

How I Make a Month of Quality Content in a Day

The whole content operating system, revealed: every tool, skill, agent, and dynamic workflow I actually use, free to download as you read.

By **Enrique Marq** · INFINITX / Nave
nave.infinityai.com · @enriquemarq-0

One focused day gets you a month of content. This is the whole method, the real one, with the real system behind it. Read it, do it, keep it. And the actual pieces I run (the skills, the agents, the dynamic workflows) are yours to download as you go. I held nothing back.

What you're actually looking at: a content operating system

For the last six months I've been building one thing. Not a stack of subscriptions. A **content operating system**: one machine that takes raw footage in and pushes finished, on-brand, scheduled content out.

I run the whole thing inside **Claude Code**. That's where the work happens: it reads your footage, cuts the clips, writes the captions, drives the renders, and schedules the month. Around it I've wired a small stack of tools, each one the best at its one job:

- **AssemblyAI (Universal-3 Pro)** — *speech-to-text*. I use it for one thing: a word-level, time-stamped, near-perfect transcript of every drop. The spine the whole day hangs off.
- **The clip engine** — *face-tracking reframe*. Turns wide 16:9 footage into clean 9:16 verticals that follow the speaker, no lazy center-crop.
- **Hyperframes** — *the editing framework*. Renders finished, captioned, animated video from plain HTML. This is what makes a clip look *made*, not auto-cropped.
- **KIE.ai** — *AI image, video, and music generation*. I use it only to fill a gap: a shot I never filmed, a cover image, a music bed. Never to fake the whole thing.
- **Zernio** — *publishing + comment-to-DM + analytics*. Posts to every platform at once, runs the "comment a keyword, get a DM" automation, and reports back what actually landed.

That's the stack. But the stack isn't the system. What turns it into an *operating system* is three working parts, and these are the whole reason it holds together:

- **Skills** — a written recipe the system follows the same way every time ("transcribe this," "score these clips," "build a carousel").


- **Agents** — a specialist worker with one job and the taste to do it (a clip-editor that knows my caption rules; a humanizer that de-AIs every line). You can run several in parallel while you make tea.
- **Dynamic workflows** — a script that *orchestrates* the agents: fan ten clips out to ten editors at once, collect the results, assemble the batch. One command, and the whole month moves.


And here's the part that surprises people: these aren't settings buried in an app. They're **plain files in a project**, exactly like opening a folder in VS Code. You can read them, change them, keep them:

```
your-project/
  .claude/
    skills/
      extracting-transcripts/SKILL.md      a recipe (the "how")
      zernio-publish/SKILL.md
    agents/
      humanizer.md                        a specialist worker
    workflows/
      footage-intelligence.workflow.js     orchestrates the work
```

Open any one and you see exactly what it tells the system to do. Nothing is hidden. That's why I can hand them to you. As we walk the steps, the core pieces have a download button next to them. Grab them. They're yours.


Do this now: Decide you're building a system this month, not posting today. That single reframe, *machine I own* instead of *treadmill I feed*, is what everything below is built on.



 **YouTube**

Watch me walk through the whole system

The same process, on camera: the folder walkthrough, the stack, and the real client run. Scan the code or tap the thumbnail.



SCAN TO WATCH

Why a content day beats a content grind (and why you need a system)

The lie I believed for too long: that staying visible meant showing up every day. Wake up, think of something, film it, caption it, fight the algorithm, do it again tomorrow. Forever. The work was scattered

and endless, and the day I got busy or sick or just tired, the whole thing fell apart. It lived or died on me being "on."

That's a grind, and a grind is fragile by design.

The fix isn't posting more. It's posting from a batch. One focused input day that produces a month: you sit down, capture your real thinking once, and turn that single block of raw material into thirty-ish pieces. The posting spreads across the month. The heavy lifting happens in one sitting.

- **Grind:** small effort, every day, forever. Never compounds. The day you stop, the content stops.
- **Content day:** one concentrated batch a month. Front-load the work, then it pays out for thirty days while you run your actual business.

Same output. Completely different cost to your life.

Here's the part nobody says, though: a content day only works if you have a **system** to run it. Do all of this on willpower and a folder full of random clips and you'll burn out worse than the grind ever managed. What makes the day repeatable, month after month, is the content operating system: the stack, the skills, and the workflow doing the heavy lifting inside a structure that already knows where everything goes.

So that's what we build first. Not the footage, not the editing. The system. And it starts with something boring that makes everything after it easy: how the whole thing is organized.

