# WolfISMS

Wolf's Information Security Management System

# Docker Installation

## Version 4.00

As of: November 2026

© 2012–2026 WolfInfoSec GmbH

# 1. Overview

This guide describes how to run WolfISMS as a Docker container behind the Nginx Proxy Manager (NPM). NPM handles TLS termination and certificate management, while WolfISMS itself speaks plain HTTP inside the Docker network.

It is aimed at administrators who operate or newly set up a Docker environment; chapter 3 describes how to prepare an Ubuntu server as the Docker host. For the classic installation directly on an Ubuntu or Debian server with Apache and mod_wsgi, see chapter 3 of the administration documentation instead.

## 1.1 Target Architecture

| Component | Role |
|---|---|
| **Nginx Proxy Manager** | Accepts HTTP (80) and HTTPS (443) from the outside, terminates TLS, manages the Let's Encrypt certificates and forwards requests to the WolfISMS container on the internal network. The administration interface runs on port 81 |
| **WolfISMS container** | The application itself. It is not directly reachable from outside, only through the Docker network proxy |
| **Database** | MariaDB or MySQL – either as its own container or as an existing database server, switched via DB_MODUS (see chapter 6) |
| **Docker network "proxy"** | Connects NPM and WolfISMS. It is created manually once and referenced as an external network by both Compose projects |

## 1.2 Why HTTPS Is Mandatory

WolfISMS sets its session cookies with the attribute Secure=True. Browsers transmit such cookies only over encrypted connections. Over plain HTTP the login page will therefore load, but signing in fails because the session cannot be retained. A valid certificate is not a recommendation but an operating prerequisite.

**Reverse proxy support is built in:**
The application evaluates the proxy's headers X-Forwarded-For, X-Forwarded-Proto, X-Forwarded-Host and X-Forwarded-Port (ProxyFix in isms.py). It therefore correctly recognises that it runs behind a TLS termination point, generates absolute links using https and the public host name, and counts the rate limit per client rather than per proxy. ISMS_PROXY_HOPS tells it how many proxies are in front – the image sets 1 (chapter 9.6). The Nginx Proxy Manager sets these headers by default – no additional configuration is required.

© 2012–2026 WolfInfoSec GmbH

# 2. Prerequisites

| Component | Requirement |
|---|---|
| **Docker Engine** | Current version with the Compose V2 plug-in 2.20 or newer (docker compose, not docker-compose) |
| **Operating system** | Ubuntu Server 24.04 LTS recommended (setup see chapter 3); any other Linux Docker host works as well |
| **Public DNS record** | The desired host name, e.g. isms.your-domain.com, must resolve to the public IP address of the Docker host |
| **Open ports** | Ports 80 and 443 must be reachable from the outside. Port 80 is required for the Let's Encrypt ACME challenge |
| **Port 81** | The NPM administration interface. It should not be publicly reachable but restricted to the internal network or VPN |
| **Installation package** | The WolfISMS repository or Docker package containing docker-compose.yml and .envdefault (provided by the vendor) |
| **Database** | MariaDB 10.11+ or MySQL 8.0+ – as a container or as an external server |

**Do not expose port 81 to the internet:**
The administration interface of the Nginx Proxy Manager controls all certificates and forwarding rules. Restrict access to the internal network using the host firewall, and change the default credentials immediately after the first sign-in.

© 2012–2026 WolfInfoSec GmbH

# 3. Preparation – Install Ubuntu Server and Docker

This chapter sets up a new Docker host. If you already run a Linux server with Docker Engine and the Compose plug-in (2.20 or newer), skip it and continue with step 1. The description covers Ubuntu Server 24.04 LTS; the commands apply unchanged to 22.04 LTS.

