

Stream Suite Administrator Manual

Dokument Typ: Manual

Version: 3.3

Datum: 02.08.2026

Autor: Stefan Pfütze

Disclaimer: Copyright by x-dream-media GmbH. All Rights Reserved.
The information in this document may not be changed or
published without agreement of x-dream-media GmbH.

Table of contents

<u>1</u>	<u>System Overview</u>	<u>6</u>
1.1	Architecture	6
1.2	Farm Architecture	7
1.2.1	Transcoding Farm	7
1.2.2	Streaming Farm (optional)	7
1.3	Job Lifecycle	7
1.4	ABR Quality Levels	8
1.5	URL Overview	8
<u>2</u>	<u>Installation (Fresh Install)</u>	<u>9</u>
2.1	System Requirements	9
2.1.1	Master Server	9
2.1.2	Worker Node (Transcoding Farm)	9
2.1.3	Streamer Node (Streaming Farm, optional)	9
2.2	Installing the Master Server	10
2.3	Setting Up SSL/HTTPS (Recommended)	10
2.4	Adding a Worker Node	10
2.5	Adding a Streamer Node (Streaming Farm)	11
2.6	Firewall Configuration	11
2.7	GPU Encoding (Optional)	12
2.8	Service Management	12
<u>3</u>	<u>Updating an Existing Installation</u>	<u>13</u>
3.1	Full Update	13
3.2	Selective Update	13
3.3	Automatic Database Migrations	14
<u>4</u>	<u>Transcoding Farm</u>	<u>15</u>
4.1	Enabling the Farm	15
4.2	Managing Workers	15
4.3	Slot Tracking	15
4.4	VOD Recording and Reconnect Logic	16
4.4.1	Automatic Reconnect (30-Second Window)	16
4.5	Worker Failure	16
<u>5</u>	<u>Streaming Farm</u>	<u>17</u>
5.1	Streamer Pool Management	17
5.2	Load Balancing	17
5.3	Heartbeat Monitoring	17

6 Super Administration 18

6.1	First Login	18
6.2	Display Language	18
6.3	Dashboard	18
6.4	Managing Tenants	18
6.5	Managing Users	19
6.6	Live Sources	19
6.7	Stream Control	19
6.8	Stream Relays	20
6.9	Recordings	20
6.10	Tenant Export / Import	20
6.10.1	Export	20
6.10.2	Import	20
6.11	Transcoding Parameters	21
6.11.1	Encoder & global settings	21
6.11.2	Quality levels	21
6.11.3	Source specific overwriting	22
6.12	Statistics	22

7 Troubleshooting 23

7.1	502 Bad Gateway	23
7.2	RTMP Stream Active, Player Shows Nothing	23
7.3	SRT Stream: Connection Error	23
7.4	VOD Not Playable	23
7.5	Stream Relay does not start	24
7.6	Farm Jobs hang	24
7.7	Thumbnails are missing	24
7.8	Diagnosis Overview	24
7.9	Other Issues	25

8 WebRTC / WHIP Ingest 26

8.1	Architecture	26
8.2	Requirements	26
8.2.1	Server	26
8.2.2	Encoder Requirements (H.264)	27
8.3	Creating Sources	27
8.4	OBS Studio with WHIP Plugin	27
8.4.1	Setup	27
8.4.2	Encoder Settings	27
8.5	Browser-Based WHIP Client	28
8.5.1	Access	28
8.5.2	Features	28
8.6	Stability and Reconnect	28
8.7	Troubleshooting	29

9 Tenant Portal 30

9.1	Login and Display Language	30
9.2	OBS Configuration	30
9.3	Stream Control	30
9.4	Stream Relays	30
9.5	Recordings	31
9.6	Account Settings	31

[10 Viewer Frontends](#) 32

10.1	Normal Player (/live/SLUG)	32
10.2	TV Player (/tv/SLUG)	32
10.3	Grid Player (/grid/SLUG)	33
10.4	Embed Player (/embed/SLUG/SOURCE-ID)	33
10.4.1	Embed-Code (fix size)	33

[11 Public REST API](#) 34

11.1	GET /api/tenant/:slug — Tenant Info	34
11.2	GET /api/tenant/:slug/live — Live Sources	34
11.3	GET /api/tenant/:slug/vods — Recordings	34
11.4	POST /api/track — Start Tracking Session	34
11.4.1	POST /api/track/update — Heartbeat	34
11.5	GET /api/streamer-redirect — Streamer Failover	35

[12 Uninstallation](#) 36

12.1	Overview: What Gets Installed Where	36
12.2	Data Backup Before Uninstallation	36
12.2.1	Back up the database	36
12.2.2	Back up recordings	37
12.3	Uninstalling the Master Server	37
12.3.1	Step 1: Stop and disable services	37
12.3.2	Step 2: Remove service files	37
12.3.3	Step 3: Remove application files	37
12.3.4	Step 4: Remove data (only if no backup is needed)	37
12.3.5	Step 5: Remove MediaMTX	37
12.3.6	Step 6: Clean up NFS configuration	37
12.3.7	Step 7: Restore nginx configuration	38
12.3.8	Step 8: Remove sudo permissions	38
12.3.9	Step 9: Remove optional packages	38
12.4	Uninstalling a Worker Node	38
12.4.1	Step 1: Stop and disable service	38
12.4.2	Step 2: Remove application files	38
12.4.3	Step 3: Remove NFS mount	39
12.4.4	Step 4: Remove sudo permissions	39
12.4.5	Step 5: Remove worker from admin backend	39
12.5	Uninstalling a Streamer Node	39
12.5.1	Step 1: Stop and disable service	39
12.5.2	Step 2: Remove application files	39

12.5.3 Step 3: Remove NFS mount	39
12.5.4 Step 4: Remove nginx configuration	39
12.5.5 Step 5: Remove streamer from admin backend	40
12.6 Quick Reference: Uninstall Commands	40

[13 Appendix](#) 41

13.1 Directory Structure	41
13.2 Database Tables (v3.1)	41
13.3 OBS Configuration	42
13.4 Stream Relay Examples	42
13.5 Multilingual Support	42
13.6 Maintenance Scripts	43

1 System Overview

Stream Suite is a professional, multi-tenant live streaming and video-on-demand platform. Multiple customers (tenants) can simultaneously broadcast their own live streams via RTMP or SRT, manage recordings, and automatically relay streams to external platforms such as YouTube, Twitch, or Facebook.

The platform is fully self-hosted and designed for professional use on Ubuntu servers. All components run as systemd services and are kept up to date through a central update script.

1.1 Architecture

Component	Function	Technology
RTMP/SRT Ingest	Receiving incoming streams from encoders (OBS, hardware)	MediaMTX v1.19+
Transcoding Farm	Distributed ABR encoding into 1–5 quality levels on worker nodes	FFmpeg (Worker)
Streaming Farm	Scalable HLS delivery across multiple streamer nodes	nginx + NFS
VOD Recording	Recording the raw stream independently from transcoding	FFmpeg stream copy
Stream Relay	Simultaneous RTMP/SRT push to external platforms without re-encoding	FFmpeg stream copy
HLS Delivery	Delivering live and VOD content as HTTP Live Streaming	nginx
Backend API	REST API, webhook handlers, session management, viewer tracking	Node.js / Express
Database	Configuration, statistics, farm status, all settings	SQLite (WAL mode)
Admin Frontend	Super administration interface for system administrators	HTML5 / Vanilla JS
Portal	Self-service interface for tenants and their users	HTML5 / Vanilla JS
Multilingual	UI translation into 9 languages with RTL support for Arabic	i18n.js / i18n-ui.js
NFS	Shared stream directory between master, workers, and streamers	NFSv3

1.2 Farm Architecture

1.2.1 Transcoding Farm

The transcoding farm distributes compute-intensive FFmpeg transcoding jobs across multiple worker servers. This offloads the master server and enables parallel streaming of many channels.