Where everything lives (the folder setup)

This is the part I actually walk through on screen in the video, because it's the difference between a system that runs itself and a pile of clips you can't find two weeks later.

One rule. **One folder per piece, and everything for that piece lives inside it.** The footage, the transcript, the clips, the edit, all in one place. Never scattered. Around those, a few shared libraries you reuse on every piece.

Here's the actual shape on my disk:

```
content-production/  
  in-production/  
    2026-0609-clip-gang/  
      transcript/  
      clips/
```

the pieces I'm working on right now
one folder per video, named date + slug
the words (transcript.json + .srt)
the cut + reframed verticals

cuts/	raw trims off the source
clip-01-.../reframed/	the 9:16 face-tracked clip
work/	scratch: audio, sampled frames
edit/	the edit composition
posted/	what's already shipped
pipeline/	what's planned next
b-roll-library/	reusable b-roll
cover-template/	the thumbnail / cover system
frame-pack/	reusable frames + assets
swipe-file/	references I borrow structure from

Here's why it matters more than it sounds. The system reads the structure. When every piece sits in a predictable folder with the same shape, the skills and the workflow know exactly where the footage is, where to write the transcript, where the clips go. You stop telling it paths. You drop footage into a new dated folder, point the system at it, and it already knows what to do. The folder layout isn't housekeeping. It's part of the machine.

Do this now: Make one folder named for today's piece (date plus a short slug, like `2026-0614-my-topic`). Inside it, make four empty folders: `transcript`, `clips`, `work`, `edit`. That empty shell is the only setup you do before a content day. Everything below fills it in.

Step 1 — One footage drop

So let me ask you the question I ask every single client: what's the first step when you want to publish a piece of content?

For me, if I'm publishing a video, I record it first. Then comes the cutting, the editing, the packaging, the publishing. If I'm making a carousel, I gather a few images or take a picture of my face and hand it to Claude. Either way, the first move is the same. You drop the footage. Everything else is downstream of it.

This is the thing nobody tells you, and it's what makes the whole month possible: **a month of content is not a month of filming. It's one good input, mined properly.**

When I started, I thought "post every day" meant "make something every day." So I'd scramble: film a thing, edit it, post it, wake up empty, do it again. That's the grind. It burns you out and the quality slides, because you're always starting from zero.

The fix is stupidly simple and it took me too long to trust it. One footage drop. A single batch of raw input the whole month gets built from. You gather it once, then you spend the day turning that drop into ten, fifteen, twenty pieces.

What counts as a footage drop

Any one of these, as long as it's one rich session:

- **A talking-head session.** You on camera, talking through something you actually know. 20 to 40 minutes, loose, unscripted is fine. The most flexible seed. It spins out the most formats.
- **A walkthrough.** You doing the thing — building, fixing, showing your screen, narrating as you go.
- **A podcast or interview you were on.** Someone already pulled good answers out of you. A goldmine you didn't have to set up.
- **A screen-record of real work.** You solving an actual problem, talking over it. Genuinely useful, zero production.

And yes, you can generate footage. AI image, AI voice, AI b-roll. Do I recommend it? In some cases, yes — to fill a gap, not to replace the real thing. Your face and your real voice are the asset. Use generated footage where you're missing a shot, never as the whole drop.

How much footage do you actually need?

The honest answer: enough to cover a full month of distribution. That's the test. Not "did I film something," but "is there enough real material here to cut a month of distinct pieces from."

How much that is depends on what you're making. When I ask a client for footage, I ask for it recorded across **at least two to three days, with a purpose.** Not random phone clips. Intentional capture: this is the thing we're documenting, here's the list of shots we need, go get them over a few days. Two or three purposeful days is almost always enough for a month. (I'll show you a real client run at the end where two days of footage became a full month of content in a single editing day.)

How I pick an input rich enough to spin a month from

Not every recording gives you a month. A 4-minute "here's a tip" clip is one piece, not a drop. Here's the lens I run:

- **Does it have multiple distinct points?** Five or six separate things means five or six spotlight clips, a compilation, and a carousel hiding inside. One point gives you one clip. Many points give you a month.
- **Is there at least one real money line?** A moment where you say the thing crisp and quotable. That's your montage anchor and probably your best short.
- **Can you talk for 20+ minutes without running dry?** That's the honest test of whether you actually know it. The topics you can ramble on forever mine the richest.

- **Would you watch it?** If the raw thing bores *you*, no edit saves it. Energy is the one thing you can't add in post.

