THE SPRINT

AI Video Automation

Step-by-Step Setup Guide

OpenClaw · HeyGen · ElevenLabs · Buffer

Build your AI clone. Post every day. Your only job is to approve.

How This System Works

This guide walks you through building a 12-agent autonomous video pipeline inside OpenClaw. Each step gives you the exact setup to perform in OpenClaw and the exact prompt to paste into each agent. Follow the steps in order. By the end, OpenClaw will run every morning, generate videos using your cloned voice and face, and post them while you sleep.

🎯 **What you'll have when this is complete**

Your digital clone creates and publishes videos to TikTok, Instagram Reels, and YouTube Shorts daily
9 agents run automatically on a schedule — you only review the QA package and approve
A self-improving analytics loop updates your content strategy every Monday
Total cost: ~$130–165/month for 30 published videos across 3 platforms

Before You Start — Accounts & API Keys

Add these to OpenClaw before building anything. In OpenClaw go to Settings → Secrets and create each one:

Secret Name	Where to Get It	Plan Required
`HEYGEN_API_KEY`	HeyGen → Profile → API	Studio or above
`ELEVENLABS_API_KEY`	ElevenLabs → Profile → API Keys	Creator or above
`BUFFER_ACCESS_TOKEN`	buffer.com → Developers → OAuth	Essentials or above
`BUFFER_IG_CHANNEL_ID`	Buffer dashboard → Channels	Any
`BUFFER_TIKTOK_CHANNEL_ID`	Buffer dashboard → Channels	Any
`BUFFER_YT_CHANNEL_ID`	Buffer dashboard → Channels	Any
`GOOGLE_SERVICE_ACCOUNT_JSON`	Google Cloud Console → IAM	Free
`SLACK_WEBHOOK_URL`	Slack → App → Incoming Webhooks	Free

⚠️ **Plan requirements matter**

HeyGen free tier only gives stock avatars — you cannot clone yourself without Studio.
ElevenLabs free tier does not support professional voice cloning — Creator tier minimum.
Do not buy any plan until you have completed Step 1 and confirmed your recordings are good.

STEP 01

Record Your Training Assets

Done once. These files are the foundation of your entire clone.

Before OpenClaw can do anything, it needs two source files: a video of your face and a recording of your voice. These are uploaded to HeyGen and ElevenLabs to create your clone. You only ever record these once unless your appearance or voice changes significantly.

Avatar Training Video

- Duration: 10–20 minutes of continuous, uncut recording
- Camera: Any 1080p webcam, DSLR, or phone propped at eye level
- Framing: Chest-up, head centered, filling roughly 60% of the frame
- Lighting: Ring light or softbox pointed at your face from slightly above — no windows behind you
- Background: Plain wall or clean solid backdrop — remove anything distracting
- Movement: Speak naturally but minimize excessive nodding or turning. HeyGen struggles with extreme head angles.
- Audio: Mute fans, HVAC, and notifications. HeyGen only needs the video — audio quality here doesn't matter.
- Content: Just talk. Explain a topic you know. The content does not matter — your visual performance does.

Voice Training Audio

- Duration: 5–15 minutes. Longer produces a noticeably better clone.
- Mic: Same microphone every time — consistency across sessions improves accuracy
- Room: Bedroom closet with clothes on the rails works perfectly as a sound booth
- Content: Read scripts aloud. Vary pace and emotion. Include questions, statements, emphasis. Monotone samples produce monotone clones.
- Zero noise: No fan, keyboard, pets, or HVAC. ElevenLabs is sensitive to ambient sound.

Where to Save the Files

```
// Save your recordings here — OpenClaw reads from these paths
~/content/avatar_training/avatar_v1.mp4
~/content/voice_training/voice_v1.mp3

# If you re-record, version up — never overwrite originals
~/content/avatar_training/avatar_v2.mp4
~/content/voice_training/voice_v2.mp3
```

✅ **You're ready for Step 2 when**

Avatar video is saved and plays cleanly at ~/content/avatar_training/avatar_v1.mp4
Voice audio is saved and sounds clean at ~/content/voice_training/voice_v1.mp3

Both files are backed up to cloud storage — these are irreplaceable

STEP
02

Create the CloneSetupAgent

Runs once. Registers your avatar and voice IDs in OpenClaw.