Component	Description
Master Server	Receives RTMP/SRT streams, dispatches transcoding jobs, manages VOD recording locally
Worker Node	Runs FFmpeg transcoding and writes HLS segments via NFS to the master storage
worker-agent.js	HTTP service on each worker (port 3100): receives jobs, reports status, enables remote restart
Dispatcher	Automatically selects the best worker based on available slots and current CPU load
NFS Mount	Workers mount the master stream directory and write HLS segments directly to it

1.2.2 Streaming Farm (optional)

The streaming farm scales HLS delivery across multiple dedicated streamer servers. Without the farm, the master delivers directly — which is fully sufficient for most deployments.

Component	Description
Streamer Node	Dedicated nginx server delivering HLS segments from the NFS mount to viewers
streamer-agent.js	Reports CPU, RAM, bandwidth, and active connections to the master every 10 seconds
Load Balancer	Selects the streamer with the best score based on configurable weights
Master (Standalone)	Delivers HLS directly when no streaming farm is configured

1.3 Job Lifecycle

Status	Meaning
dispatched	Job sent to worker, awaiting confirmation
running	Worker confirmed the job, FFmpeg process is running
stopping	Stop signal sent, FFmpeg is shutting down cleanly
stopped	Job finished cleanly, VOD is being finalized
failed	Job ended with error or worker went offline — VOD is finalized regardless

1.4 ABR Quality Levels

Stream Suite supports up to five quality levels. The player automatically selects the appropriate level based on viewer bandwidth. All levels can be individually enabled and are fully configurable.

Level	Resolution	Video Bitrate	Audio Bitrate	Default
Ultra (optional)	3840 × 2160	12,000 kbit/s	320 kbit/s	disabled
High	1920 × 1080	4,000 kbit/s	192 kbit/s	active
Medium	1280 × 720	2,000 kbit/s	128 kbit/s	active
Low	640 × 360	600 kbit/s	96 kbit/s	active
Extra (optional)	426 × 240	300 kbit/s	64 kbit/s	disabled

! All quality levels, bitrates, and encoder settings are configurable in Admin → Settings → Parameters — no code changes required.

1.5 URL Overview

Area	URL	Access
Super Admin	https://SERVER/admin	Username + password
Tenant Portal	https://SERVER/portal	Tenant login
Normal Player	https://SERVER/live/SLUG	Public — main viewer interface
TV Player	https://SERVER/tv/SLUG	Public — optimized for Smart TVs
Grid Player	https://SERVER/grid/SLUG	Public — all streams in one view
Embed Player	https://SERVER/embed/SLUG/SOURCE-ID	iFrame embedding in external websites
RTMP Ingest	rtmp://SERVER:1935/live/STREAM-KEY	Stream key from admin or portal
SRT Ingest	srt://SERVER:PORT	Port per source starting at 9000, OBS: Caller mode

2 Installation (Fresh Install)

Installation is fully automated via a single shell script. It sets up all components: MediaMTX as the RTMP/SRT ingest server, nginx as the HLS delivery server, Node.js backend, SQLite database, NFS server, and all systemd services.

2.1 System Requirements

2.1.1 Master Server

Requirement	Minimum	Recommended
Operating System	Ubuntu 22.04 LTS	Ubuntu 24.04 LTS
CPU	4 cores	8+ cores
RAM	4 GB	8+ GB
Storage	50 GB	500+ GB (for recordings)
Network	100 Mbit/s	1 Gbit/s
GPU (optional)	—	NVIDIA, Intel, or AMD

2.1.2 Worker Node (Transcoding Farm)

Requirement	Minimum	Recommended
Operating System	Ubuntu 24.04 LTS	Ubuntu 24.04 LTS
CPU	4 cores	8+ cores (1 core per High rendition)
RAM	2 GB	4+ GB
Network	1 Gbit/s (NFS)	10 Gbit/s
GPU (optional)	—	NVIDIA NVENC recommended

2.1.3 Streamer Node (Streaming Farm, optional)

Requirement	Minimum	Recommended
Operating System	Ubuntu 24.04 LTS	Ubuntu 24.04 LTS
CPU	2 cores	4 cores
RAM	1 GB	2 GB
Network	1 Gbit/s	10 Gbit/s (high outgoing HLS traffic)
NFS Mount	Required	/home/Stream Suite/streams from master

2.2 Installing the Master Server

Transfer the ZIP package to the server and extract it:

```
scp Stream Suite_v3_farm.zip user@SERVER:/tmp/  
cd /tmp && unzip Stream Suite_v3_farm.zip
```

Run the installer:

```
sudo bash /tmp/sf_out/scripts/install.sh
```

⚠ Default password: admin / admin123 — change it immediately after first login!

ℹ The installer sets up: MediaMTX, nginx, Node.js, SQLite, NFS server, and all systemd services.

2.3 Setting Up SSL/HTTPS (Recommended)

SSL/HTTPS is strongly recommended. Setup requires a publicly reachable domain (not an IP address).

```
sudo bash /opt/Stream Suite/scripts/install.sh --ssl
```

ℹ The update.sh script automatically detects and repairs faulty Certbot configurations on every update.

⚠ SSL requires a real domain — not an IP address. Certbot must be able to reach the server on port 80.

2.4 Adding a Worker Node

Set up an SSH key for farm communication:

```
ssh-keygen -t ed25519 -f ~/.ssh/Stream Suite_farm -N ""  
ssh-copy-id -i ~/.ssh/Stream Suite_farm.pub root@WORKER-IP
```

Run the worker node script from the master:

```
sudo bash /opt/Stream Suite/scripts/add-worker.sh
```

ℹ The script installs on the worker: Node.js, worker-agent.js, NFS client, and systemd service. The worker then appears automatically in Admin → Farm.

Explanation

In Stream Suite, the SSH key is required for **a single purpose**:

Initial setup of worker nodes—when `add-worker.sh` adds a new worker for the first time, it must access the worker via SSH to:

- copy `worker-agent.js`
- set up the systemd service
- configure the NFS mount

After that, SSH is no longer needed. All subsequent operations take place via HTTP:

- Updates: `POST /update-agent` on port 3100
- Restart: `POST /restart` on port 3100
- Heartbeats and job status: all via port 3100

Installation directly on the worker node

The `install-worker.sh` script runs directly on the worker node—no SSH from the master is required:

Step 1: Copy the ZIP file to the worker (from the master or directly):

```
bash
scp user@192.168.10.51:/tmp/Stream Suite_v3_farm.zip /tmp/
# or download directly on the worker
```

Step 2: Execute on the worker:

```
bash
cd /tmp && unzip Stream Suite_v3_farm.zip
sudo bash Stream Suite_dist/scripts/install-worker.sh
```

The script prompts for:

- Master IP (192.168.10.51)
- Agent Secret (from Admin → Farm → Settings)
- Worker ID (e.g., worker-01)

2.5 Adding a Streamer Node (Streaming Farm)

```
sudo bash /opt/Stream Suite/scripts/install-streamer.sh
```

After installation, the streamer automatically registers with the master and appears in Admin → Streaming Farm with status "Pending". Approve it manually there.