Don't over-produce the drop. No lighting rig, no script, no five takes. You want a deep, useful, real recording, not a polished one. The polish happens later, format by format.

DYNAMIC WORKFLOW · YOURS, FREE

footage-intelligence

This is the intake, and it's a real dynamic workflow. It reads every clip in a drop in parallel: WHO is on screen (faces to names), WHAT it shows (usable b-roll vs. filler), and what's SAID (it transcribes each clip and pulls the quotable lines), into one library the rest of the system works from. The transcribe step (next) is this workflow's "said" layer. You get the actual orchestration script, free.

```
footage-intelligence.workflow.js – fans every clip out at once:  
per-clip who/what/said + the quotable lines → ONE library doc,  
so every later step reads structured footage, not raw files.
```



Get the workflow

FREE

Do this now: Pick your one topic, the one you could talk about for 30 minutes without notes, that has at least five distinct points inside it. Block time this week to record it across a couple of intentional sessions. Don't film ten small things. Film one deep thing, with a purpose. That's the seed for everything that follows.

Step 2 — Transcribe with word-level timing

This is really the first thing footage-intelligence does. Remember the three things it reads off your drop: who's on screen, what's usable, and what's *said*. That last one, the "said," is this step. Transcription isn't a separate tool bolted on; it's the part of footage-intelligence that turns your talking into text the rest of the system can read. I'm pulling it out into its own step because it's the single most important file you'll make all day.

Before I cut a single frame, I get the words. Not the gist. The words, each one stamped with the exact millisecond it's spoken. That timed transcript is the spine of the whole day. Captions hang off it. Cuts land on it. The "find me the best 45 seconds" pass reads it. Skip it and you're back to scrubbing a timeline by ear, the slow way I used to lose whole afternoons.

I run every drop through **AssemblyAI** on its **Universal-3 Pro** model.

Why this tool, specifically

I've tried the others. Universal-3 Pro is the most accurate speech engine I've used on *real* creator audio: phone mics, room echo, two people talking over each other, Spanglish, all of it. And it gives you word-level timestamps natively, no hacks. That accuracy isn't a nice-to-have. Every cut, every caption, every clip-pick downstream is only as good as this transcript. Get it wrong and the whole day is built on sand.

It's also cheap, and it's *yours*. You bring your own AssemblyAI key. No middleman, no markup, nobody sitting between you and the tool. You pay them directly and it's cents: about **\$0.27 an hour of audio**. A 30-minute interview costs you fourteen cents.

What it looks like under the hood (the real API)

You don't have to touch this. Claude Code does it for you. I'm showing it so you see there's no magic, just a clean API you own the keys to. Two calls:

1. **Upload** your file to `POST https://api.assemblyai.com/v2/upload` with your key in the `authorization` header. You get back a URL.
2. **Transcribe** it: `POST https://api.assemblyai.com/v2/transcript` with `{"audio_url": "...", "speech_model": "universal-3-pro"}` . Poll until it's done. The response hands you every word with `start` and `end` times in milliseconds.

That's the entire integration. Your key, your bill, your transcript on your disk.

How to actually do it (non-coder friendly)

1. **Get your key.** Free account at assemblyai.com, open the dashboard, copy your API key. Treat it like a password.
2. **Hand the audio and your key to Claude Code.** Point it at your file and say it plainly: *"Transcribe this with AssemblyAI Universal-3 Pro and give me word-level timestamps. Here's my key."* It uploads, runs the job, waits, saves the result. You never touch a command line.
3. **Be picky about two things:** the model name is `universal-3-pro` , and you want the *word-level* output, not just the paragraph. Default to a lesser model and you lose accuracy; grab only the plain text and you lose the timestamps, which are the entire point.

SKILL · YOURS, FREE

extracting-transcripts

The exact skill I use for this step. Hand it a video or audio file and your AssemblyAI key; it uploads, runs Universal-3 Pro with word-level timing, and writes a clean transcript.json + .srt to disk. The locked, accuracy-first config — no guessing at model names or flags.

```
extracting-transcripts/SKILL.md
→ speech_model: universal-3-pro (locked default)
→ word-level timestamps ON
→ writes <out>/transcript.json + transcript.srt
→ keyterms_prompt for names/jargon it should spell right
```



**Download the
transcribe skill**

FREE

Do this now: Make your free AssemblyAI account, copy your key into a private note, take the longest drop from Step 1, hand it and your key to Claude Code, and say: *"Transcribe this with AssemblyAI Universal-3 Pro, word-level timestamps, save it to a file."* When there's a words file on disk with real start and end times, Step 2 is done. And you never opened a terminal.