## 3.1 Server Resources

| Resource | Recommendation |
|---|---|
| **CPU** | 2 virtual cores, 4 for more than 20 concurrent users |
| **Memory** | 4 GB, preferably 8 GB with the internal database container |
| **Disk** | 40 GB for system, images and database, plus space for uploaded documents and backups |
| **Network** | Fixed IPv4 address (static or via DHCP reservation) and internet access for package sources, Docker images and Let's Encrypt |

## 3.2 Install Ubuntu Server

The installation image "Ubuntu Server 24.04 LTS" is available at https://ubuntu.com/download/server. Write it to a USB stick or attach it as a virtual drive in your hypervisor (VMware, Proxmox, Hyper-V). The installer asks for the following:

- **Language and keyboard**: language of your choice, matching keyboard layout
- **Type of installation**: Ubuntu Server (not the minimized variant)
- **Network**: set the interface to "Edit IPv4" → Manual and enter address, gateway and DNS servers – or keep DHCP if the DHCP server reserves the address
- **Proxy and mirror**: leave empty or accept the default
- **Storage**: Use an entire disk with "Set up this disk as an LVM group". The installer initially uses only part of the volume group; extend the logical volume to the full size in the storage summary
- **Profile**: name, server name (e.g. isms), user name and password of the administration account. root has no password; administration is done with sudo
- **Ubuntu Pro**: skip (Skip for now)
- **SSH**: enable "Install OpenSSH server" so the server can be managed remotely afterwards
- **Featured server snaps**: select nothing – in particular not the docker snap

**Do not install Docker as a snap:**
The docker snap offered by the installer keeps data and configuration below /var/snap, does not support all Compose features and has not been tested with this guide. Docker is installed from the official Docker repository in the section Install Docker Engine.

After the reboot, the administrator signs in via SSH (ssh <user>@<server-ip>).

## 3.3 Update the System and Basic Settings

First bring all packages up to date and set the time zone. The time zone determines the time in the host's logs; certificates and two-factor codes also require a synchronised clock, which Ubuntu already maintains via systemd-timesyncd.

```
sudo apt update && sudo apt full-upgrade -y
sudo timedatectl set-timezone Europe/Berlin
```

© 2012–2026 WolfInfoSec GmbH

```
timedatectl
sudo reboot
```

Ubuntu installs security updates automatically via unattended-upgrades. To check that the service is active:

```
sudo dpkg-reconfigure -plow unattended-upgrades
```

## 3.4 Configure the Firewall

The host firewall ufw allows SSH as well as HTTP and HTTPS for the Nginx Proxy Manager. Allow SSH before enabling it, otherwise you lock yourself out:

```
sudo ufw allow OpenSSH
sudo ufw allow 80/tcp
sudo ufw allow 443/tcp
sudo ufw enable
sudo ufw status verbose
```

**ufw does not protect Docker ports:**
Docker writes published ports (ports: in docker-compose.yml) directly into iptables, bypassing ufw. A ufw rule for port 81 therefore has no effect. Bind the NPM administration interface to the internal address instead, e.g. "10.0.0.5:81:81" instead of "81:81" in step 2, or shield it with the upstream firewall.

## 3.5 Install Docker Engine

The Ubuntu packages docker.io and docker-compose-v2 are older than Docker's own and are removed first, if present. On a freshly installed server the command merely reports that the packages are not installed:

```
for pkg in docker.io docker-doc docker-compose docker-compose-v2 \
           podman-docker containerd runc; do
  sudo apt-get remove -y $pkg
done
```

Then add the official Docker repository and its signing key:

```
sudo apt-get update
sudo apt-get install -y ca-certificates curl git
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg \
     -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc
echo "deb [arch=$(dpkg --print-architecture) \
  signed-by=/etc/apt/keyrings/docker.asc] \
  https://download.docker.com/linux/ubuntu \
  $(. /etc/os-release && echo "$VERSION_CODENAME") stable" \
  | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

And install Docker Engine with the Compose plug-in:

© 2012–2026 WolfInfoSec GmbH