Streamer nodes require an NFS mount of /home/Stream Suite/streams from the master.

2.6 Firewall Configuration

Port	Protocol	Purpose	Public?
80	TCP	HTTP and Let's Encrypt challenge	Yes
443	TCP	HTTPS (after SSL setup)	Yes
1935	TCP	RTMP — stream ingest	Yes
8890	UDP	SRT — stream ingest	Yes
9000–9099	UDP	SRT — one port per source	Yes

Port	Protocol	Purpose	Public?
3100	TCP	Worker agent API (internal only!)	No
2049	TCP	NFS between master/worker/streamer	No

```
sudo ufw allow 80/tcp && sudo ufw allow 443/tcp
sudo ufw allow 1935/tcp && sudo ufw allow 8890/udp
sudo ufw allow 9000:9099/udp
```

2.7 GPU Encoding (Optional)

Auto-detection order: NVIDIA (NVENC) → Intel (QSV/VAAPI) → AMD (VAAPI) → CPU fallback.

```
journalctl -u Stream Suite-worker -n 30 | grep -i gpu
```

For AMD/Intel VAAPI: www-data needs access to render devices:

```
sudo usermod -aG render,video www-data && sudo systemctl restart Stream Suite
```

2.8 Service Management

Command	Function
sudo systemctl status Stream Suite	Show backend status
sudo systemctl restart Stream Suite	Restart backend
sudo systemctl restart mediamtx	Restart RTMP/SRT ingest server
journalctl -u Stream Suite -f	Backend log in real time
journalctl -u Stream Suite-worker -f	Worker agent log in real time
journalctl -u Stream Suite-streamer -f	Streamer agent log in real time

3 Updating an Existing Installation

Updates are applied via update.sh. The script copies all changed files, restarts services, repairs configuration issues, and runs smoke tests.

3.1 Full Update

```
scp Stream Suite_v3_farm.zip user@SERVER:/tmp/  
cd /tmp && unzip -o Stream Suite_v3_farm.zip  
sudo bash /tmp/sf_out/scripts/update.sh
```

The script automatically:

- Copies backend, frontend, scripts, and i18n files
- Restarts services, reloads nginx and MediaMTX
- Repairs Certbot configuration errors automatically
- Updates worker and streamer nodes via HTTP (no SSH required)
- Runs smoke tests on all key endpoints (HTTP or HTTPS)

! Smoke tests automatically switch to HTTPS if an SSL certificate is present.

3.2 Selective Update

Changed Feature	Target File on Server
Admin Frontend	/opt/Stream Suite/frontend/admin/index.html
Portal Frontend	/opt/Stream Suite/frontend/portal/index.html
Admin/Portal Translations	/opt/Stream Suite/frontend/i18n-ui.js
Player Translations	/opt/Stream Suite/frontend/player/i18n.js
Normal Player	/opt/Stream Suite/frontend/player/index.html
TV Player	/opt/Stream Suite/frontend/player/tv.html
Grid Player	/opt/Stream Suite/frontend/player/grid.html
Admin API	/opt/Stream Suite/backend/routes/admin.js
Portal API	/opt/Stream Suite/backend/routes/tenant-portal.js
Transcoder	/opt/Stream Suite/backend/transcoder.js
RTMP Hooks	/opt/Stream Suite/backend/routes/nginx-hooks.js
Database Schema	/opt/Stream Suite/backend/database.js

```
sudo systemctl restart Stream Suite
```

3.3 Automatic Database Migrations

New database columns are added automatically via ALTER TABLE when the backend starts. No manual intervention required. Migrations are idempotent.

***i** New in v3.1: `tenants.language`, `tenants.player_theme`, `tenant_users.language`, `admins.language`.*

4 Transcoding Farm

The transcoding farm is optional. In standalone mode, the master transcodes locally with FFmpeg. The farm is useful for many parallel streams at high quality.

4.1 Enabling the Farm

In Admin → Farm → Settings:

Setting	Description	Default
Farm Mode	Enables distribution to worker nodes. Without farm, the master handles everything locally.	Disabled
Master RTMP URL	URL from which workers pull the stream. Normally points to the master.	rtmp://127.0.0.1/live
CPU Overload Threshold	Above this CPU percentage, a worker is excluded from new job assignments.	95%
VOD Recording Slot Cost	Slots consumed by local VOD recording on the master.	1
Agent Secret	Shared key for worker authentication. Must match on all nodes.	Auto-generated

4.2 Managing Workers

Button	Function
 Edit	Adjust name, host, port, maximum slots, and rendition costs
↻ Restart	Restart worker agent via HTTP — no SSH required
↻ Master Services	Restart backend and worker agent on the master
↻ nginx + MediaMTX	Restart RTMP ingest and nginx

4.3 Slot Tracking

Rendition	Slot Cost (Default)	Reason
High (1080p)	8 slots	Highest CPU load, multiple cores used
Medium (720p)	5 slots	Moderate CPU load
Low (360p)	3 slots	Low CPU load
Extra (240p)	1 slot	Very low CPU load
Ultra (4K)	12 slots	Highest load, GPU encoding recommended

Rendition	Slot Cost (Default)	Reason
VOD Copy (master)	1 slot	Minimal load (stream copy, no re-encoding)

Slot costs are individually configurable per worker.

4.4 VOD Recording and Reconnect Logic

Recording always runs on the master, independent of the transcoding worker. A worker failure does not affect recordings.

Process	Runs On	Task
Worker FFmpeg	Worker Node	Live transcoding into quality levels → HLS segments via NFS to master storage
Master FFmpeg (VOD Copy)	Master	Recording with -c:v copy directly from MediaMTX — minimal CPU load

4.4.1 Automatic Reconnect (30-Second Window)

Stream Suite accepts brief stream interruptions and seamlessly continues writing to the same recording:

- Stream stops → recording finalization is delayed by 35 seconds
- Reconnect within 30 seconds → finalization timer is cancelled
- FFmpeg resumes recording into the same VOD file
- After 35 seconds without reconnect → recording is finalized and immediately playable

Even already-finalized VODs can be reactivated within the 30-second window.

4.5 Worker Failure

- Worker status is set to "offline" and removed from the dispatcher pool
- Running jobs are marked as "failed", occupied slots are released
- Affected VODs are finalized — existing segments remain intact and playable
- On the next stream, the dispatcher automatically selects an available worker

VOD recording on the master is not affected by worker failures.

5 Streaming Farm

The optional streaming farm distributes HLS delivery across multiple streamer nodes. For most installations, the master delivering directly is fully sufficient.

5.1 Streamer Pool Management

Area	Description
Streamer Cards	Real-time metrics: CPU%, RAM%, outgoing bandwidth, active connections, heartbeat status
Approve	Newly registered streamers must be manually approved before receiving traffic
Disable/Enable	Temporarily remove streamer from pool without restarting the service
Maintenance Mode	No new traffic, existing connections continue — ideal for updates
Settings	Configure max connections and max bandwidth as hard limits per streamer

5.2 Load Balancing

The load balancer calculates a score for each streamer and selects the one with the lowest load:

Parameter	Weight (Default)	Description
Bandwidth (w_bw)	0.4	Outgoing bandwidth relative to configured maximum
Connections (w_conn)	0.3	Active nginx connections relative to maximum
CPU (w_cpu)	0.2	CPU load of the streamer node
RAM (w_ram)	0.1	RAM usage of the streamer node

! Weights are configurable. Sum must equal 1.0.