Step 3 — Mine the moments

You've got a word-level transcript. Right now it's a wall of text. Don't let that intimidate you. That wall is your whole month of content, you just can't see the pieces yet.

Here's what nobody tells beginners: a "month of content" is not a month of ideas. It's a handful of strong moments you already said once, pulled out and cleaned up. Your job isn't to write anything new. It's to read your own transcript like a stranger would and mark the parts worth someone's attention.

This is the taste layer. It sounds like the hard part — like something you need ten years of editing to do. You don't. Here's the exact lens, and it's three questions.

The three questions I ask every moment

- **Would I stop scrolling for this?** Be brutally honest. Picture this line in your own feed between a hundred other videos. Stop, or keep going? Most of your transcript is keep-going. That's fine. You're hunting for the few seconds that are stop.
- **Does it teach something or prove something?** A moment earns its place one of two ways. It teaches: a real step, a how. Or it proves — a result, a number, a story, a receipt. If it does neither, it's filler. Cut it loose, no matter how good it sounded coming out of your mouth.

- **Is there a clean hook in the first 3 seconds?** The strongest moments open on the point: a claim, a number, a contradiction, "here's the mistake everyone makes." If the good line is buried 15 seconds in, fine. Just mark where the real start is.

Clears all three, it's a piece of content. Clears two, it's a maybe. Clears one, it's a no.

What you're looking for

- **Hooks:** a line sharp enough to be the first 3 seconds by itself.
- **Money lines** — the one-sentence version of something true and a little contrarian.
- **Teachable steps:** anywhere you explain how to do a thing in order. These become carousels and "here's exactly how" clips.
- **Proof beats:** a result, a before/after, a number, a short story. Trust gets built here, not in the teaching.

From one good drop you'll usually pull 8 to 15 moments. Star your top 5. Those are your spotlight clips. The rest become montage fuel, carousel slides, and supporting cuts.

One thing that frees people up: most of your transcript is not content, and that's normal. A 40-minute recording might give you 12 real moments. Twelve is a great month. Don't force the weak parts in to hit a number.

SKILL · PART OF THE SYSTEM

clip-selection

This is the taste layer, automated. It reads your timed transcript and scores every candidate moment on a 5-category framework (hook strength, payoff, standalone clarity, emotion, shareability), then hands back a ranked shortlist with exact in/out timestamps. The same three questions above — run on every line, the same way every time. You can absolutely do this by hand with the three questions; this is the version that's pre-tuned and assembled inside Nave.

```
clip-selection — scores each candidate 1-5 on: hook ·  
teaches/proves · stands alone · emotion · shareable →  
ranked list with start/end (ms) + the reason it scored.
```



Assembled in Nave
IN NAVE

Do this now: Open your transcript. Read it once clean. Read it again and pull every moment that clears all three questions into a notes file, timestamp and shape (hook / money line / step / proof) on the line above each. Stop when you've got your shortlist and stars on your top 5. That list is your month.

Step 4 — Cut and reframe to vertical

You've got your transcript and your clip picks. Now you turn a wide 16:9 frame into a 9:16 vertical that looks like it was *shot* that way, not a lazy crop with someone's head jammed against the top.

This is where most people give up and slap a center crop on it. Don't. The gap between a cheap reframe and an intentional one comes down to about five rules, and here are all five.

Cut on a completed sentence, with a short tail

Set your in and out points off the transcript. Not by ear, by the words.

- **Start** right as the first real word lands. Trim the dead air before the hook. If there's an "um" or a breath in front, cut it.
- **End** on a finished thought. Never stop mid-sentence. A clip that cuts off "...and that's why you should —" feels broken and people bounce.
- Leave a short tail after the last word, about 1.5 to 2 seconds. That little breath is what makes it feel edited instead of yanked.

Face-track the reframe, don't static-crop

A static crop assumes the person never moves. They always move. The crop has to follow the face. Two things make tracked motion look professional instead of seasick:

- **Smoothing.** The crop should drift, not snap. Let it lag the face slightly so small head bobs don't whip the frame around.
- **A deadzone.** Tiny movements (someone shifting in their seat) shouldn't move the crop at all. Only follow real movement across the frame.

The caption-safe center band (the one nobody tells you)

This is the fix I learned the hard way. Your captions sit in the lower-middle of a vertical video. If your face is *also* dead-center, the captions land right on your mouth and chin. Looks cramped, looks amateur.