```
sudo apt-get update
sudo apt-get install -y docker-ce docker-ce-cli containerd.io \
     docker-buildx-plugin docker-compose-plugin
```

## 3.6 Verify the Installation and Authorise the User

```
sudo systemctl enable --now docker
docker --version
docker compose version
sudo docker run --rm hello-world
```

docker compose version must report 2.20 or newer (required for the database switch in step 3). hello-world prints "Hello from Docker!". To let the administrator use Docker without sudo, add the account to the docker group; the membership takes effect only after signing in again:

```
sudo usermod -aG docker $USER
exit
# sign in again via SSH, then:
docker ps
```

**The docker group means root privileges:**
Anyone allowed to use Docker can start any container with access to the host's file system. Add administrators only.

The host is now prepared. Continue with the Docker network in step 1.

© 2012–2026 WolfInfoSec GmbH

# 4. Step 1 – Create the Docker Network

Both Compose projects – the Nginx Proxy Manager and WolfISMS – reference the same external network. It is created once on the Docker host:

```
docker network create proxy
```

Verify that it exists:

```
docker network ls | grep proxy
```

> The network must exist before the first docker compose up. Without it, both projects abort with the message "network proxy declared as external, but could not be found".

© 2012–2026 WolfInfoSec GmbH

# 5. Step 2 – Deploy the Nginx Proxy Manager

Create a dedicated directory for the reverse proxy on the Docker host and place the file docker-compose.yml inside it:

```
mkdir -p /opt/npm
cd /opt/npm
```

Contents of /opt/npm/docker-compose.yml:

```
services:
  npm:
    image: jc21/nginx-proxy-manager:2
    restart: unless-stopped
    ports:
      - "80:80"     # HTTP + ACME challenge
      - "443:443"   # HTTPS
      - "81:81"     # administration interface
    volumes:
      - npm_data:/data
      - npm_letsencrypt:/etc/letsencrypt
    networks:
      - proxy

networks:
  proxy:
    external: true
    name: proxy

volumes:
  npm_data:
  npm_letsencrypt:
```

Then start the container:

```
docker compose up -d
```

## 5.1 First Sign-In

The administration interface is then available at http://<docker-host>:81. On first start the Nginx Proxy Manager creates a default account whose credentials are documented with the image.

> Change the email address and password of the default account immediately after the first sign-in. As long as the default values apply, anyone with network access to port 81 can modify the certificates and forwarding rules of the entire environment.

© 2012–2026 WolfInfoSec GmbH

# 6. Step 3 – Deploy WolfISMS

The WolfISMS package contains docker-compose.yml as well as the template .envdefault. The actual configuration is done exclusively through environment variables in the file .env.

## 6.1 Obtain the Package and Create the Configuration

```
git clone https://github.com/wolfinfosec/wolfisms.git
cd wolfisms
cp .envdefault .env
```

Then open .env and adapt the values to your environment. Chapter 8 contains the complete reference of all variables.

If you need your own certificates – for example to connect to an external database server – copy them into the directory db-certs/ (see Database: Internal or External). This is not required for outward-facing TLS termination; the Nginx Proxy Manager takes care of that.

## 6.2 Database: Internal or External

The variable DB_MODUS in the .env decides whether MariaDB runs as a container within the stack. docker-compose.yml includes the file docker-compose.db-<DB_MODUS>.yml for this. If the entry is missing – for example in a .env from an older version –, internal applies; existing installations therefore keep running unchanged.

| DB_MODUS | Behaviour |
|---|---|
| **intern (default)** | docker-compose.db-intern.yml: container db with MariaDB 11.4 and volume db_data. On first start it creates database and user and imports a dump from ./db-init. DB_HOST stays empty, DB_ROOT_PASSWORD is mandatory |
| **extern** | docker-compose.db-extern.yml: no database container is started. App and cron connect to DB_HOST and DB_PORT; DB_ROOT_PASSWORD is not used |

For an external database, database and user must already exist on the server, for example:

```
CREATE DATABASE isms CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
CREATE USER 'isms'@'%' IDENTIFIED BY '<password>';
GRANT ALL PRIVILEGES ON isms.* TO 'isms'@'%';
```

- **Connection**: DB_HOST, DB_PORT, DB_NAME, DB_USER and DB_PASSWORD in the .env; DB_TYPE stays mariadb (or mysql). The values apply to app and cron alike
- **Database on the same host**: DB_HOST=localhost points to the container itself inside the container – enter the LAN IP of the host. MariaDB must listen on that address (bind-address), port 3306 must be open in the firewall for the Docker network, and the user must be allowed from the Docker network, e.g. 'isms'@'172.16.0.0/255.240.0.0' with the same GRANT as above. If the container shares the database with an existing installation, do not start the cron service (docker compose up -d --scale cron=0) – otherwise every cron job runs twice and reminder emails go out twice
- **TLS**: DB_SSL_MODE=required encrypts without checking the certificate, verify-full checks certificate and host name against DB_SSL_CA. The certificate files belong in the folder db-certs/ next to docker-compose.yml; inside the container they are found under /var/www/isms/db-certs (read-only), e.g. DB_SSL_CA=/var/www/isms/db-certs/db-ca.pem

© 2012–2026 WolfInfoSec GmbH

- **Data backup**: the backup in the admin area uses the same connection. The user also needs read access to routines, triggers and events for it – GRANT ALL on the database is sufficient
- **Moving data**: an existing installation moves by importing its dump (backup in the admin area) on the new server before DB_MODUS is switched
- **Prerequisite**: include and the optional dependency (depends_on … required: false) require Docker Compose 2.20 or newer

**Switching the mode:**
First docker compose down (the volume db_data is kept), then change DB_MODUS in the .env, then docker compose up -d. The file docker-compose.db-extern.yml contains the service db only as a disabled placeholder (profile nie-starten) – please do not add that profile to COMPOSE_PROFILES.

## 6.3 Start the Containers

```
docker compose up -d
```

Check the start-up:

```
docker compose ps
docker compose logs -f
```

## 6.4 Connecting to the Reverse Proxy

For the Nginx Proxy Manager to reach the application container, both must be on the proxy network and the container must be addressable under a stable name. That name is set through the variable PROXY_ALIAS and entered in the proxy host as the forward hostname. The supplied docker-compose.yml already joins the network; the following example shows the relevant sections:

```
services:
  wolfisms:
    # ... image, volumes, depends_on ...
    env_file: .env
    restart: unless-stopped
    networks:
      proxy:
        aliases:
          - ${PROXY_ALIAS}    # e.g. wolfisms-app