This is the only agent you run manually. It uploads your training files to HeyGen and ElevenLabs, waits for processing to complete, then stores the returned IDs as secrets. Every other agent in the system reads those IDs — this step is what connects your recordings to the automation.

OPENCLAW SETUP — CLONESETUPAGENT

In OpenClaw, create a new Agent with these settings:

- Name: CloneSetupAgent
- Model: claude-sonnet-4-6
- Run mode: Manual only — this agent should never trigger automatically

Paste this into the system prompt field:

PROMPT — CloneSetupAgent

```
You are CloneSetupAgent. You run exactly once to initialize the video automation system.
Do not run again unless explicitly told to re-initialize.

Your task sequence:

1. Upload ~/content/avatar_training/avatar_v1.mp4 to HeyGen.
   Use the HeyGen create avatar endpoint. Set name: 'main_creator_clone'.
   This is a multipart/form-data POST to https://api.heygen.com/v2/avatar/create
   using the HEYGEN_API_KEY secret in the X-Api-Key header.

2. Poll https://api.heygen.com/v2/avatar/{avatar_id} every 60 seconds.
   Wait until status = 'completed'. Maximum wait: 20 minutes.
   If status = 'failed', report the full error response and stop.

3. Store the avatar_id as OpenClaw secret HEYGEN_AVATAR_ID.

4. Upload ~/content/voice_training/voice_v1.mp3 to ElevenLabs.
   POST to https://api.elevenlabs.io/v1/voices/add
   Header: xi-api-key: {ELEVENLABS_API_KEY}
   Body: multipart/form-data with name='creator_voice_clone' and the audio file.

5. Store the returned voice_id as OpenClaw secret ELEVENLABS_VOICE_ID.

6. Send a test TTS request using the new voice_id with the text:
   'Setup complete. This is a voice verification test.'
   Confirm the audio file is returned and non-empty.

7. Report success: output both stored IDs and confirmed: true.
```

```
  Or report exactly which step failed and why.

Rules:
- Never skip the HeyGen polling loop. The ID is only valid after status = 'completed'.
- If any step fails after 3 retries, stop completely. Do not proceed past the failure.
- Do not store a placeholder or partial ID. Only store confirmed, working IDs.
```

How to Run CloneSetupAgent

1. In OpenClaw, open Agents → CloneSetupAgent
2. Click Run Manually
3. Watch the execution log — HeyGen avatar processing takes 5–20 minutes
4. When complete, go to Settings → Secrets and confirm both HEYGEN_AVATAR_ID and ELEVENLABS_VOICE_ID appear with non-empty values
5. Play the verification audio to confirm the voice sounds like you

🛑 **Do not build any other agents until both IDs are confirmed**

Every agent downstream depends on HEYGEN_AVATAR_ID and ELEVENLABS_VOICE_ID.
If either is missing or incorrect, the entire pipeline will fail silently.
Take the extra 5 minutes here to verify. It saves hours of debugging later.

STEP

03

Set Up the VideoJob Tracker

Every video is a record. Every agent reads and updates it.

Before building any pipeline agents, set up the VideoJob data structure. This is how every agent in the system knows what to work on, what's been done, and what failed. Think of it as the shared whiteboard all 9 agents read and write.

VideoJob Fields

Field	What it holds
`job_id`	Unique ID — format: job_YYYYMMDD_XXXX
`topic`	The subject of this video
`hook`	The opening 1–2 sentence line (written by ScriptWriterAgent)
`script`	Full generated script text
`final_transcript`	Post-generation transcript from Whisper (plain text)
`voice_audio_path`	Path to the generated voice.mp3 file
`avatar_video_path`	Path to the raw HeyGen avatar.mp4
`captioned_video_path`	Path to the finished, captioned final.mp4
`status`	Current pipeline stage (see status flow below)
`platform_targets`	Which platforms to post to: TikTok, IG Reels, YouTube Shorts
`scheduled_time`	When PublisherAgent scheduled the post
`heygen_video_id`	HeyGen's internal video ID — used for polling status

