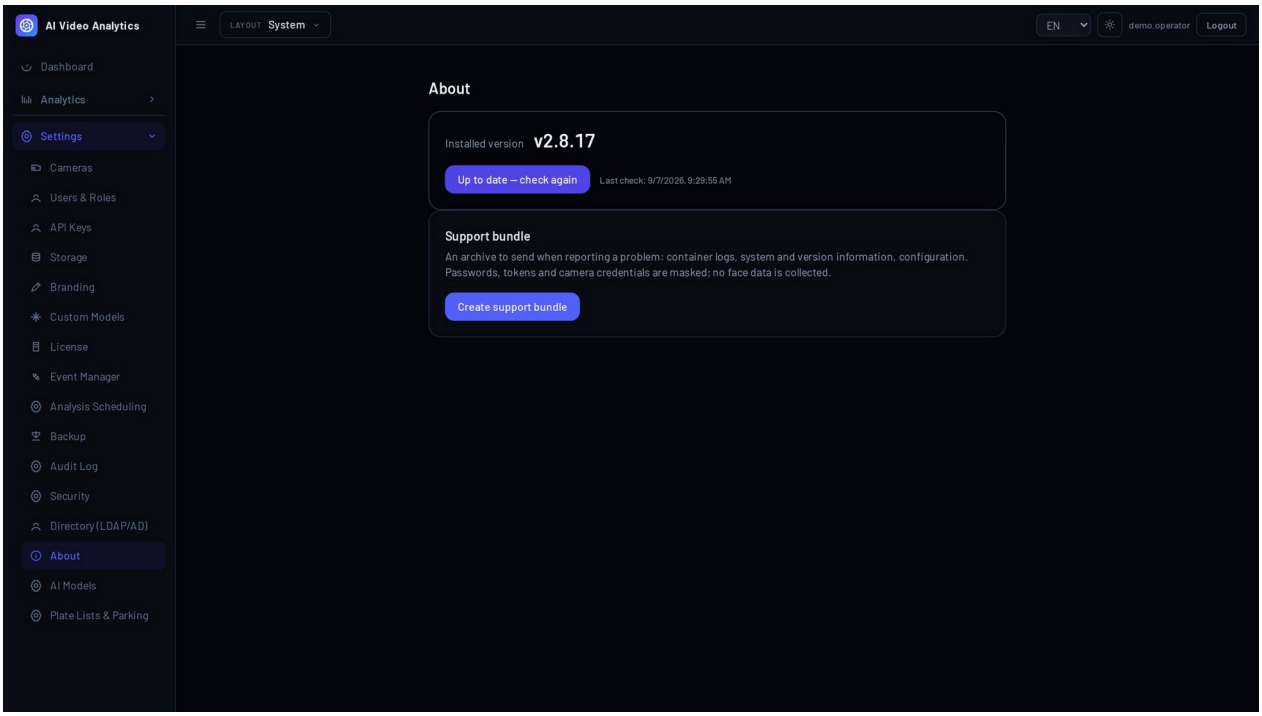
Every release is described by a **release manifest** signed by the supplier. It lists the exact identity of every container image in the release. Before any service is changed, the update procedure:

1. downloads the new images,
2. downloads the signed release manifest,
3. verifies the signature against the supplier key installed with the platform,
4. checks that every downloaded image is exactly the image in the manifest,
5. refuses a release older than the one already installed.

If any check fails the update stops **before** any service is changed, and the running system is left as it was.

Updating from the web console

Settings → **About** shows the installed version and whether a newer one is available. **Update** runs the update on the server in the background; the page shows progress and reloads when the platform is back. Do not close the page during the update.



Settings → About: version, update check and support bundle

Some releases change the deployment configuration, not only the images. For these the console says the version cannot be applied from the interface — run the installer package on the server instead (below).

Updating from the command line

```
cd <install-dir>
sudo ./update.sh
```

Option	Effect
(none)	Download, verify and apply; then remove old images
`--no-clean`	Keep old images (allows a quick rollback)
`--dry-run`	Show what would be downloaded and removed without changing anything
`--clean-only`	Only remove old images

Only services whose image changed are recreated. Analyses resume automatically afterwards.

Releases that need the installer

When the release notes or the console say a release **requires the installer**, extract the new installation package over the existing install directory and run `./install.sh` again. Existing secrets, settings and data are kept; the installer only adds what the new release needs.

> **Warning:** If a signature or image check fails, do not try to work around it. It means the downloaded content is not the release your supplier published. Contact support with the update log.

7. Troubleshooting and Support

Service status and logs

```
cd <install-dir>
sudo docker compose ps
sudo docker logs server --tail 200
```

The main containers are `server`, `api`, `vca` (analytics), `stream` (camera streams), `integration`, `front` (web console) and `gateway`.

> **Important:** Always use `--tail` with `docker logs`. After an unclean shutdown the log reader can stop silently at a damaged entry without `--tail`, and it then looks as if nothing was logged.

Support bundle

Settings → About → Support bundle collects container logs, version and hardware information and configuration into one archive for your supplier. Passwords, tokens and camera credentials are masked, the contents of `.env` are never included (only the names of its settings), and no face data is collected.

Common problems

Symptom	What to check
An analysis type is missing from the menu	Its AI engine is not built yet — **Settings → AI Models**
An analysis shows **Not running**	The reason is shown next to the badge. Usually the GPU is full: stop another analysis or add GPU capacity; the analysis then starts by itself
Starting an analysis is refused with a license message	**Settings → License → License Status**
No alarms although the analysis is running	The camera stream is reachable (RTSP address, credentials); the analysis region and classes are set
Storage page warns about the system disk	Mount the data disk (`./install.sh --mount-disk`)
Update stops at "image verification"	Do not bypass it; send the update log to support
Browser security warning	Expected with the self-signed certificate; replace it with your own

Removing the platform

`purge.sh` in the install directory removes the installation completely: containers, images, the shared data directory **including the license and all AI engines**, alarm images on the data disk, the host services, and Docker Engine itself. It cannot be undone. Take a configuration backup and a copy of `.env` first, and use it only when you really intend to remove the platform from the server.