AI Automation For Short Form Content

Full Guide

Summary

Introduction

This guidebook explains how to use the **Clipping Automation** tool included with this project.
The goal of this project is to make professional-grade video clipping **completely free** and **locally hosted**, with **zero cost** to you.

Start by reading **Section 1: Prerequisites** carefully — don't skip any point, as it lays the foundation for the setup.
Then, move on to **Section 2: Usage**, which will guide you through the setup, show how the workflow operates, and explain how to start using it efficiently.

Finally, explore **Section 3: Customization** to learn how to fine-tune the automation and control the output exactly the way you want.

Prerequisites

Before setting up, here's an overview of what we'll install, what tokens we'll need, and why each service is important.
We will install everything locally using Docker, but also expose the services securely to the internet using Ngrok.

Services We Will Install

1. Ngrok – Exposes your local services to the internet with a secure tunnel, required for external API callbacks to work with n8n.
2. Docker – Runs all required tools (n8n, MinIO, NCA Toolkit) in isolated containers.
3. n8n – Automation/orchestration tool for the entire video generation pipeline.
4. MinIO – Local object storage for files (audio, video segments, etc.).
5. NCA Toolkit – Video editing tool based on ffmpeg.

Disk Space Recommendation:
You should have at least 50+ GB free. Docker images and containers will consume ~30 GB, plus Docker's own system files.

Required Tokens & API Keys

We'll be using several free-to-use API services. You will need to create accounts and generate API keys for each one.

Service	Sign-Up / Docs
Groq Console	Get Groq Token
Google Cloud Console (Tutorial - Youtube)	Get Google Cloud Token
Telegram Bot Token (Tutorial - Youtube)	Get Telegram Bot Token
Google Ai Studio	Get Ai Studio Key

=> Keep all API keys in a secure file (.txt) so you can easily grab and use them later.

1. Ngrok

Purpose: Allows you to make your local services (like n8n) accessible from the internet via a temporary public HTTPS URL.
This is critical because n8n's webhooks must be reachable by external APIs.

Installation:

1. Download Ngrok: Ngrok link
2. Install it on your system.

3. follow this guide to get your public HTTP URL : Tutorial - Youtube

2. Docker

Purpose:
Container engine that will run all our tools without messy installations.

Installation:

- Download from Docker Desktop
- On Windows:
- Install with WSL2 (faster than Hyper-V)
- You can cap RAM/CPU usage via `.wslconfig` :

```
[wsl2]
memory=5GB                 # Max RAM for WSL2
processors=3               # Number of CPU cores to use
swap=1GB                   # Swap file size
localhostForwarding=true
```

3. n8n

Purpose:
Automation tool for orchestrating the video generation process.

Docker Command:

```
docker run -d                                   # Run container in detached mod
--name n8n                                    # Name the container "n8n"
-p 5678:5678                                  # Map port 5678 for n8n UI
-e N8N_COMMUNITY_PACKAGES_ALLOW_TOOL_USAGE=true  # Allow extra packages
-e N8N_EDITOR_BASE_URL=YOUR_OWN_LINK          # Public URL from Ngrok
-e WEBHOOK_URL=YOUR_OWN_LINK                  # Public URL from Ngrok for webho
-e N8N_DEFAULT_BINARY_DATA_MODE=filesystem    # Store binary data on filesystem
-v /c/Docker/n8n-data:/home/node/.n8n         # Persist n8n config/data  docker
```

Note: Replace `YOUR_OWN_LINK` with your Ngrok public link.

4. MinIO

Purpose:
Local S3-compatible storage for assets.