Status Flow

Every agent reads status to find its work and writes status when it finishes. This is the backbone of the entire pipeline:

```
// VideoJob status progression
draft
  → scripted          (ScriptWriterAgent wrote the script)
  → voice_ready       (VoiceAgent generated the audio)
  → avatar_processing (AvatarVideoAgent submitted to HeyGen)
  → avatar_ready      (HeyGen finished, video downloaded)
  → captioned         (CaptionAgent added captions + formatted)
  → qa_passed          (QAAgent checks passed — sent to you for review)
  → approved           (YOU approved it)
  → scheduled          (PublisherAgent scheduled in Buffer)
  → posted

# Failure states — reset to previous passing state to retry
script_too_long  |  voice_failed  |  avatar_failed
avatar_timeout   |  caption_failed  |  qa_failed  |  rejected
```

💡 **How to retry a failed job**

Find the VideoJob record in your database.
Change the status field back to the last passing state.
The next time that agent runs on its schedule, it will pick the job up automatically.
Example: status = 'avatar_failed' → change to 'voice_ready' → AvatarVideoAgent resubmits on next run.

STEP
04

Create the IdeaIntakeAgent

Pulls topics from your queue and creates VideoJob records every morning.

IdeaIntakeAgent runs at 9:00 AM every day. It reads your Google Sheet content queue, creates a VideoJob record for each new topic, and marks the sheet rows as In Progress. This is how topics enter the pipeline.

Set Up Your Google Sheet First

Create a Google Sheet named 'Content Queue'. The first tab needs exactly these columns:

Col	Header	Example	Required?	Notes
A	Topic	How I cut AI costs 90% in 30 days	Yes	Keep under 80 chars
B	Pillar	OpenClaw	Yes	Dropdown: see pillars below
C	Status	Queue	Yes	Queue / In Progress / Done
D	Priority	1	No	1–5, lower = runs first
E	Notes	Focus on Haiku routing trick	No	Optional context for script writer

Add 30–50 topics before enabling the schedule. Content pillars in priority order: OpenClaw automation → Autonomous workforce → Founder ops → Case studies.

OPENCLAW SETUP — IDEAINTAKEAGENT

Create a new Agent in OpenClaw with these settings:

- Name: IdeaIntakeAgent
- Model: claude-haiku-4-5 (this is a read/write task — no deep reasoning needed)
- Schedule: 0 9 * * * (9:00 AM every day)

Paste this into the system prompt field:

PROMPT — IdeaIntakeAgent

```
You are IdeaIntakeAgent. You run every morning at 9:00 AM. Your job takes 2–3 minutes.

Task:
1. Read the Google Sheet at spreadsheet ID {GOOGLE_SHEETS_QUEUE_ID}, tab 'Content Queue'.
   Pull all rows where column C = 'Queue'. Maximum 8 rows per run.
   Sort by column D (Priority) ascending — lower number runs first.

2. For each topic row:
   a. Check if a similar topic has been processed in the last 30 days.
      If a near-duplicate exists, skip this row and log: 'Skipped [topic] — duplicate of [existing topic]'.
   b. If not a duplicate:
      - Generate job_id in format: job_YYYYMMDD_XXXX (XXXX = 4-digit random)
      - Create a VideoJob record: { job_id, topic (col A), pillar (col B), status: 'draft' }
      - Update the sheet row: set column C to 'In Progress'

3. Output a clean summary:
   - Rows read: X
   - Jobs created: X  (list each job_id and topic)
   - Rows skipped: X  (list each with reason)

Rules:
- Never create a VideoJob for a duplicate topic. When in doubt, skip.
- If Google Sheets returns an error, retry once after 30 seconds, then stop and alert.
- Never create more than 8 jobs in a single run.
```

STEP 05 — Create the ScriptWriterAgent

Turns each topic into a high-performing short-form script in your voice.

ScriptWriterAgent runs at 9:15 AM, right after IdeaIntakeAgent. It picks up every VideoJob with status 'draft' and generates a complete script. This is the most important agent in the system — script quality directly determines whether videos get watched.

⚠️ **Use Sonnet here — this is the one agent worth the extra cost**
Script quality directly determines view count, save rate, and followers gained per video. Haiku produces noticeably lower-quality scripts. The $15/month difference in model cost is recouped by a single extra conversion from content to SPRINT membership.