networks:
  proxy:
    external: true
    name: proxy
```

**Application port on 127.0.0.1 only:**
The WolfISMS container is addressed exclusively through the internal proxy network. docker-compose.yml publishes port 8000 only on 127.0.0.1 of the Docker host – for diagnostics, e.g. with curl on the host or through an SSH tunnel; it cannot be reached from outside. Please do not change this binding: a port reachable from outside would bypass TLS termination, make signing in unusable because of the Secure cookies and – since the application trusts the proxy's X-Forwarded-* headers – allow any client to forge its IP address or an HTTPS connection.

© 2012–2026 WolfInfoSec GmbH

The authoritative docker-compose.yml is part of the installation package. The example above shows only the sections relevant to the proxy connection and does not replace the supplied file.

© 2012–2026 WolfInfoSec GmbH

# 7. Step 4 – Configure the Proxy Host and Certificate

Publication is then configured in the NPM interface under Hosts → Proxy Hosts → Add Proxy Host.

## 7.1 "Details" Tab

| Field | Value |
|---|---|
| **Domain Names** | isms.your-domain.com |
| **Scheme** | http |
| **Forward Hostname** | wolfisms-app (the name set in PROXY_ALIAS) |
| **Forward Port** | 8000 |
| **Block Common Exploits** | enabled |
| **Websockets Support** | enabled |

## 7.2 "SSL" Tab

- **SSL Certificate**: select "Request a new SSL Certificate"
- **Force SSL**: enable – redirects HTTP requests to HTTPS automatically
- **HTTP/2 Support**: enable
- **Email address**: provide it for the Let's Encrypt expiry notifications and accept the terms of service

For the standard challenge Let's Encrypt requires the domain name to resolve publicly to the host IP address and port 80 to be reachable from the outside. If that is not the case – for example in a purely internal environment – configure a DNS challenge with your DNS provider instead, or upload your own certificate.

[ IMAGE PLACEHOLDER ]

**Figure: Creating a proxy host**

*Screenshot of the "Add Proxy Host" dialog in the Nginx Proxy Manager with the "Details" tab filled in*

[ IMAGE PLACEHOLDER ]

**Figure: Requesting a certificate**

*Screenshot of the "SSL" tab with the options "Request a new SSL Certificate", "Force SSL" and "HTTP/2 Support"*

© 2012–2026 WolfInfoSec GmbH

## 7.3 Verify the Installation

| Check | Expected result |
|---|---|
| **Open https://isms.your-domain.com** | The login page appears and the certificate is shown as valid |
| **Request the same address over HTTP** | Automatic redirect to HTTPS (Force SSL) |
| **Sign in** | Redirect to the dashboard – if signing in fails, see chapter 10 |
| **Container log** | docker compose logs without traceback entries |
| **Admin area** | The "Updates" menu entry is absent – this is intended in container mode (see chapter 8) |

**Immediately after the first sign-in:**
1) Change the default password of the admin user
2) Install the licence file customer.lic (Admin → Licence)
3) Restart the container so the licence and the secret key derived from it take effect: docker compose restart

© 2012–2026 WolfInfoSec GmbH

# 8. Operating in Container Mode

With the environment variable WOLFISMS_CONTAINER=1 the application recognises that it runs inside a container (config.container_mode). This changes its behaviour in two places.

## 8.1 In-App Updater Disabled

The entire update area including /admin/updates is disabled and responds with HTTP 404; the tile in the admin area is hidden. This is intentional: an update applied inside a running container would be lost the next time the image is restarted.

In Docker operation, updates are applied exclusively through a new image:

```
docker compose pull