5.3 Heartbeat Monitoring

Each streamer agent sends a heartbeat to the master every 10 seconds containing:

- CPU load (measured over a 5-second interval)
- RAM usage in megabytes
- Outgoing bandwidth in Mbit/s (from /proc/net/dev)
- Active nginx connections (from nginx stub_status)

! If a heartbeat is missing for more than 30 seconds, the streamer is marked offline and removed from the pool.

6 Super Administration

The admin backend is the central management interface for system administrators. Tenants, users, farm configuration, and all global settings are managed here.

6.1 First Login

URL: <https://SERVER/admin> — Default credentials: admin / admin123

⚠ Change the default password immediately under Settings → Account Settings → Change Password!

6.2 Display Language

Each admin user can set their UI language independently:

- Topbar: language dropdown to the right of "Sign out" — takes effect immediately
- Settings → Account Settings → Display Language

ℹ Language is stored per admin user and loaded automatically at each login. Currently available: German, English.

6.3 Dashboard

- Active live streams with viewer counts and view chart (last 14 days)
- Server load: CPU, RAM, disk (VOD), network ↑↓, FFmpeg processes
- Farm mode: worker status, slot usage, active transcoding jobs
- Standalone mode: local FFmpeg processes split by recording and transcoding

ℹ In farm mode, the network display sums the bandwidth of all active streamer nodes.

6.4 Managing Tenants

Field	Description
Name	Full company name (only visible internally in the admin interface)
Slug	URL identifier for all player URLs (lowercase and hyphens, e.g. "my-customer")
Player Title	Display name in browser tab and all public player interfaces

Field	Description
Accent Color	Primary color for all players of this tenant (hex code or color picker)
Player Theme	Dark (dark background, default) or Light (light background)
Language	Player language for viewers: DE, EN, FR, ES, NL, PT, AR, TR, RU
Logo	PNG or JPG, max 5 MB — appears in all player frontends as branding

! Arabic automatically enables RTL (right-to-left) layout in all players.

6.5 Managing Users

Field	Description
Username	Login name for the portal (must be unique system-wide)
Display Name	Display name within the portal interface
Password	At least 6 characters. Leave empty when editing to keep current password.
Active	Deactivated users cannot log in but remain in the database
Language	UI language for this portal user — independent of the tenant's player language

6.6 Live Sources

Setting	Description
Name	Display name of the source in admin, portal, and player tabs
Protocol	RTMP (default, port 1935) or SRT (dedicated UDP port starting at 9000)
Stream Key	Automatically generated key for RTMP connections
SRT Port	Dedicated UDP port for this SRT source (e.g. 9000, 9001 ...)
Recording	Enable or disable VOD recording for this source
Transcode Config	Per-source override of the global quality level settings

6.7 Stream Control

State	Available Actions
Offline	▶ Start Stream (enable transcoding) / ● Record Only (without transcoding)
Live, no recording	■ Stop Stream / ● Start Recording
Live + recording active	■ Stop Stream / ■ Stop Recording (both fully independent)

6.8 Stream Relays

Any number of relays can be configured per source. FFmpeg stream copy — minimal CPU load, no quality loss.

Status	Meaning
● Active (green)	FFmpeg relay running, stream is being forwarded
⌛ Connecting (yellow)	Connection error — automatic restart (max. 10 attempts)
○ Inactive (grey)	Manually stopped
✗ Error (red)	10 restart attempts exhausted — manual start required

Relays start automatically 2s after the main stream and stop when the stream ends.

6.9 Recordings

Action	Description
 Edit	Adjust title and description of the recording
 Play	Play in the built-in browser VOD player
 Stop	Immediately stop and finalize a running recording
 Download	Download as raw .ts file
 Repair	Rebuild index.m3u8 playlist from existing .ts segments
 Visibility	Hide or show in portal and public player
Checkbox	Select multiple recordings and delete them together

6.10 Tenant Export / Import

6.10.1 Export

- Admin → Tenants →  Export: metadata as ZIP (sources, VODs, users, settings)
-  + : additionally includes statistics data
- Video files (.ts segments) are not exported — metadata only

6.10.2 Import

- Admin → Tenants →  Import: upload ZIP file (max. 100 MB)
- v2 formats are automatically converted to v3

Transfer video files separately after import:

```
rsync -avz -e "ssh -i ~/.ssh/Stream Suite_farm" \
/home/Stream Suite/streams/SLUG/ root@TARGET:/home/Stream Suite/streams/SLUG/
```

⚠ Fix permissions after rsync: `sudo chown -R www-data:www-data /home/Stream Suite/streams/`

6.11 Transcoding Parameters

All encoding settings can be configured under Admin → Parameters. Changes take effect the next time the stream starts.

6.11.1 Encoder & global settings

Parameter	DB-Key	Standard	Description
GPU-Encoder	transcode.gpu_encoder	auto	auto nvenc qsv vaapi cpu
CPU/QSV Preset	transcode.preset	veryfast	ultrafast / superfast / veryfast / faster / fast / medium
NVENC Preset	transcode.nvenc_preset	p4	p1 (fast) ... p7 (best quality) — only NVIDIA
Framerate	transcode.framerate	0	0 = apply source framerate
GOP Size	transcode.gop_size	60	Keyframe distance in frames (60 = 2s at 30fps)
HLS Segment Length	transcode.hls_segment_time	4	Seconds per .ts-Segment
HLS List Size	transcode.hls_list_size	10	Nubmer of segmente in playlist (0 = all)
Audio Channels	transcode.audio_channels	2	1 = mono, 2 = stereo
Audio Sample Rate	transcode.audio_samplerate	48000	44100 or 48000 Hz

6.11.2 Quality levels

Level	Active	Resolution	Video bitrate	Audio bitrate
☒ Ultra	No	3840x2160	12000k	320k
☒ High	Yes	1920x1080	4000k	192k
☒ Medium	Yes	1280x720	2000k	128k
☒ Low	Yes	640x360	600k	96k
☐ Extra	No	426x240	300k	64k

ⓘ Bitrates in NNNk format (e.g., 4000k). Resolutions in WIDTHxHEIGHT format (e.g., 1920x1080). At least one level must be active.

6.11.3 Source specific overwriting

At the "Edit"-Modal of a source global parameters can be overwritten. Not overwritten parameters are falling back to the global settings.

6.12 Statistics

Tab	Content
Overview	Total/live/VOD views, unique viewers, active viewers, view chart (14 days)
Geography	World map with city coordinates, top cities and countries by view count
Devices & Browsers	Donut charts for device categories and browser distribution
Engagement	Unique viewers, average playback duration, average plays per day

! Heartbeat every 30s. Final heartbeat via sendBeacon on browser close — even short sessions are fully recorded.