OPENCLAW SETUP — SCRIPTWRITERAGENT

- Name: ScriptWriterAgent
- Model: claude-sonnet-4-6
- Schedule: 15 9 * * * (9:15 AM every day)

Paste this into the system prompt field. The VOICE section is where you personalize — replace the placeholder line with real transcripts from your best-performing videos:

PROMPT — ScriptWriterAgent

```
You are ScriptWriterAgent. You write short-form video scripts for TikTok,
Instagram Reels, and YouTube Shorts.

TASK:
1. Query the database for all VideoJob records with status = 'draft'.
2. For each job, generate a complete script using the format below.
3. Update the VideoJob record:
   - script = your full output
   - hook = the opening line only
   - status = 'scripted'
4. Log: job_id, topic, word count, estimated spoken duration in seconds.

VOICE AND STYLE:
Write exactly like this creator speaks. Key characteristics:
- Direct and confident. Never hedging. 'This works' not 'This might work'.
- Short sentences. Two to eight words each. Occasional longer sentence for rhythm.
- Use 'you' constantly. Every sentence should feel personal to the viewer.
- Numbers over adjectives. '90% cheaper' beats 'significantly cheaper' every time.
- No corporate language. Never: leverage, synergy, utilize, game-changer, incredible.
- Confident teacher energy — authoritative but not yelling.

>>> REPLACE THIS LINE with 2–3 full transcripts from your best-performing videos.
```

```
    The more examples you give, the less the output sounds like generic AI. <<<

SCRIPT FORMAT — follow exactly:

[HOOK] — 8 to 15 words. Bold claim or surprising fact. Make them stop scrolling.
Good hook examples:
  'I cut my AI costs from $3,000 a month to $100. Here is exactly how.'
  'Most developers are routing to the wrong Claude model. This is the fix.'
  'OpenClaw ran my entire business overnight. You just do not know this yet.'

[PROBLEM] — 1 to 2 sentences. Name the pain your viewer already feels.
Do not explain it. Just name it. They already know it.

[FRAMEWORK] — 3 steps or 3 points, labeled Step 1 / Step 2 / Step 3.
Each step: 1 to 2 sentences max. If they cannot remember it, it is too long.

[EXAMPLE] — 2 to 3 sentences. One specific, real example with numbers and tool names.
Example: 'I routed all intake agents to Haiku. That one change saves $40 a day.'

[CTA] — Exactly one sentence. One action only.
Options: comment a keyword, follow for more, or save this.
Example: 'Comment ROUTING and I will send you the full config.'

HARD RULES:
- Total word count: 90 to 130 words. Count them. Stay in range.
- Never start with 'In this video'.
- Never end with 'Like and subscribe'.
- One idea per video. If the script has two main points, cut one.
- Read the output aloud mentally. If it sounds like an article, rewrite it.
```

STEP

06

Create the VoiceAgent

Converts scripts to audio using your ElevenLabs cloned voice.

VoiceAgent runs at 9:30 AM. It picks up every VideoJob with status 'scripted', calls ElevenLabs with your cloned voice ID, and saves the audio file to the job folder. This is a mechanical task — Haiku handles it well at a fraction of the cost.

OPENCLAW SETUP — VOICEAGENT

- Name: VoiceAgent
- Model: claude-haiku-4-5
- Schedule: 30 9 * * * (9:30 AM every day)