Docker Command:

```
docker run -p 9000:9000                         # API endpoint port
-p 9001:9001                                  # Web console port
--name minio                                  # Name container "minio"
-v C:\Docker\minio-data:/data                 # Data persistence folder
-e MINIO_ROOT_USER=admin                      # Admin username
-e MINIO_ROOT_PASSWORD=password123            # Admin password  quay.io/minio/m
server /data --console-address ":9001"        # Start MinIO server
```

After starting MinIO:

1. Visit: `http://localhost:9001`
2. Login with the username & password set above.
3. Create a bucket named exactly: `nca-toolkit` .
4. Go to Access Keys → Create Access Key → Save both Access Key & Secret Key. These will be used in NCA Toolkit's configuration.

YouTube Guide:

- MinIO Tutorial - Youtube

---

5. NCA Toolkit

Purpose:
Processes videos, applies effects, adds captions, compiles final outputs.

Docker Command:

```
docker run -d \                               # Run container in detached (back
  -p 8080:8080 \                             # Expose API on port 8080
  --name nca-toolkit \                       # Name the container "nca-toolkit"
  -e API_KEY=thekey \                        # API key used for authentication
  -e GUNICORN_TIMEOUT=3000 \                 # Allow long-running jobs (caption
  -e GUNICORN_WORKERS=1 \                    # Single worker for stability (FFm
  -e S3_ENDPOINT_URL=http://host.docker.internal:9000 \  # MinIO / S3-compatibl
  -e S3_ACCESS_KEY=MINIO_ACCESS_KEY \         # S3 access key (from MinIO)
  -e S3_SECRET_KEY=MINIO_SECRET_KEY \         # S3 secret key (from MinIO)
  -e S3_BUCKET_NAME=nca-toolkit \             # Bucket name (must already exist
  -e S3_REGION=None \                         # Region (use None for local MinI
  stephengpope/no-code-architects-toolkit:latest  # NCA Toolkit API image
```

Notes:

- Ensure MinIO bucket `nca-toolkit` exists before running this container.
- Use your MinIO Access Key & Secret Key here.
- Save the `API_KEY` (default `thekey`) — it will be needed for n8n API calls.

YouTube Tutorials:

- NCA Toolkit Tutorial - YouTube By Stephen G. Pope (Creator of the Tool)
- NCA Toolkit Tutorial (Alt) - YouTube

GitHub Repository:

The official NCA Toolkit - GitHub is well-documented and maintained by the tool's creator. It also includes a custom GPT for additional assistance.

⚠️ IMPORTANT: NCA-Toolkit Integration with n8n

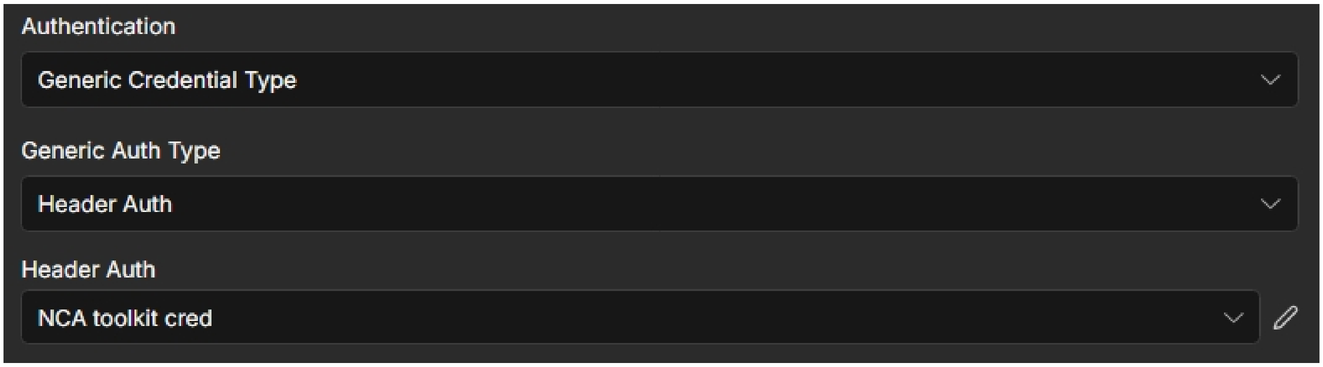
(Important: Without this setup, NCA-Toolkit nodes will not run)

When using n8n to connect with NCA-Toolkit via an HTTP Request node, you must configure credentials — otherwise, the request will fail.