7 Troubleshooting

7.1 502 Bad Gateway

Cause	Diagnosis	Solution
Service stopped	<code>sudo systemctl status Stream Suite</code>	<code>sudo systemctl restart Stream Suite</code>
Syntax error in code	<code>node --check /opt/Stream Suite/backend/server.js</code>	Fix the error line, restart
Port 3000 in use	<code>sudo ss -tlnp grep 3000</code>	Stop the conflicting application

7.2 RTMP Stream Active, Player Shows Nothing

- Is FFmpeg running? → `ps aux | grep ffmpeg`
- HLS segments present? → `ls -la /home/Stream Suite/streams/SLUG/live/`
- Is MediaMTX receiving? → `journalctl -u mediamtx -f`
- Is worker accepting jobs? → `journalctl -u Stream Suite-worker -f`

! In standalone mode, FFmpeg runs directly on the master. In farm mode, a worker must be active.

7.3 SRT Stream: Connection Error

Check	Solution
Listener running?	<code>ss -ulnp grep PORT</code>
Firewall open?	<code>sudo ufw allow PORT/udp (UDP, not TCP!)</code>
OBS configured correctly?	Connection type: Caller, no stream key for SRT
Port configured?	Admin → Tenants → Source → check SRT port

7.4 VOD Not Playable

Cause: empty or missing index.m3u8. Solution: Admin → Recording →  Repair.

Manual repair of a single recording:

```
VOD_DIR="/home/Stream Suite/streams/SLUG/vods/UUID"
{ echo "#EXTM3U"; echo "#EXT-X-VERSION:3"; echo "#EXT-X-PLAYLIST-TYPE:VOD"
  for f in $(ls $VOD_DIR/seg_*.ts | sort); do
    echo "#EXTINF:9.000,"; echo "${basename $f}"; done
  echo "#EXT-X-ENDLIST"; } > $VOD_DIR/index.m3u8
```

7.5 Stream Relay does not start

Check	Solution
Logs?	journalctl -u Stream Suite grep Output
Target-URL correct?	Checking rtmp://SERVER/PFAD and Stream-Key
Stream runs?	Relays starts only 2s after Stream-Start
Max Retries?	After 10 fail attempts Auto-Restart stops – start manuell

7.6 Farm Jobs hang

```
sudo sqlite3 /var/lib/Stream Suite/Stream Suite.db \
  "UPDATE farm_jobs SET status='failed', ended_at=datetime('now') WHERE status IN
  ('running','dispatched','stopping');"

```

⚠ Only run when no streams are active!

7.7 Thumbnails are missing

```
sudo bash /opt/Stream Suite/scripts/regen_thumbnails.sh --missing
sudo bash /opt/Stream Suite/scripts/regen_thumbnails.sh --slug SLUG

```

7.8 Diagnosis Overview

Problem	First Check	Solution
502 Bad Gateway	journalctl -u Stream Suite	Service restart
RTMP no HLS	ps aux grep ffmpeg	Checking Worker-Agent and Farm-Jobs
SRT does not connect	ss -ulnp grep PORT	Share UDP-Port
VOD Failure	Admin →  Button	M3U8 will be recreated
Relay stoped	journalctl grep Output	Checking Target URL, manuell restart
GPU not recognized	journalctl grep GPU	Parameter Page → GPU-Encoder
Farm-Job hangs	Admin → Farm → Worker-Status	Cleaning DB, restart Worker

7.9 Other Issues

Problem	Solution
Farm jobs stuck (no active stream)	Admin → Farm → Jobs. DB: UPDATE farm_jobs SET status='failed' WHERE status IN ('running','dispatched')
Thumbnails missing or black	sudo bash /opt/Stream Suite/scripts/regen_thumbnails.sh
GPU not detected	Admin → Parameters → set GPU encoder manually
Streamer stays offline	Check systemctl status Stream Suite-streamer on the streamer node
SSL port 80 returns 404	sudo bash /opt/Stream Suite/scripts/update.sh — repairs automatically
i18n texts not translated	Browser F12 → Network: /i18n-ui.js must return HTTP 200

8 WebRTC / WHIP Ingest

StreamForge supports WebRTC as an ingest protocol in addition to RTMP and SRT. WebRTC enables direct video and audio transmission from browsers and apps without additional software. WHIP (WebRTC-HTTP Ingestion Protocol) is used as the ingest protocol.

8.1 Architecture

WebRTC ingest uses the following components:

Component	Function
Browser / App	Sends video (H.264) and audio (Opus) via WHIP to the server
nginx	Proxy: HTTPS → MediaMTX port 8889
MediaMTX	Receives WebRTC/WHIP, forwards stream internally
StreamForge Backend	Detects Opus audio, sends RTSP URL to worker
Worker Agent (FFmpeg)	Reads RTSP, converts Opus→AAC, creates HLS
HLS Delivery	Player receives HLS just like RTMP/SRT sources

With RTMP/SRT sources, the worker agent reads the stream directly via RTMP. With WebRTC/WHIP sources, RTSP is used because RTMP cannot carry Opus audio. The Opus→AAC conversion happens automatically on the worker.

8.2 Requirements

8.2.1 Server

Port	Protocol	Purpose
443/tcp	HTTPS	WHIP proxy via nginx (e.g. /whip/live/KEY/whip)
8889/tcp	HTTP	MediaMTX WebRTC direct (without nginx)
8888/udp	UDP	WebRTC ICE media
8887/tcp	TCP	WebRTC ICE (fallback)
8554/tcp	RTSP	Internal RTSP for worker agent (VLAN access)

Ports 8888 and 8887 are required for ICE negotiation between client and server. Without these ports, the WebRTC connection will fail.

8.2.2 Encoder Requirements (H.264)

WebRTC senders must use H.264 with the Constrained Baseline profile. Higher profiles (Main, High) may only send SPS/PPS information with the first keyframe. Since the worker agent reads the stream via RTSP from the server (not from the beginning), this information may be missing and the stream cannot be decoded.

⚠ *H.264 profile must be Baseline (Constrained Baseline). Main and High profiles do not work reliably with WebRTC/WHIP ingest.*

8.3 Creating Sources

WebRTC/WHIP sources are created in StreamForge like regular live sources — there is no separate source type for WebRTC. Each source has a stream key that is equally valid for RTMP, SRT, and WebRTC/WHIP.

The WHIP URL is automatically generated from the stream key and displayed in the admin backend and tenant portal:

```
https://SERVER/whip/live/STREAM-KEY/whip
```

Protocol	Connection URL
RTMP	rtmp://SERVER/live/STREAM-KEY
SRT	srt://SERVER:PORT (source-specific port)
WebRTC/WHIP	https://SERVER/whip/live/STREAM-KEY/whip

8.4 OBS Studio with WHIP Plugin

8.4.1 Setup

Setting	Value
Service	WHIP
Server	https://SERVER/whip/live/STREAM-KEY/whip
Bearer Token	(leave empty)

8.4.2 Encoder Settings

Under Settings → Output → Streaming → Encoder: select x264. Enter the following in the "Custom Encoder Settings" field:

```
profile=baseline tune=zerolatency keyint=50 min-keyint=50
```

Parameter	Description
profile=baseline	H.264 Baseline profile — SPS/PPS in every keyframe
tune=zerolatency	Minimal encoding buffer for live streaming

Parameter	Description
keyint=50	Keyframe every 50 frames (at 25fps = 2 seconds)
min-keyint=50	Minimum keyframe interval — prevents too frequent keyframes

⚠ Main and High profiles do not work with WHIP. Without these settings, transcoding will not start.