PROMPT — VoiceAgent

```
You are VoiceAgent. You convert approved scripts to audio files using ElevenLabs.

TASK:
1. Query the database for all VideoJob records with status = 'scripted'.
2. For each job:
   a. Retrieve the script field from the VideoJob record.
   b. POST to https://api.elevenlabs.io/v1/text-to-speech/{ELEVENLABS_VOICE_ID}
      Header: xi-api-key: {ELEVENLABS_API_KEY}
      Body JSON:
        text: the full script
        model_id: eleven_multilingual_v2
        voice_settings:
          stability: 0.55
          similarity_boost: 0.80
          style: 0
          use_speaker_boost: true
   c. Save the returned binary audio to: ~/content/jobs/{job_id}/voice.mp3
   d. Verify the file exists and is larger than 10KB.
   e. Estimate duration: MP3 at 128kbps ≈ 1MB per minute.
      If estimated duration exceeds 75 seconds, set status = 'script_too_long' and skip.
   f. Otherwise: update VideoJob voice_audio_path and status = 'voice_ready'.

3. Log each job: job_id, file size in KB, estimated duration, status set.

Rules:
- If ELEVENLABS_VOICE_ID secret is empty, stop immediately and alert.
- If ElevenLabs returns HTTP 429, wait 30 seconds and retry once.
- Process jobs sequentially — not in parallel. ElevenLabs limits concurrent requests.
- Never proceed with a file under 10KB. Something went wrong — mark voice_failed.
```

🎛️ **ElevenLabs voice settings explained**

stability 0.55 — lower = more expressive, higher = more robotic. Start at 0.55 and adjust by ear.
similarity_boost 0.80 — how closely the output matches your recorded clone. Keep at 0.80.
style 0 — leave at zero. Non-zero values add unnatural exaggeration.
use_speaker_boost true — enhances likeness. Always keep enabled.

STEP
07

Create the AvatarVideoAgent

Submits voice audio to HeyGen and retrieves your talking-head video.

AvatarVideoAgent runs twice a day. The first run at 9:45 AM submits new jobs to HeyGen. The second run at 10:30 AM checks which jobs have finished processing and downloads the results. HeyGen takes 1–5 minutes per video — running the agent twice ensures completed jobs don't wait until the next day.

OPENCLAW SETUP — AVATARVIDEOAGENT

- Name: AvatarVideoAgent
- Model: claude-haiku-4-5
- Schedule: Two schedules — 45 9 * * * (submit run) and 30 10 * * * (retrieve run)

PROMPT — AvatarVideoAgent

```
You are AvatarVideoAgent. You run twice daily — once to submit jobs, once to retrieve them.

SUBMIT PHASE (9:45 AM run):
1. Query the database for all VideoJob records with status = 'voice_ready'.
2. For each job:
   a. POST to https://api.heygen.com/v2/video/generate
      Header: X-Api-Key: {HEYGEN_API_KEY}
      Body JSON:
        video_inputs: [{
          character: { type: 'avatar', avatar_id: '{HEYGEN_AVATAR_ID}' },
          voice: { type: 'audio', audio_url: '{voice_audio_path}' },
          background: { type: 'preset', value: '{your_background_id}' }
        }]
        aspect_ratio: '9:16'
   b. Store the returned video_id in the VideoJob heygen_video_id field.
   c. Set status = 'avatar_processing'.

RETRIEVE PHASE (10:30 AM run):
1. Query the database for all VideoJob records with status = 'avatar_processing'.
2. For each job:
   a. GET https://api.heygen.com/v2/video/{heygen_video_id}
      Header: X-Api-Key: {HEYGEN_API_KEY}
   b. If status = 'completed':
      - Download video_url to ~/content/jobs/{job_id}/avatar.mp4
      - Update VideoJob: avatar_video_path = path, status = 'avatar_ready'
   c. If status = 'failed': set status = 'avatar_failed', log the full error.
   d. If the job has been in 'avatar_processing' for more than 25 minutes:
      Set status = 'avatar_timeout'.
   e. Otherwise (still processing): leave status unchanged.
```

```
Rules:
- Never re-submit a job that already has a heygen_video_id. Check before submitting.
- Download videos immediately on completion — do not defer.
- Replace '{your_background_id}' with your actual HeyGen background preset ID.
  Use the same background for every video — consistency builds brand recognition.
```

STEP **08**

Create the CaptionAgent

Transcribes, captions, and formats the final video for posting.

CaptionAgent runs at 10:45 AM. It picks up jobs with avatar videos ready, generates an SRT caption file using Whisper, burns the captions into the video with ffmpeg, and produces the final 9:16 formatted file ready for QA.