docker compose up -d
```

## 8.2 Configuration Comes Only From the Environment

Outside Docker, config.py reads the credentials from the file /etc/wolfisms/secret.env. Inside the container that file usually does not exist – which is not an error: if it is missing, the application uses only the environment variables that are set. Those are exactly what .env provides through docker compose.

**Different path:**
If you do want to use an env file inside the container, its path can be set with the variable WOLFISMS_ENV_FILE. Environment variables that are already set always take precedence over the values from the file.

## 8.3 Data Persistence and Backup

The database and the upload directory must be persisted. Both have to be mounted as volumes, otherwise they are lost when the container is replaced.

| Data | Location | Note |
|---|---|---|
| **Database** | Volume of the database container or external database server | Contains all ISMS content |
| **Uploaded files** | UPLOAD_DIR (default /var/www/isms/wolfisms/uploads) | Attachments, playbooks, logo, templates and letterhead |
| **Sessions** | SESSION_DIR (default /var/www/isms/wolfisms/sessions) | Must be writable; losing it merely signs all users out. The application sets the directory to 0700 and the session files to 0600 at startup |
| **Licence file** | license/customer.lic in the application directory | After switching images, check that it is still present |

The backup itself works as in the classic installation, through the admin area at /admin/backup or the direct link /backup/download. The archive contains the database dump and the upload

© 2012–2026 WolfInfoSec GmbH

directory; application code, configuration and the licence file are not included and must be backed up separately. Chapter 5 of the administration documentation covers the details.

## 8.4 Logs

| Purpose | Command |
|---|---|
| **Follow live output** | docker compose logs -f |
| **Application only** | docker compose logs -f wolfisms |
| **Last 200 lines** | docker compose logs --tail=200 |
| **Proxy access and certificates** | docker compose logs -f npm (in the directory /opt/npm) |

© 2012–2026 WolfInfoSec GmbH

# 9. Configuration Reference (.env)

All parameters are set as environment variables. They are resolved in this order: environment variables that are already set, then the env file, and finally the defaults defined in the code. There is deliberately no default for DB_PASSWORD – if the variable is missing, the application will not start.

## 9.1 Mandatory Settings

| Variable | Example | Meaning |
|---|---|---|
| **DB_PASSWORD** | (strong password) | Database password – the only value without a default |
| **DB_USER** | isms | Database user (default: wolf) |
| **DB_HOST** | db | Empty = internal container db; with DB_MODUS=extern host name or IP of the database server |
| **DB_NAME** | isms | Name of the database |
| **WEBPATH** | https://isms.your-domain.com | Public address of the instance with https:// and without a trailing slash – exactly as it appears in the browser. After sign-in the application redirects there, links in emails build on it. A sub-path such as /isms is not supported; use a (sub)domain of its own |
| **ISMS_SECRET_KEY** | (empty) | Optional. Leaving it empty is fine: on first start the application creates a session key and stores it in the volume sessions, where it survives restarts and updates. Only set it if several instances need the same key |
| **WOLFISMS_CONTAINER** | 1 | Enables container mode (see chapter 8) |
| **PROXY_ALIAS** | wolfisms-app | Network alias of the container; enter it as the forward hostname in the proxy host |

If ISMS_SECRET_KEY is to be set, generate a random value like this:

```
python3 -c "import secrets; print(secrets.token_hex(32))"
```

## 9.2 Database

| Variable | Default | Meaning |
|---|---|---|
| **DB_MODUS** | intern | intern = MariaDB container within the stack, extern = existing database server (see the section Database: Internal or External) |
| **DB_TYPE** | mariadb | mariadb or mysql; mssql is reserved and currently rejected at start-up |
| **DB_PORT** | (empty) | Empty = 3306 |
| **DB_ROOT_PASSWORD** | – | intern only: root password of the database container |
| **DB_SSL_MODE** | (empty) | empty = unchanged, disabled, required or verify-full |

© 2012–2026 WolfInfoSec GmbH

| Variable | Default | Meaning |
|---|---|---|
| **DB_SSL_CA** | (empty) | CA file inside the container, mandatory for verify-full |
| **DB_SSL_CERT / DB_SSL_KEY** | (empty) | Client certificate and key, if the server requires them |

Invalid values – an unknown DB_TYPE, a wrong DB_SSL_MODE or verify-full without DB_SSL_CA – stop the application from starting at all, with a clear message in the log.

## 9.3 Paths and Instance

| Variable | Default | Meaning |
|---|---|---|
| **ISMS_BASE_URL** | (empty) | Only needed if WEBPATH is not a full address: base of the links in emails. If WEBPATH holds an address with https://, that one is used and ISMS_BASE_URL is ignored |
| **PATHNAME** | isms | Directory name of the instance |
| **UPLOAD_DIR** | /var/www/isms/wolfisms/uploads | Path for uploaded files – mount as a volume |
| **SESSION_DIR** | /var/www/isms/wolfisms/sessions | Path for server-side sessions – must be writable |
| **WOLFISMS_ENV_FILE** | /etc/wolfisms/secret.env | Alternative path of an env file inside the container |

## 9.4 Language

| Variable | Default | Meaning |
|---|---|---|
| **DEUTSCH** | True | Allow the German user interface |
| **ENGLISCH** | False | Allow the English user interface. With DEUTSCH=0 and ENGLISCH=1 the installation runs in English only |

## 9.5 Single Sign-On (optional)

| Variable | Meaning |
|---|---|
| **MS_AUTH** | 1 = enable sign-in through Microsoft Entra ID |
| **GOOGLE_AUTH** | 1 = enable sign-in through Google |
| **SSO_ALLOWED_DOMAINS** | Comma-separated allow list of permitted email domains. An empty list admits nobody – that is the safe default |
| **SSO_AUTO_CREATE** | 1 = create unknown SSO users automatically (default 0) |
| **MS_TENANT_ID / MS_CLIENT_ID / MS_CLIENT_SECRET / MS_REDIRECT_URI** | Credentials and callback URL for Microsoft |
| **GOOGLE_CLIENT_ID / GOOGLE_CLIENT_SECRET / GOOGLE_REDIRECT_URI** | Credentials and callback URL for Google |

© 2012–2026 WolfInfoSec GmbH

The callback URL must point to the public host name served by the reverse proxy – not to the container name.

## 9.6 Further Variables

| Variable | Meaning |
|---|---|
| **UPDATE_PROXY** | Proxy for outbound connections. In container mode it has no effect on the disabled updater |
| **RATELIMIT_STORAGE_URI** | Storage for the rate limit counters. The default memory:// counts per worker process – with several workers every limit is multiplied accordingly. The image starts three workers; as long as memory:// applies, a warning is therefore logged at start-up. Point it at shared storage, e.g. redis://redis:6379 |
| **ISMS_PROXY_HOPS** | Number of reverse proxies in front of the application whose X-Forwarded-* headers are trusted. The image sets 1 (the Nginx Proxy Manager). With two proxies in a row, enter 2; 0 switches the evaluation off. Set too high, a client could send the headers itself and forge its IP or the host name |
| **ISMS_DEBUG** | 1 = switch on Flask debug mode. For troubleshooting only; remove it afterwards |

ISMS_DEBUG must not remain set in production. Debug mode enables PROPAGATE_EXCEPTIONS, disables template caching and is a finding in every hardening baseline.

© 2012–2026 WolfInfoSec GmbH

# 10. Troubleshooting

| Symptom | Cause and remedy |
|---|---|
| **"network proxy declared as external, but could not be found"** | The Docker network is missing. Create it with docker network create proxy (chapter 4) and start the containers again |
| "permission denied while trying to connect to the Docker daemon socket" | The user is not in the docker group or has not signed in again since usermod (chapter 3). Sign in again or run the command with sudo |
| **NPM reports "502 Bad Gateway"** | The application container cannot be reached under the configured forward hostname. Check that the container is running, that it is on the proxy network, and that PROXY_ALIAS matches the name entered in the proxy host |
| **The login page appears but signing in fails silently** | You are accessing the site over HTTP instead of HTTPS. Session cookies are Secure=True and are not set over unencrypted connections. Enable "Force SSL" in the proxy host |
| **The container does not start and the log mentions the database** | DB_PASSWORD is missing or the database is not ready yet. The variable is mandatory and deliberately has no default; check the start order via depends_on |