So drop the face a little higher than center. Eyes in the upper third, lower-middle band clear for text. The captions get a clean lane and the face gets room to breathe. This single adjustment is the biggest "why does mine look cheap" fixer there is.

Never leave black gaps

When the crop slides to follow a face near the edge, a naive tool runs *off* the source and fills the gap with black bars. That reads as broken instantly. The rule: clamp the crop to the edge of the source. When the

face nears the edge, the crop stops and lets the face drift within frame instead of exposing black. Full frame, edge to edge, every second.

Music ducks under the voice

If you lay music under the clip (you usually should), it can't fight the talking. Keep it low the whole time — a bed, not a track. Duck it further whenever someone speaks so the voice always sits on top. Let it come up in the gaps, then drop it the moment words return. You should catch every word without straining.

ENGINE · PART OF THE SYSTEM

clip-machine + clip-extractor (the reframe engine)

This is the actual face-tracking reframe engine I run, not a CapCut button. It finds the mouth every few frames and slides a smoothed, deadzoned crop to keep the speaker centered — clamped to the source so it never shows black bars, with the caption-safe band built in. The five rules above are yours to apply in any editor; this is the engine that does them automatically, and it's the heavy machinery I keep inside Nave (mediapipe, models, the Windows render fixes — the stuff that's a pain to stand up alone).



Engine in Nave

IN NAVE

```
clip-extractor - mouth-centered face track:  
smoothing + deadzone (no seasick) · clamp-to-source (zero  
black gaps) · caption-safe band · 1080x1920 CFR-30 out.
```

Do this now: Take one clip pick from Step 3. Set the in-point on the first real word and the out-point on a finished sentence plus a ~2-second tail. Reframe it to 9:16. Then check the thing everyone misses. Is the face sitting above center so the lower third is clear for captions? If the head's dead-center, nudge the crop up. That's a cheap reframe turned intentional in two minutes.

Step 5 — Edit to a real standard

Here's the line it took me too long to learn: a clip that's only reframed still gets scrolled past. Reframing makes it the right shape for the phone. *Editing* makes it the thing someone actually watches. That gap is everything.

Auto-reframed vs. deliberately edited

Auto-reframe says "this fits the screen." Deliberate editing says "this was made for you to watch." On top of the reframe you:

- Cut the boring windup so it starts on the point.
- Burn in captions people can read with the sound off.
- Pop the one or two words that carry the meaning.
- Put a clean title on it and one consistent accent color so it looks made on purpose.

Cut the windup, start **ON** the point

Almost everyone buries the good part: "so basically what I want to talk about today is..." Nobody waits through that. Find the first sentence that actually lands and make it your first frame. Quick gut check: play your first 3 seconds on mute. If a stranger wouldn't know what the clip is about, you started too early.

Burn-in captions (non-negotiable)

About 80% of people watch with the sound off. If your words aren't on screen, most of your audience hears nothing. "Burn-in" means baked into the pixels, not a separate subtitle file the platform might not show. Good ones: bold and big, lower third, **word-level timed** (each word appears as it's spoken, this is why Step 2's word-level transcript matters), high-contrast white with a dark stroke.

Keyword pops and one brand accent

Pick the one or two words in a sentence that carry it and emphasize them — a little bigger, in your accent color. Don't pop every word; if everything's emphasized, nothing is. And pick one accent color and use it everywhere. Mine is orange, **#FF7614**. When every clip carries the same color, your stuff gets recognizable in the feed before anyone reads a word.

Where the render actually happens

Most people reach for CapCut or Premiere here, and that's fine. But the reason my clips look consistent is that I don't hand-place any of this. I render the whole edit from **Hyperframes**, an engine that turns plain HTML into finished video. The captions, the keyword pops, the title card, the brand color: defined once as a template, then rendered identically onto every clip. That's how you get twenty clips that look like one brand instead of twenty one-off edits. It's also why the editing can be handed to an agent. The edit is *code*, not a mouse dragging clips around.

EDITING CREW · PART OF THE SYSTEM

hyperframes-edit + the clip-editor crew

hyperframes-edit is the editing standard above, written down — dense motivated moves, always-on word-level captions, keyword pops, title card, brand accent, all rendered from HTML so every clip matches. The clip-editor agent runs that standard end to end, coordinating its own little crew (scene-director, move-designer, sound-designer, asset-fetcher). This is the premium edit — the part that turns "reframed" into "made for you to watch" — and it's the heart of what Nave does for you. The standard is right here in the playbook; the assembled crew + render engine is Nave.