Step 1 – Set Authentication

In the HTTP Request node:

1. Set Authentication to Generic Credential Type.
2. Select Header Auth as the method.



Authentication

Generic Credential Type

Generic Auth Type

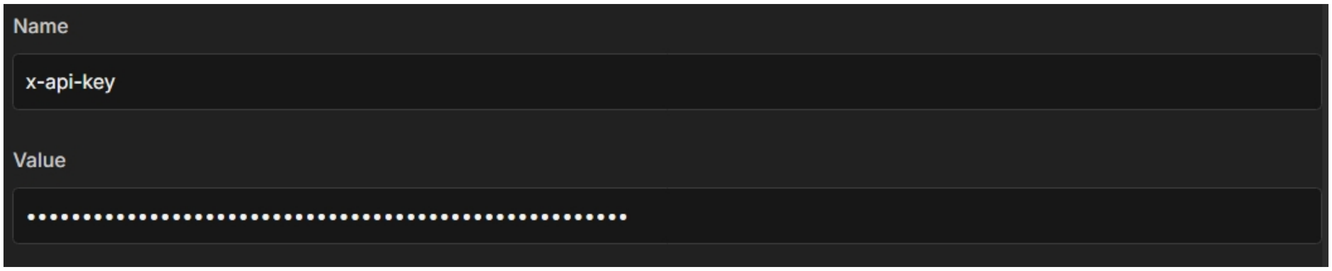
Header Auth

Header Auth

NCA toolkit cred

Step 2 – Add Credentials

1. For the credential name, enter it exactly as shown in the image below.
2. For the value, use the API key you configured in the Docker setup.



Name

x-api-key

Value

••

Important Note:

Do not modify the ports or advanced settings unless you are experienced with Docker and n8n configurations, and changing them incorrectly will cause the tool to malfunction.

Resource Limits (Prevent Crashes)

To avoid Docker crashes due to excessive resource usage, you can cap CPU and memory for your containers (especially Nca-toolkit).

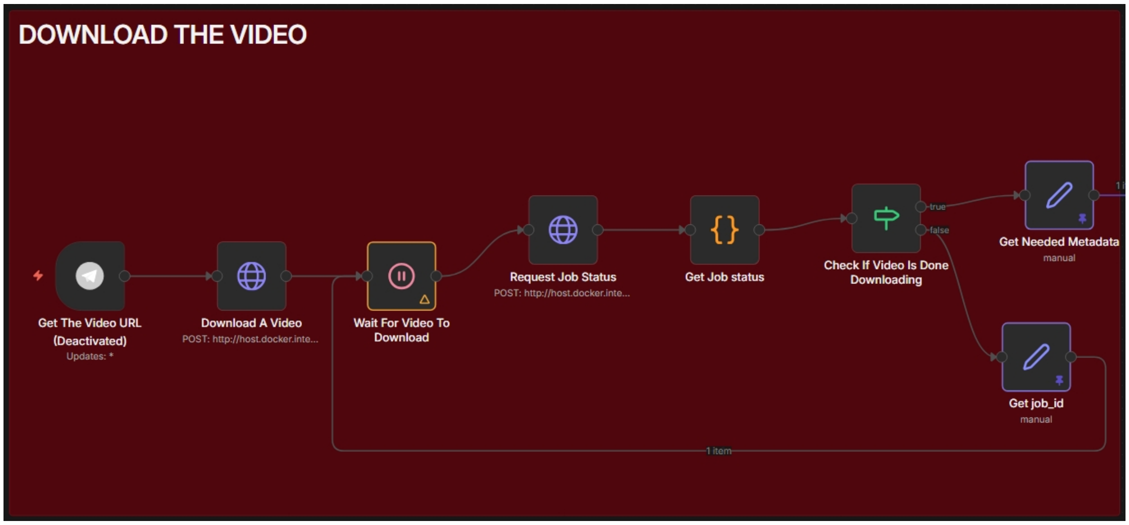
Recommended Limits:

```
docker update
--cpus=2        # Limit to 2 CPU cores
--memory=2G     # Limit to 2GB RAM
```

```
  --memory-swap=3G  # Allow 1GB swap (3G total incl. RAM)
  container-name
```

