FFmpeg + NGINX HTTPS HLS Streaming Server Ubuntu 24.04

In this tutorial, we build a lightweight streaming server on **Ubuntu 24.04** that takes **RTSP video feeds from IP cameras** and publishes them to the web using **HLS (HTTP Live Streaming)**.

We use **FFmpeg** to pull video streams from each camera, remove audio, segment the video into HLS chunks, and keep a rolling archive.
Then, **NGINX with HTTPS** serves those HLS playlists securely so they can be streamed in any modern browser — no plugins required.

This setup is perfect when you want:

- Live CCTV viewing over HTTPS
- Minimal resource usage (video passthrough, no transcoding)
- A clean HLS output that works in browsers + mobile
- Automatic reconnection if the RTSP camera drops
- Simple, scalable directory-based configuration

1 — Update and install dependencies

```
sudo apt update
sudo apt install -y ffmpeg nginx python3-yaml openssl
```

2 — Directory setup

```
sudo mkdir -p /usr/local/ffmpeg-hls
sudo mkdir -p /var/www/html/cctv
sudo chown -R www-data:www-data /var/www/html
```

3 — Camera configuration file

Create file `/usr/local/ffmpeg-hls/cams.yml`

```
- name: cam1
  url: rtsp-url-here
- name: cam2
  url: rtsp-url-here
- name: cam3
  url: rtsp-url-here
```

```
# Add up to cam16 if needed...
```

4 — Auto FFmpeg launcher script

Create file `/usr/local/ffmpeg-hls/start-all-cams.sh`

```
#!/bin/bash
CONFIG_FILE="/usr/local/ffmpeg-hls/cams.yml"
OUTPUT_ROOT="/var/www/html/cctv"
HLS_DURATION=10            # seconds per segment
TOTAL_HISTORY_MIN=10       # total minutes to keep
SEGMENTS_TO_KEEP=$(( (TOTAL_HISTORY_MIN*60) / HLS_DURATION ))

mkdir -p "$OUTPUT_ROOT"

#Generate list of active cams
python3 - <<'PYCODE' > /tmp/camlist.txt
import yaml, sys
data = yaml.safe_load(open("/usr/local/ffmpeg-hls/cams.yml"))
for cam in data:
    print(f"{cam['name']}|{cam['url']}")
PYCODE

#Build array of active cam names
ACTIVE_CAMS=($(awk -F'|' '{print $1}' /tmp/camlist.txt))

#Clean up old folders not in YAML
for dir in "$OUTPUT_ROOT"/*/; do
  [ -d "$dir" ] || continue
  CAM_NAME=$(basename "$dir")
  if [[ ! " ${ACTIVE_CAMS[@]} " =~ " ${CAM_NAME} " ]]; then
    echo "Removing old folder: $dir"
    rm -rf "$dir"
  fi
done

#Start each active camera
while IFS="|" read -r NAME URL; do
    [ -z "$NAME" ] && continue
    mkdir -p "$OUTPUT_ROOT/$NAME"
```

```
    echo "Starting $NAME ..."
    (
      while true; do
        ffmpeg -hide_banner -loglevel warning \
          -rtsp_transport tcp \
          -i "$URL" \
          -fflags +genpts -use_wallclock_as_timestamps 1 \
          -an -c:v copy \
          -f hls \
          -hls_time $HLS_DURATION \
          -hls_list_size $SEGMENTS_TO_KEEP \
          -hls_flags delete_segments+append_list+program_date_time \
          -hls_segment_filename "$OUTPUT_ROOT/$NAME/segment_%05d.ts" \
          "$OUTPUT_ROOT/$NAME/index.m3u8"

        echo "$NAME disconnected, retrying in 5s..."
        sleep 5
      done
    ) &
done < /tmp/camlist.txt

echo "All active camera streams started!"
wait
```

Make it executable:

```
sudo chmod +x /usr/local/ffmpeg-hls/start-all-cams.sh
```

5 — Systemd service

Create file `/etc/systemd/system/cctv-hls.service`

```
[Unit]
Description=FFmpeg HLS Auto Streams (All Cameras)
After=network-online.target
Wants=network-online.target

[Service]
ExecStart=/usr/local/ffmpeg-hls/start-all-cams.sh
Restart=always
RestartSec=10
LimitNOFILE=65535
```

```
[Install]
WantedBy=multi-user.target
```

Apply changes

```
sudo nano /usr/local/ffmpeg-hls/start-all-cams.sh
# (paste script above)
sudo chmod +x /usr/local/ffmpeg-hls/start-all-cams.sh
sudo systemctl restart cctv-hls
```

6 — HTTPS setup with NGINX (self-signed)
Generate SSL certificate:

```
sudo mkdir -p /etc/ssl/cctv
sudo openssl req -x509 -nodes -days 3650 -newkey rsa:2048 \
  -keyout /etc/ssl/cctv/cctv.key \
  -out /etc/ssl/cctv/cctv.crt \
  -subj "/CN=cctv.local"
```

Configure NGINX site:
Create file `/etc/nginx/sites-available/cctv`

```
server {
    listen 443 ssl;
    server_name _;

    ssl_certificate     /etc/ssl/cctv/cctv.crt;
    ssl_certificate_key /etc/ssl/cctv/cctv.key;

    root /var/www/html;
    index index.html;

    location /cctv/ {
        add_header Cache-Control no-cache;
        types {
            application/vnd.apple.mpegurl m3u8;
            video/mp2t ts;
        }
        autoindex on;
        add_header Access-Control-Allow-Origin *;
```

```
    }
}

server {
    listen 80;
    server_name _;
    return 301 https://$host$request_uri;
}
```

Enable and reload:

```
sudo ln -s /etc/nginx/sites-available/cctv /etc/nginx/sites-enabled/
sudo rm /etc/nginx/sites-enabled/default
sudo systemctl restart nginx
```

7 — Verify everything

```
sudo systemctl status cctv-hls
```

Then visit:

```
https://<server-ip>/cctv/cam1/index.m3u8
https://<server-ip>/cctv/cam2/index.m3u8
https://<server-ip>/cctv/cam3/index.m3u8
```

Revision #8
Created 3 November 2025 12:40:50 by Neunix
Updated 4 November 2025 12:18:41 by Neunix