The crew is Nave
IN NAVE

```
clip-editor → scene-director · move-designer ·  
sound-designer · asset-fetcher → renders via  
hyperframes-edit → premium 9:16, captions always on.
```

Do this now: Take one clip from Step 4 and run the full pass: trim until the first frame is the actual point; burn in bold, word-level, lower-third captions; pop the single most important word in your hook in your accent hex; add a one-line title card. Then play the first 3 seconds on mute. If a stranger would get it, you've edited to a real standard. That's the bar for every clip after this.

Step 6 — Spin one drop into many formats

You already did the hard part. One drop, a clean transcript, the moments mined, a clip edited to a real standard. That raw material is worth far more than one post. The mistake most people make is treating a shoot as one video. It isn't. It's a month of content waiting to be cut a bunch of different ways. (When the footage has a gap, a shot you never filmed, an image, a voice line, that's where **KIE.ai** comes in, on your own key, to fill the hole instead of re-shooting. Use it only where you're actually missing something.)

Here's the shift. A format is just a recipe. Same steps, same order, every time. You build the recipe once and from then on you never think about it again. You just run it on the new footage. That's the whole trick to making a month of content in a day. You're not creating five things from scratch. You're running five recipes you already own.

The five recipes

1. Spotlight clips. The bread and butter. One strong moment, 20–45 seconds, vertical, word-by-word captions, face kept centered. From one good interview I pull 3 to 6 of these. (This is just Steps 4 and 5, run again and again.)

2. The montage. A fast, rhythmic stitch of the best 8–15 seconds across several moments. The energy reel: punch after punch cut to the beat. The one people share.

3. The compilation. Several speakers (or takes) answering the same question, stacked into one piece. Title card up top — "We asked them..." — then segment, segment, segment, then your CTA card. A name chip on each so viewers know who's talking. This is the exact recipe behind a client's "advice" compilation. Same build, just new segments fed in.

4. The carousel. The footage becomes words on slides. Pull the 6 to 10 best lines straight from your transcript. They're already the speaker's real phrasing, so it doesn't sound written. One idea per slide, hook slide first, soft CTA slide last, on-brand colors. This format costs almost nothing because the writing is already done. It's the transcript.

5. The long-form package. The full piece, lightly cleaned: dead air and "ums" cut, a title card, captions, intro and outro. The anchor everything short points back to, and what lives on YouTube.

Every one of those is the same footage. One drop, five formats, and none of them is "make a video" anymore. Each is a checklist you've already written.

FORMAT CREW · PART OF THE SYSTEM

carousel + the format specialists

The carousel skill builds a real swipeable carousel from your transcript lines (brand fonts, colors, retina slides). The format agents each own one recipe — a spotlight specialist, a compilation specialist — so they run in parallel on the same drop. The recipes above are yours to run by hand today; this assembled, brand-tuned crew is what Nave runs for you.

```
carousel → 8-12 brand slides from transcript lines.  
format agents → spotlight · compilation · etc.,  
run in parallel on one drop.
```



The crew is Nave
IN NAVE

But one piece of that crew you get free, because it makes everything you write better no matter who runs it:

AGENT · YOURS, FREE

humanizer

The actual agent I run on every caption, hook, script, and slide before it's saved. It strips 50+ AI tells (cliques, hedging, em-dash overuse, rule-of-three, robotic structure), then runs an adversarial second-opinion pass. Voice-neutral — it makes your words sound like you, not like a model. Runs anywhere, no special setup. This one's yours.

```
humanizer (agent) - de-AIs any text before save:  
cliques · hedging · buzzwords · em-dash overuse ·  
forced rule-of-three → then a devil's-advocate pass.  
(quotes exempt) Runs in your Claude Code, no deps.
```



Get the humanizer

FREE

Do this now: Take your drop, pick two recipes (I'd start with Spotlight and Carousel, they share the most material), write each as a literal numbered checklist, and run both by hand. Time yourself. That number is exactly what you're weighing when you decide whether to keep doing it by hand or hand it to a worker. Either way, you now own the recipe.

The piece that runs the whole batch: a dynamic workflow

This is the newest part of my system, and the one that turns "a long day" into "kick it off and walk away." So I want to show it to you on its own.

Up to now every step has been a skill (a recipe) or an agent (a worker). A **dynamic workflow** is the layer above both: a script that *orchestrates* the workers. Instead of editing clip 1, then clip 2, then clip 3, the workflow fans all of them out at once. Ten clips to ten clip-editor agents, running in parallel, then collected back into one batch. One command moves the whole month.