Usage & Customization

Part 01: Downloading The Long Form Video


DOWNLOAD THE VIDEO
Get The Video URL (Deactivated)
Updates: *
Download A Video
POST: http://host.docker.inte...
Wait For Video To Download
Request Job Status
POST: http://host.docker.inte...
Get Job status
Check If Video Is Done Downloading
true
false
Get Needed Metadata
manual
Get job_id
manual
1 item

In this step, we download the long-form video that will later be split into short clips. The process works as follows:

1. **Get the Video URL**

- The URL of the video is retrieved from a Telegram chat.

1. **Download the Video**

- We use the **NCA-Toolkit** service, which leverages **yt-dl** under the hood to download the video.
- The video is downloaded in **Full HD (FHD)** for the best quality output.
- ⚠️ Depending on your network speed, downloading may take some time.

1. **Check Download Status**

- The `Request Job Status` node monitors the download progress.
- It continuously checks whether the job is still running or has completed.

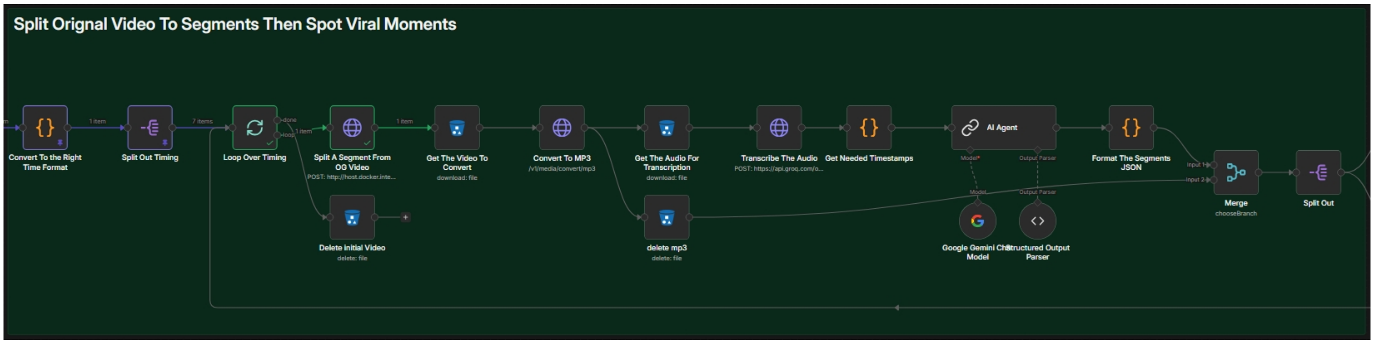
1. **Retrieve Video Metadata**

- Once the download is complete, the system provides:

    - **Title** of the video
    - **Media URL** stored in **MinIO** (local storage, so no additional download is needed)

- If the download is not yet complete, the node will re-check the status until the video is fully available.

Part 02: Splitting, Transcribing, and Detecting Viral Moments



In this step, the long-form video is processed to identify the most engaging segments. The workflow is as follows:

1. **Segment the Video**

- The video is split into **5-minute segments** to make processing more manageable.
- Each segment is processed in a loop until the entire video is covered.
    - Example: a 1-hour video will produce **12 segments**.

1. **Convert to Audio**

- Each segment is converted into **MP3 format** for transcription.

1. **Transcribe the Audio**

- Using the **Groq Console Api, the audio is transcribed.
- The output includes **timestamps** corresponding to periods of speech and silence.