8.5 Browser-Based WHIP Client

StreamForge includes a built-in WebRTC/WHIP client that runs in the browser. It is suitable for testing and simple broadcasts without additional software.

8.5.1 Access

Context	URL
Admin (all sources)	https://SERVER/webrtc-test
Admin (tenant-filtered)	Via  WebRTC button in the tenant row
Tenant portal	Via  WebRTC Test button in the sources view
Source pre-selected	Via  button in the sources table

8.5.2 Features

Feature	Description
Source selection	Automatically loads available WHIP sources from the session
Camera/microphone selection	All connected devices selectable
Resolution selection	Full HD, HD, SD, Low
H.264 Constrained Baseline	Enforced automatically — no manual configuration required
Server status	Shows whether the server is receiving the stream and writing segments
Auto-restart	Automatically restarts the connection if the server goes offline
ICE connection log	Shows connection setup and potential issues

ⓘ The browser client requires HTTPS. On HTTP connections, camera/microphone access is not available (browser security policy). The WHIP URL can be entered manually in this case.

8.6 Stability and Reconnect

WebRTC is optimized for peer-to-peer connections and may be less stable than RTMP or SRT for server ingest. StreamForge includes several mechanisms to compensate for connection interruptions:

Mechanism	Function
ICE Grace Period (5s)	Short connection interruptions are ignored — no immediate stream stop
Reconnect window (30s)	VOD recording continues if reconnected within 30 seconds
Auto-restart (client)	Browser client automatically restarts WHIP connection if server is offline
Poller timeout	Configurable — recommended for WebRTC: 45–60 seconds
omit_endlist	HLS playlist is not marked as ended — player buffers instead of stopping

The poller timeout can be configured under Admin → Transcoding → Parameters → transcode.stale_timeout_sec. The default value of 20 seconds is sufficient for RTMP; for WebRTC, 45–60 seconds is recommended.

8.7 Troubleshooting

Problem	Cause	Solution
WHIP 404	nginx proxy missing or stream not active	Check nginx WHIP location; source must exist in StreamForge
PeerConnection Failed	ICE ports (8888/8887) not reachable	Check UFW rules: ufw allow 8888/udp and 8887/tcp
No audio	Opus audio — worker could not access RTSP	Check port 8554 UFW rule for worker IP
No video / decode error	H.264 not Baseline profile	OBS: add profile=baseline to encoder settings
Stream dies after ~40s	Poller timeout too short	Set transcode.stale_timeout_sec to 60
Old content in player	Browser cache	Ctrl+Shift+R (hard reload) in browser

9 Tenant Portal

The portal is the self-service interface for customers. Portal users can control streams, manage recordings, configure relays, and customize the player design — without access to the system administration.

9.1 Login and Display Language

URL: `https://SERVER/portal` — username and password (assigned by the administrator).

Set language: Account → User Settings → Interface Language — takes effect immediately.

! The portal user's UI language is independent of the tenant's player language.

9.2 OBS Configuration

Protocol	URL	OBS Setting
RTMP	<code>rtmp://SERVER:1935/live/STREAM-KEY</code>	Service: Custom — enter server and stream key separately
SRT	<code>srt://SERVER:PORT</code>	Service: Custom — enter full URL, leave stream key field empty

! For SRT: set OBS connection type to "Caller". Default latency: 200ms.

9.3 Stream Control

State	Available Actions
Offline	▶ Start Stream (enable transcoding) / ● Record Only (without transcoding)
Live, no recording	■ Stop Stream / ● Start Recording
Live + recording active	■ Stop Stream / ■ Stop Recording (both fully independent)

9.4 Stream Relays

Any number of relays can be configured per source. FFmpeg stream copy — minimal CPU load, no quality loss.

Status	Meaning
● Active (green)	FFmpeg relay running, stream is being forwarded
⌚ Connecting (yellow)	Connection error — automatic restart (max. 10 attempts)

Status	Meaning
○ Inactive (grey)	Manually stopped
✗ Error (red)	10 restart attempts exhausted — manual start required

Relays start automatically 2s after the main stream and stop when the stream ends.

9.5 Recordings

Action	Description
 Edit	Adjust title and description of the recording
 Play	Play in the built-in browser VOD player
 Stop	Immediately stop and finalize a running recording
 Download	Download as raw .ts file
 Repair	Rebuild index.m3u8 playlist from existing .ts segments
 Visibility	Hide or show in portal and public player
Checkbox	Select multiple recordings and delete them together

9.6 Account Settings

Setting	Description
Interface Language	UI language for this user — takes effect immediately, saved permanently
Player Title	Display name in browser tab and all player interfaces
Player Theme	 Dark (default) or  Light
Accent Color	Primary color for all players (color picker or hex code)
Logo	PNG or JPG up to 5 MB — appears in all players as branding
Language (Channel)	Player language for viewers: DE, EN, FR, ES, NL, PT, AR, TR, RU
Change Password	Current password + new password (minimum 6 characters)

10 Viewer Frontends

Stream Suite provides four different player frontends. All automatically load language and design from the tenant settings.

10.1 Normal Player (/live/SLUG)

Main player for desktop and mobile browsers. Video.js 8 with VHS plugin (Videojs HTTP Streaming) for maximum compatibility and reliable ABR playback.

Feature	Description
ABR	Automatic bitrate adaptation via Video.js qualityLevels API
Quality Switcher	Auto + all active levels manually selectable
Multi-Source	Tabs to switch between multiple live sources of a tenant
VOD Gallery	Recordings accessible directly below the live player
Multilingual	9 languages, RTL for Arabic, loaded from tenant settings
Stall Detection	Automatic player restart after 15s without progress
Reconnect	Automatic reconnect attempt on stream interruption
Tracking	Heartbeat every 30s, geo-lookup, device detection

10.2 TV Player (/tv/SLUG)

Optimized for Smart TVs (Samsung Tizen 2015+, LG webOS) and set-top boxes. ES5-compatible code, NodeList.forEach polyfill. Player is reused on source change instead of re-initialized.

Device	Support
Samsung Tizen 2015+	✓ Native HLS — full compatibility
LG webOS	✓ Native HLS — full compatibility
Android TV / Fire TV	✓ Video.js VHS — full compatibility
Samsung Orsay 2012–2014	⚠ Usually compatible
Samsung / others before 2012	✗ No HLS support

10.3 Grid Player (/grid/SLUG)

Shows all live sources of a tenant simultaneously in a 3×3 grid view. Ideal for monitoring screens. Uses HLS.js for optimal multi-stream performance.

Feature	Description
Layout	3×3 grid, responsive — 2×2 on tablets, 1×1 on mobile
Audio	All streams muted — click  to enable sound for one cell
Volume	Saved volume is restored after page changes (paging)
Paging	With more than 9 sources: automatic paging with 9 streams per page
Auto Update	Status polling every 15s — new streams appear automatically
Error Handling	Automatic retry after HLS fatal error (5s delay)

10.4 Embed Player (/embed/SLUG/SOURCE-ID)

iFrame-embeddable player for external websites. Each live source has its own embed URL.