This is the actual shape of my main run:

1. **Ingest** the footage (or a URL).
2. **Understand it:** footage-intelligence builds the library; clip-selection ranks the moments.
3. **Fan out:** each top moment becomes a job handed to its own editor agent, all at the same time.
4. **Assemble:** the workflow waits for every clip, runs the quality gate, and stacks the finished batch plus the carousels and the long-form package.
5. **Stage** the whole month for your approval.

The reason this matters: a workflow is *deterministic*. It runs the same way every time. It doesn't get tired on clip number nine, and it does ten things at once instead of one. That's the difference between a day of

clicking and an hour of reviewing.

And you already have a real one. The `footage-intelligence.workflow.js` you grabbed back in Step 1 is a dynamic workflow. Open it and you'll see this exact pattern: it fans every clip out in parallel, reads each one (who / what / said), and collects it into a single library. That's the orchestration move, in a file you can read and copy. The full run above, fanning out to a whole crew of editor agents and assembling the finished batch, is the same pattern scaled up, and that assembled crew is what Nave runs for you.

Do this now: Open the footage-intelligence workflow you already downloaded and read how it fans the work out. You don't have to write one. You just have to see the shape once, because that shape, one command moving the whole batch, is the engine behind the whole day.

Step 7 — Schedule the month and the comment-to-DM mechanic

By now you have the pieces. Step 7 turns one day of work into a month that runs itself: staged across platforms, with a distribution move that quietly out-performs how most people post.

One drop feeds three channels

I don't make different content for different platforms. I make one strong idea, then let each platform do the job it's good at.

- **LinkedIn: conviction.** The take, the teardown, the "here's what I think and why." People read on LinkedIn to think. Give them something to argue with.
- **Instagram — reach.** Same idea, short and punchy. A clip, a hook, a carousel. Your widest net and your lead engine.
- **YouTube: depth.** The long version, the full build on screen. One good video keeps getting found for months, the only piece that works while you sleep a year from now.

Stage the whole month at once

Don't post by hand every day. That's the trap that kills consistency. Lay the month out in one sitting while your head is already in it.

1. **Pick a rhythm you'll actually hold.** A real, doable month: 3 LinkedIn posts a week, 4–5 IG Reels a week, 1 YouTube video a week. Pick numbers you won't quietly abandon by week two.
2. **Drop every piece onto a calendar.** A spreadsheet is fine. One row per piece: date, platform, hook, file, keyword. Seeing the whole month on one screen is the point.

3. **Schedule it, don't babysit it.** Load the posts into a scheduler so they fire on their own. I use **Zernio** because it publishes to all three platforms *and* runs the comment-to-DM mechanic below from the same place: your keys, your accounts, no middleman. Starting out, the native scheduler inside each app is free. Schedule by hand your first month, automate once it works.

The give-first distribution move: keyword in the comments, not links in the post

Here's the part most people get backwards, and it's free.

Do not drop your external link in the post body. Every platform wants to keep people on the platform, so the second you paste a link that sends them away, the algorithm quietly throttles your reach. You're paying for that link in views you'll never see.

Do this instead: tell people to comment a keyword, and you DM them the thing.

*"Want the free playbook? Comment **PLAYBOOK** and I'll send it to you."*

Two reasons it wins:

- **It protects your reach.** No outbound link means the platform isn't punishing the post. It gets shown to more people, so more people see the offer.
- **Comments are the strongest signal there is.** A comment counts for far more than a like. Asking for one doesn't just collect leads, it grows the post while it collects them.

And it's more honest, not less. The person asked. They typed the keyword. You hand them the real thing in the DM, exactly like you said.

How Zernio runs the whole thing for you

Zernio runs this whole move from one place: it publishes the post, watches the comments, and when someone types your keyword it auto-sends the DM with a tappable button to the resource. You set it once per post — *this post, this keyword, this DM* — and it runs itself, on your own accounts, your own keys. Same keyword logic you'd do by hand, at scale, while you sleep.

Reply to every comment, fast

Whether the DM is automated or not, you still show up. Reply to every comment, ideally inside 90 minutes. The first hour or two decides how far a post travels: early replies pull more comments, more comments tell the algorithm the post is alive, and it gets pushed wider right when it counts. Cheapest reach you'll ever buy.