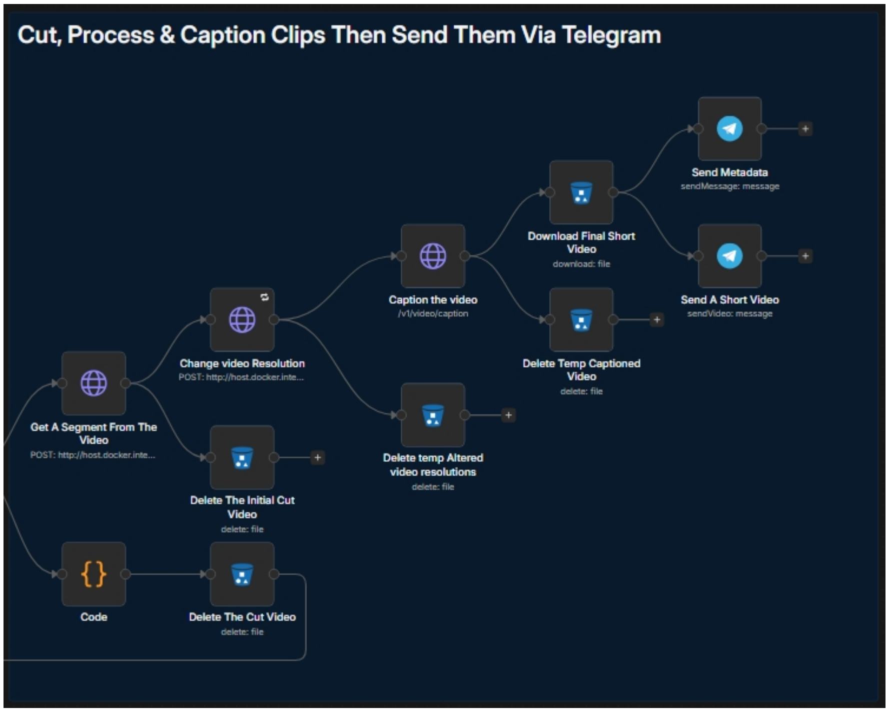
1. **Detect Viral Moments**

- The **AI agent** receives both the transcription and the timestamps.
- It analyzes the content to **identify viral moments**—segments worth highlighting or clipping.

1. **Prepare for Final Output**

- Once all segments are processed, the system moves on to generating the **final result**, containing all the selected clips from the current segment of the original video.

Part 03: Get & Process Then Send The Final Videos


Cut, Process & Caption Clips Then Send Them Via Telegram
Send Metadata
sendMessage: message
Download Final Short Video
download: file
Send A Short Video
sendVideo: message
Caption the video
/v1/video/caption
Delete Temp Captioned Video
delete: file
Change video Resolution
POST: http://host.docker.inte...
Get A Segment From The Video
POST: http://host.docker.inte...
Delete temp Altered video resolutions
delete: file
Delete The Initial Cut Video
delete: file
Code
Delete The Cut Video
delete: file

In this step, we take the segments identified as containing viral moments and turn them into ready-to-upload short videos. The process works as follows:

1. **Clip Extraction**

- From each **5-minute segment**, clips are extracted based on the **AI agent's viral moment analysis**.

1. **Format for Short Videos**

- The extracted clips are resized to **1080×1920**, ideal for **YouTube Shorts, TikTok, or any vertical short video platform**.

1. **Captioning**

- Each clip is captioned automatically using **NCA-Toolkit**.
- If you want to **customize caption styles**, go to the `Caption the Video` node.
    - Reference values and options can be found in this GitHub repository.

1. **Attach Metadata**

- The final video is sent along with important metadata to help with uploading:
  - **Title**
  - **Description**
  - **Hashtags**
  - **Viral Score**

1. **Delivery and Cleanup**

- The final clip and metadata are sent via **Telegram**.
- Temporary files stored in **MinIO** are automatically removed to **free up space** and **avoid issues**.