Responsive embed code:

```
<div style="position:relative;padding-bottom:56.25%;height:0">
  <iframe src="https://SERVER/embed/SLUG/SOURCE-ID"
    style="position:absolute;top:0;left:0;width:100%;height:100%;border:none"
    allowfullscreen allow="autoplay; fullscreen"></iframe>
</div>
```

Find Source-ID: Admin → Tenants → Sources, or GET `/api/tenant/SLUG/live`

10.4.1 Embed-Code (fix size)

```
<iframe src="https://SERVER/embed/SLUG/SOURCE-ID"
  width="640" height="360" style="border:none;border-radius:8px"
  allowfullscreen allow="autoplay; fullscreen"></iframe>
```

Properties	Description
Autoplay	Trys starting automatically; Click overlay when blocked
Quality	ABR switch
Full Frame	Button at menu bar
Branding	Logo + Name of the tenant at the bottom right
Status Polling	Every 15s: starts automatically when stream goes live

 Demo-Site with code generator: <http://SERVER/embed-demo>

11 Public REST API

Public endpoints without authentication for integration into external websites or applications. Base URL: <https://SERVER/api>

11.1 GET /api/tenant/:slug — Tenant Info

Returns design, language, and logo of a tenant. Loaded at player startup.

```
→ { "slug":"my-tenant", "title":"My Tenant",  
    "primary_color":"#5793c9", "player_theme":"light",  
    "language":"en", "logo_path":"/uploads/..." }
```

11.2 GET /api/tenant/:slug/live — Live Sources

All live sources with current status. Player polls every 20s (3s when offline).

```
→ [ { "id":"uuid", "name":"Main Camera", "status":"live",  
      "hls_url":"/streams/my-tenant/live/uuid/index.m3u8" } ]
```

! Possible status values: "live" | "offline" | "recording"

11.3 GET /api/tenant/:slug/vods — Recordings

All public, finalized recordings (newest first). Hidden recordings are not returned.

```
→ [ { "id":"uuid", "title":"Service - 14.7.2026", "duration":5400,  
      "hls_url":"/streams/.../index.m3u8",  
      "thumbnail_path":"/streams/.../thumbnail.jpg" } ]
```

11.4 POST /api/track — Start Tracking Session

Starts a viewer tracking session. Returns `stat_id` for heartbeat updates.

```
Body (Live): { "tenant_id":"uuid", "type":"live", "source_id":"uuid" }  
Body (VOD): { "tenant_id":"uuid", "type":"vod", "vod_id":"uuid" }  
→ { "success":true, "stat_id":42 }
```

11.4.1 POST /api/track/update — Heartbeat

Updates playback duration. Sent every 30s, on page close via `sendBeacon`.

```
Body: { "stat_id":42, "play_duration":120, "error_count":0 }
```

11.5 GET /api/streamer-redirect — Streamer Failover

In farm mode, returns the best available streamer. In standalone mode: standalone:true.

```
→ Farm:      { "reload":true, "streamer_url":"...", "stream_url":"..." }  
→ Standalone: { "reload":false, "capacity_exceeded":false, "standalone":true }  
→ Exhausted: { "reload":false, "capacity_exceeded":true }
```

12 Uninstallation

This chapter describes how to completely or partially remove Stream Suite from a server. Uninstallation follows this order: stop services → remove files → clean up configuration → optional data backup.

⚠ Back up all important data before uninstalling. Recordings, the database, and configuration are permanently deleted during a full uninstallation.

12.1 Overview: What Gets Installed Where

Stream Suite places files in several locations. The following table shows what is removed during uninstallation:

Path	Content	Uninstall
/opt/Stream Suite/	Application code, scripts, configuration	Yes
/var/lib/Stream Suite/	Database (Stream Suite.db), uploads	Yes — backup recommended
/home/Stream Suite/streams/	HLS segments, VOD recordings	Yes — backup recommended
/etc/systemd/system/Stream Suite*.service	systemd service files	Yes
/etc/mediamtx/mediamtx.yml	MediaMTX configuration	Yes
/etc/nginx/nginx.conf	nginx configuration	Partially — restore original
/etc/sudoers.d/Stream Suite-services	sudo permissions for www-data	Yes
/etc/exports	NFS export entry	Partially — remove line
/etc/nfs.conf	mountd port configuration	Partially — remove section
/usr/local/bin/mediamtx	MediaMTX binary	Yes

12.2 Data Backup Before Uninstallation

Back up the following data before uninstalling:

12.2.1 Back up the database

```
cp /var/lib/Stream Suite/Stream Suite.db ~/Stream Suite_backup_$(date +%Y%m%d).db
```

12.2.2 Back up recordings

Recordings may amount to several hundred gigabytes depending on storage volume. Back up to external storage:

```
rsync -avz /home/Stream Suite/streams/ /mnt/backup/Stream Suite-streams/
```

i The database contains all tenants, users, settings, and VOD metadata. Video files are stored in the stream directory and are not included in the database.

12.3 Uninstalling the Master Server

The following steps fully remove all Stream Suite components from the master server.

12.3.1 Step 1: Stop and disable services

```
sudo systemctl stop Stream Suite Stream Suite-worker mediamtx nginx  
sudo systemctl disable Stream Suite Stream Suite-worker mediamtx
```

12.3.2 Step 2: Remove service files

```
sudo rm -f /etc/systemd/system/Stream Suite.service  
sudo rm -f /etc/systemd/system/Stream Suite-worker.service  
sudo rm -f /etc/systemd/system/mediamtx.service  
sudo systemctl daemon-reload
```

12.3.3 Step 3: Remove application files

```
sudo rm -rf /opt/Stream Suite
```

12.3.4 Step 4: Remove data (only if no backup is needed)

```
sudo rm -rf /var/lib/Stream Suite  
sudo rm -rf /home/Stream Suite/streams
```

⚠ These commands permanently delete all recordings, the database, and all uploads. Back up first!

12.3.5 Step 5: Remove MediaMTX

```
sudo rm -f /usr/local/bin/mediamtx  
sudo rm -rf /etc/mediamtx
```

12.3.6 Step 6: Clean up NFS configuration

Remove the Stream Suite NFS export entry:

```
sudo sed -i '/Stream Suite/d' /etc/exports
sudo exportfs -ra
```

Remove the mountd port configuration:

```
sudo sed -i '/^\[mountd\]/,/^port=32767/d' /etc/nfs.conf
sudo systemctl restart nfs-server
```

12.3.7 Step 7: Restore nginx configuration

Reset nginx to the default configuration:

```
sudo cp /etc/nginx/nginx.conf.bak.* /etc/nginx/nginx.conf
sudo nginx -t && sudo systemctl reload nginx
```

! If no backup exists: `sudo apt-get install --reinstall nginx` restores the default configuration.

12.3.8 Step 8: Remove sudo permissions

```
sudo rm -f /etc/sudoers.d/Stream Suite-services
```

12.3.9 Step 9: Remove optional packages

These packages were installed by Stream Suite and can be removed if no longer needed elsewhere:

```
sudo apt-get remove --purge nfs-kernel-server vainfo intel-media-va-driver
sudo apt-get autoremove
```

*! nginx, ffmpeg, and Node.js are not removed automatically as they may be used by other applications.
Remove manually if desired: `sudo apt-get remove --purge nginx ffmpeg nodejs`*

12.4 Uninstalling a Worker Node

The following steps remove the transcoding worker agent from a worker node.