2 SKILLS · YOURS, FREE

zernio-publish + zernio-comment-to-dm

zernio-publish posts to every platform from one request, with platform-native captions, on your own accounts. zernio-comment-to-dm wires the comment-KEYWORD-to-DM automation: set the keyword and the DM once, and every commenter gets the resource automatically, with the button. Both are the real skills — also public at github.com/Enriquemarq1/zernio-library-skills.

```
zernio-publish → one request, all platforms,  
platform-native captions, your accounts.  
zernio-comment-to-dm → post + keyword + DM(button),  
auto-replies the resource to every commenter.
```



Get the Zernio
skills
FREE

Do this now: Open a blank spreadsheet with five columns: **Date · Platform · Hook · File · Keyword**. Fill the next 7 days from the content you already built. For every Instagram row, write one keyword and rewrite the caption to "comment [KEYWORD] and I'll DM it to you," no link in the post. Schedule those 7, post the first today, and reply to every comment within 90 minutes. That's your month started, and the loop that runs it.

The real client run: a month of content from two days of footage

Let me make all of this concrete with a real one.

I run this system for a client, a sports-facility business. Real operation, real audience, no background in content production at all. Their whole job is coaching kids, not making videos.

So I did what I do with every client: I asked for the footage first. Not "make content." Just *capture, with a purpose, for a couple of days*. I handed them a shot list built for exactly what the formats need:

- Kids playing on the field (the energy, the montage fuel)
- Coach interviews (the talking-head spine, the spotlights and the compilation)
- B-roll of players standing on the field (the cutaways)
- Group talks (the culture, the proof of how they actually coach)

Two days of purposeful footage. That's all it took. They filmed across two days, dropped it on me, and the whole system above ran on it: transcribe, mine, reframe, edit, spin into every format, schedule. **A full month of content, produced in a single editing day.** Spotlight clips of the coaches, an advice

compilation across the kids, montages of training, carousels pulled straight from the interview lines, the long-form anchor. All staged across the month, comment-to-DM running underneath.

This is the same content operating system I've been building for the last six months. The same one that has generated about **\$25K in the last two months** of this work, and the same one that turns two days of someone's raw footage into a month they could never have made by hand. Not a different "pro version." The exact system in this playbook.

That's the whole thing — and the honest upgrade

That's it. Not a sample of it. The whole method, the whole stack, and the actual pieces (the skills, the agents, the dynamic workflow) yours to download and keep. Close this PDF right now and never hear from me again, and you've still got everything you need to build a month of content from one footage drop, on your own machine, your own keys.

Here's the part most people selling you something won't say out loud.

By hand, it's a full day. A real one. Wrangling files, watching the work, nudging it, re-rendering the one clip that came out soft, fixing the caption that read wrong. It's doable. I did it for months. But it's a day, every month, and I got tired of giving that day away.

So I did what I always do. I built first, then I taught. I took the day I just handed you, and the tools you just downloaded with it, and I assembled them into one system that runs the whole thing for me. That system is **Nave**.

Nave doesn't know a secret you don't. It runs this exact method: same tools, same skills, same agents, same workflow, your own keys, your own Claude Code subscription, no middleman skimming your bill. The only thing it changes is the clock. The pieces you downloaded today are the parts. Nave is them *assembled, installed on your machine, tuned to your brand, and run for you*. The day becomes a one-time setup, and after that it's about an hour a month. You drop the footage, you say "make my month," you approve the slate.

That's the whole pitch, said straight. Nave isn't the gate in front of the method. The method is right here, free, finished, and the tools are on your disk. Nave is what you reach for when doing it by hand once a month stops being worth the day. If you'd rather keep the day and keep the cash, do that. You're not missing anything. I mean it.

If the hour sounds better than the day, it lives at **nave.infinitxai.com**, and I'm the one who answers when you reply. And if you'd rather I set it up *with* you — your keys placed, a real render verified, your

first month produced together on a call — that's there too.

Either way, hand or machine, I want you walking out with the thing I built all of this to get back for myself in the first place. Your time, and a system that's actually yours, not rented from someone else by the month.

Do this now: Decide which person you are this month. Want to own the day? Block a real one, title it "make my month," and run the steps end to end with the tools you just grabbed. Want to own the hour? Go to **nave.infinityai.com** and grab a setup seat. Pick one, put it on the calendar today, and stop renting your output.

Build first, **Enrique Marq** · @enriquemarq-0 · nave.infinityai.com



Enrique Marq

I build content systems for a living. The content operating system in this playbook has produced a month of content in a single day and ~\$25K in two months of client work, and I run it on real client footage every month. INFINITX partner, creator of Nave. I answer my own DMs, so come build.