| “depends on undefined service db” or an error at include | Docker Compose is older than 2.20, or one of the files docker-compose.db-intern.yml / docker-compose.db-extern.yml is missing, or DB_MODUS has a value other than intern or extern |
| Log: “WARNUNG: Datenbank-Host 'db' ist nicht auflösbar” | DB_MODUS=extern is set but DB_HOST is empty – or DB_HOST=db with an external database. Set DB_HOST to the database server, or DB_MODUS=intern for the internal database |
| **The certificate request fails** | The DNS record does not point to the host, or port 80 is not reachable from the outside. Check both or switch to a DNS challenge |
| **Links inside the ISMS point to http:// or the container name** | The proxy does not pass X-Forwarded-Proto or X-Forwarded-Host, or WEBPATH is set incorrectly. Check both |
| **The "Updates" menu entry is missing in the admin area** | Not an error: in container mode the in-app updater is disabled. Update through a new image (chapter 8.1) |
| **Demo mode persists after installing the licence** | The licence check is cached. Restart the container: docker compose restart |
| **Uploaded files disappeared after an update** | UPLOAD_DIR was not mounted as a volume. Add the volume and restore the files from the most recent backup |
| Sign-in succeeds but the page jumps straight back to the login | WEBPATH differs from the address in the browser – another host name, access via the IP or still the default from .envdefault. The redirect after sign-in then leads to a host without the session cookie. Set WEBPATH exactly to the address you open and run docker compose up -d |
| External database: “Access denied for user 'isms'@'172.…'” | The database user only exists for localhost or another address; the connection from the container comes from the Docker network. Create the user for that network as well (chapter 6.2) and check what arrives in the container: docker compose exec app printenv DB_USER DB_PASSWORD DB_HOST |
| Proxy and application respond | Usually a name lookup is hanging. MariaDB resolves the |

© 2012–2026 WolfInfoSec GmbH

| Symptom | Cause and remedy |
| --- | --- |
| very slowly | address of every client in reverse: set skip-name-resolve in the MariaDB configuration (then create users with an IP, a network or %, not with host names). If the Proxy Manager waits for IPv6 answers, add DISABLE_IPV6: 'true' under environment in its service and recreate it with docker compose up -d --force-recreate |
| Sign-in attempts end with “too many requests” (429) although hardly anyone is working | The rate limit counts per client IP. If the client address does not arrive – for example because ISMS_PROXY_HOPS=0 was set or a second proxy sits in front – the application only sees the proxy IP and all users share one quota. Set ISMS_PROXY_HOPS to the number of proxies and check that the proxy forwards X-Forwarded-For |

© 2012–2026 WolfInfoSec GmbH

# 11. Checklist

| ✓ | Task |
|---|---|
| ☐ | DNS record resolves to the public IP address of the Docker host |
| ☐ | Ports 80 and 443 are reachable from the outside |
| ☐ | Port 81 is restricted to the internal network |
| ☐ | Ubuntu Server installed and updated, ufw active, Docker Engine and Compose plug-in 2.20+ installed from the Docker repository |
| ☐ | Docker network proxy created |
| ☐ | Nginx Proxy Manager started, default credentials changed |
| ☐ | .env created from .envdefault and fully populated |
| ☐ | DB_PASSWORD set |
| ☐ | WOLFISMS_CONTAINER=1 set |
| ☐ | PROXY_ALIAS set and entered as the forward hostname in the proxy host |
| ☐ | WEBPATH set to the public address (with https://, without a trailing slash) |
| ☐ | Volumes mounted for the database and UPLOAD_DIR |
| ☐ | With an external database: DB_MODUS=extern, DB_HOST/DB_PORT set, database and user created, log without connection errors |
| ☐ | Proxy host created, certificate requested, Force SSL and HTTP/2 enabled |
| ☐ | Sign-in over HTTPS tested successfully |
| ☐ | Default password of the admin user changed |
| ☐ | Licence file installed and the container restarted |
| ☐ | Second admin account created as emergency access |
| ☐ | Backup via /admin/backup tested and stored off-site |
| ☐ | ISMS_DEBUG not set |
| ☐ | .env not committed to a Git repository |

© 2012–2026 WolfInfoSec GmbH

# 12. Further Documentation

| Document | Content |
|---|---|
| **WolfISMS Administration Documentation** | Complete system administration: user and permission management, backup, licence management, REST API, two-factor authentication, single sign-on, AI settings and security notes. It also explains what differs in container mode compared with the classic installation on Apache with mod_wsgi |
| **WolfISMS User Documentation** | Description of all functional modules for end users |
| **WolfISMS REST API** | Interface description for connecting external systems: incidents, measures, assets, risks and suppliers – with all fields, filters, value lists and a complete example sync client |

**WolfISMS – Wolf's Information Security Management System**
© 2012–2026 WolfInfoSec GmbH
Update server: https://update.wolfisms.de
Version: 4.00 | As of: November 2026