OPENCLAW SETUP — CAPTIONAGENT

- Name: CaptionAgent
- Model: claude-haiku-4-5
- Schedule: 45 10 * * * (10:45 AM every day)

PROMPT — CaptionAgent

```
You are CaptionAgent. You add captions to avatar videos and produce the final output file.

TASK:
1. Query the database for all VideoJob records with status = 'avatar_ready'.
2. For each job:
  a. Transcribe ~/content/jobs/{job_id}/voice.mp3 using Whisper.
    Request SRT format output.
    Save the SRT to: ~/content/jobs/{job_id}/captions.srt

  b. Review and correct the SRT file:
    - These brand names are commonly mis-transcribed. Apply these corrections:
      'Open Claw' or 'OpenCloth' → OpenClaw
      'Hey Gen' or 'Hagen' → HeyGen
      'Eleven Labs' → ElevenLabs
      'Haiku' mispellings → Haiku
      'Sonnet' mispellings → Sonnet
    - Split any caption segment longer than 30 characters into two segments.

  c. Run this ffmpeg command to burn captions and format the video:
    ffmpeg -i {avatar_video_path}
      -vf "subtitles={srt_path}:force_style='FontName=Arial,FontSize=20,
      Bold=1,PrimaryColour=&HFFFFFF,OutlineColour=&H000000,Outline=2,
      Alignment=2'"
      -ar 44100 ~/content/jobs/{job_id}/final.mp4

  d. Verify ~/content/jobs/{job_id}/final.mp4 exists and is at least 2MB.
    If under 2MB, set status = 'caption_failed' and log the ffmpeg error output.

  e. Update VideoJob:
    - captioned_video_path = ~/content/jobs/{job_id}/final.mp4
    - final_transcript = plain text extracted from the SRT (no timestamps)
```

```
      - status = 'captioned'

Rules:
- Never pass a job to QA with a missing or corrupt output file.
- If ffmpeg fails, log the full error output — not just 'failed'.
- Verify ffmpeg is installed: run 'ffmpeg -version' before processing the first job.
```

STEP
09

Create the QAAgent

Automated checks + Slack review package sent to you.

QAAgent runs at 11:00 AM. It checks every captioned video against a set of quality rules, then sends you a Slack notification with everything you need to approve or reject. Your only job at this stage is to watch the video and make a decision.

OPENCLAW SETUP — QAAGENT

- Name: QAAgent
- Model: claude-sonnet-4-6 (judgment calls need better reasoning)
- Schedule: 0 11 * * * (11:00 AM every day)

PROMPT — QAAgent

```
You are QAAgent. You are the last automated checkpoint before a human reviews content.
Your job is to catch problems so the human only sees content that is ready.

TASK:
1. Query the database for all VideoJob records with status = 'captioned'.
2. For each job, run ALL checks:

  TECHNICAL CHECKS:
  a. File exists: ~/content/jobs/{job_id}/final.mp4 must exist.
  b. File size: must be between 2MB and 200MB.
  c. Transcript match: compare script field vs. final_transcript.
     They must match at least 90%. Flag if below threshold.

  CONTENT CHECKS:
  d. Word count: final_transcript must be 80 to 145 words.
  e. CTA present: script must end with a clear call to action.
  f. Banned phrases — flag if any appear in script or transcript:
     'guaranteed to', 'get rich', 'make money fast',
     'this is not financial advice', 'passive income with no work'

3. Determine result:
  PASS: all checks pass → status = 'qa_passed'
  FLAG: soft issues detected → status = 'qa_flagged', list each issue
  FAIL: critical issue → status = 'qa_failed' (do not send for review)

4. For PASS and FLAG jobs only, send a Slack notification to {SLACK_WEBHOOK_URL}:
  Include:
  - Job ID and topic
  - Video file path
  - Full script text
  - QA result + any flags
  - Suggested caption:
    First 125 characters of the script + '...'
  - Suggested hashtags based on pillar:
    OpenClaw: #openclaw #aiagents #automation #vibecoding #aitools
    Autonomous workforce: #aiworkforce #multiagent #aiops #futureofwork
    Founder ops: #solofounder #saasfounder #buildinginsaas #founderlife
    Case studies: #airesults #realresults #automationwin #buildinpublic
    Always append: #sprint #mattganzak

5. Log: total reviewed, pass/flag/fail counts, issues per job.

Rules:
- Never mark qa_passed if the video file is missing or under 2MB.
- FLAG = you decide. FAIL = blocked. Never send a FAIL job for review.
```

✅ Your only job in the whole system

When QAAgent sends you the Slack notification: open the video file, read the script, check the flags.
If you want to post it: update the VideoJob status to 'approved'.
If you want to reject it: update status to 'rejected' and add a note.
That's it. Everything else is automated.