12.4.1 Step 1: Stop and disable service

```
sudo systemctl stop Stream Suite-worker
sudo systemctl disable Stream Suite-worker
sudo rm -f /etc/systemd/system/Stream Suite-worker.service
sudo systemctl daemon-reload
```

12.4.2 Step 2: Remove application files

```
sudo rm -rf /opt/Stream Suite
```

12.4.3 Step 3: Remove NFS mount

Remove the NFS mount from `/etc/fstab` and unmount:

```
sudo sed -i '/Stream Suite/d' /etc/fstab
sudo umount /home/Stream Suite/streams 2>/dev/null || true
```

12.4.4 Step 4: Remove sudo permissions

```
sudo rm -f /etc/sudoers.d/Stream Suite-worker-restart
```

12.4.5 Step 5: Remove worker from admin backend

In the admin backend under Transcoding Farm, delete the worker entry. This removes the worker from the database and releases its allocated slots.

! Before uninstalling, ensure no active stream is dispatched to this worker. Temporarily disable farm mode or stop the stream.

12.5 Uninstalling a Streamer Node

The following steps remove the streaming farm agent from a streamer node.

12.5.1 Step 1: Stop and disable service

```
sudo systemctl stop Stream Suite-streamer nginx
sudo systemctl disable Stream Suite-streamer
sudo rm -f /etc/systemd/system/Stream Suite-streamer.service
sudo systemctl daemon-reload
```

12.5.2 Step 2: Remove application files

```
sudo rm -rf /opt/Stream Suite
```

12.5.3 Step 3: Remove NFS mount

```
sudo sed -i '/Stream Suite/d' /etc/fstab
sudo umount /home/Stream Suite/streams 2>/dev/null || true
```

12.5.4 Step 4: Remove nginx configuration

```
sudo rm -f /etc/nginx/sites-enabled/Stream Suite-streamer
sudo rm -f /etc/nginx/sites-available/Stream Suite-streamer
sudo systemctl stop nginx && sudo systemctl disable nginx
```

12.5.5 Step 5: Remove streamer from admin backend

In the admin backend under Streaming Farm, remove the streamer entry. This removes the streamer from the load balancer pool.

i The streamer can also be set to "Disabled" in the admin first to remove it from the pool without immediately deleting it. This allows existing connections to drain gracefully.

12.6 Quick Reference: Uninstall Commands

Summary of all uninstall commands for the master server:

```
# Stop services
sudo systemctl stop Stream Suite Stream Suite-worker mediamtx nginx
sudo systemctl disable Stream Suite Stream Suite-worker mediamtx

# Remove service files
sudo rm -f /etc/systemd/system/Stream Suite{-worker,-streamer}.service
sudo rm -f /etc/systemd/system/mediamtx.service
sudo systemctl daemon-reload

# Remove application
sudo rm -rf /opt/Stream Suite
sudo rm -f /usr/local/bin/mediamtx /etc/mediamtx
sudo rm -f /etc/sudoers.d/Stream Suite-services

# Remove data (permanent!)
sudo rm -rf /var/lib/Stream Suite /home/Stream Suite/streams

# Clean up NFS
sudo sed -i '/Stream Suite/d' /etc/exports && sudo exportfs -ra

# Restore nginx
sudo cp /etc/nginx/nginx.conf.bak.* /etc/nginx/nginx.conf
sudo nginx -t && sudo systemctl reload nginx
```

What remains after uninstallation	Note
nginx, ffmpeg, Node.js, nfs-kernel-server	Manual removal required if no longer needed
www-data user and group	System user — do not remove
UFW rules	Remove manually: <code>sudo ufw delete allow PORT</code>
Let's Encrypt certificates	<code>/etc/letsencrypt/</code> — remove manually if no longer needed

13 Appendix

13.1 Directory Structure

Path	Content
/opt/Stream Suite/	Application root directory
/opt/Stream Suite/backend/	Node.js backend (API, transcoder, dispatcher, agents)
/opt/Stream Suite/frontend/admin/	Super admin frontend (index.html)
/opt/Stream Suite/frontend/portal/	Tenant portal (index.html)
/opt/Stream Suite/frontend/i18n-ui.js	Translations for admin/portal (DE/EN)
/opt/Stream Suite/frontend/player/	Viewer frontends (index.html, tv.html, grid.html, embed.html)
/opt/Stream Suite/frontend/player/i18n.js	Player translations (9 languages)
/opt/Stream Suite/scripts/	Maintenance and deployment scripts
/var/lib/Stream Suite/Stream Suite.db	SQLite database
/home/Stream Suite/streams/	HLS segments (live + VODs) — shared via NFS
/etc/mediamtx/mediamtx.yml	MediaMTX configuration (RTMP/SRT)

13.2 Database Tables (v3.1)

Table	Content
admins	Super admin credentials and language setting
tenants	Tenants with name, slug, logo, color, theme, and player language
tenant_users	Portal users per tenant with individual UI language
live_sources	RTMP/SRT sources with transcode config
stream_outputs	Stream relays (RTMP/SRT push)
vods	Recordings with metadata and file paths
view_stats	Viewer tracking (IP, geo, device, browser, duration)
settings	Transcoding parameters and farm settings
farm_workers	Worker nodes (host, slots, rendition costs)
farm_jobs	Job history (status, worker, timestamps)
streamers	Streaming farm streamer nodes
streamer_heartbeats	Heartbeat history of streamer nodes

13.3 OBS Configuration

Protocol	OBS Field	Value
RTMP	Server	rtmp://SERVER:1935/live
RTMP	Stream Key	Stream key from Admin/Portal → Sources
SRT	Server	srt://SERVER:PORT (port from Admin/Portal)
SRT	Stream Key	Leave empty
WebRTC/WHIP	Server	https://SERVER/whip/live/STREAM-KEY/whip
WebRTC/WHIP	Stream Key	Leave empty

! SRT: set OBS connection type to "Caller". Default latency 200ms.

13.4 Stream Relay Examples

Service	Protocol	Server URL	Stream Key
YouTube Live	RTMP	rtmp://a.rtmp.youtube.com/live2	YouTube stream key (from YouTube Studio)
Facebook Live	RTMP	rtmp://live-api-s.facebook.com:80/rtmp/	Facebook primary key
Twitch	RTMP	rtmp://live.twitch.tv/app/	Twitch stream key (from Creator Dashboard)
SRT Target	SRT	srt://TARGETSERVER:PORT	Stream ID (optional)

13.5 Multilingual Support

Language	Code	Player	Admin/Portal
German	de	✓	✓
English	en	✓	✓
French	fr	✓	coming
Spanish	es	✓	coming
Dutch	nl	✓	coming
Portuguese	pt	✓	coming
Arabic (RTL)	ar	✓	coming
Turkish	tr	✓	coming
Russian	ru	✓	coming

! Player language configured per tenant in Admin → Tenants → Language. Admin/portal language is individually configurable per user.

13.6 Maintenance Scripts

Script	Function
update.sh	Full system update including worker/streamer nodes and smoke tests
add-worker.sh	Add a new transcoding worker node interactively
install-streamer.sh	Install a new streaming farm node
install.sh --ssl	Set up or renew SSL/HTTPS with Let's Encrypt
regen_thumbnails.sh	Regenerate thumbnails for all recordings
regen_thumbnails.sh --missing	Create only missing thumbnails
regen_thumbnails.sh --slug SLUG	Only for a specific tenant
cleanup_vods.sh --dry-run	Show orphaned VODs without deleting (test mode)
cleanup_vods.sh	Permanently delete orphaned VOD directories