STEP
10

Create the PublisherAgent

Schedules approved videos across all three platforms automatically.

PublisherAgent runs at 12:00 PM every day. It picks up every job you've marked as 'approved', writes a platform-appropriate caption with hashtags, determines the next available posting slot for each platform, and schedules everything through Buffer.

OPENCLAW SETUP — PUBLISHERAGENT

- Name: PublisherAgent
- Model: claude-haiku-4-5
- Schedule: 0 12 * * * (12:00 PM every day)

PROMPT — PublisherAgent

```
You are PublisherAgent. You schedule approved videos for posting via Buffer.

TASK:
1. Query the database for all VideoJob records with status = 'approved'.
2. For each job:
  a. Write a caption using this structure:
    - Line 1: the hook field from VideoJob (verbatim)
    - Lines 2–3: 1–2 sentences summarizing the framework from the script
    - Final line: the CTA from the script
    - Total length: 150–300 characters for TikTok/Instagram, 500 max for YouTube

  b. Build hashtag string from content pillar:
    OpenClaw: #openclaw #aiagents #automation #vibecoding #aitools
    Autonomous workforce: #aiworkforce #multiagent #aiops #futureofwork
    Founder ops: #solofounder #saasfounder #buildinginsaas #founderlife
    Case studies: #airesults #realresults #automationwin #buildinpublic
    Always append: #sprint #mattganzak

  c. Determine next available posting slot per platform.
    Posting windows (Eastern Time):
      TikTok: 7:30 AM, 12:00 PM, 7:00 PM
      Instagram Reels: 9:30 AM, 1:00 PM
      YouTube Shorts: 1:00 PM
    Never schedule two videos on the same platform within 4 hours.
    If no slots available today, schedule for tomorrow's earliest slot.

  d. For each platform, POST to https://api.bufferapp.com/1/updates/create.json
    Authorization: Bearer {BUFFER_ACCESS_TOKEN}
    Body: profile_ids, text (caption + hashtags), media[video], scheduled_at (ISO 8601 UTC)

  e. Update VideoJob: status = 'scheduled', scheduled_time, platform_targets.
```

```
3. Log: jobs scheduled, platforms per job, scheduled datetimes.

Rules:
- If Buffer errors on one platform, still schedule the others. Log the failed platform.
- Never schedule the same job twice on the same platform.
- scheduled_at must be in ISO 8601 UTC format: 2026-03-01T14:00:00Z
- scheduled_at must be at least 10 minutes in the future from time of API call.
```

STEP
11

Create the AnalyticsAgent

Pulls weekly performance data and makes the system smarter.

AnalyticsAgent runs every Monday at 8:00 AM. It pulls metrics from TikTok and Instagram, scores every video using a weighted formula, identifies patterns in what worked and what didn't, adds 5 new topics to your queue based on winners, and sends you a Slack report before your week starts.

OPENCLAW SETUP — ANALYTICSAGENT

- Name: AnalyticsAgent
- Model: claude-sonnet-4-6 (pattern recognition and insight generation need better reasoning)
- Schedule: 0 8 * * 1 (8:00 AM every Monday)

PROMPT — AnalyticsAgent

```
You are AnalyticsAgent. Every Monday at 8:00 AM you analyze last week's content
performance and update the system's content priorities for the week ahead.

TASK:
1. Query the database for VideoJob records posted in the last 7 days.

2. For each posted video, pull metrics from TikTok and Instagram APIs:
  views, watch_time_seconds, average_retention_pct,
  shares, saves, comments, follows_from_video.
  Store all metrics against the VideoJob record.

3. Compute composite score for each video:
  score = (saves × 5) + (shares × 3) + (follows_from_video × 4)
        + (average_retention_pct × 2) + (views × 0.001)

4. Identify patterns from the top 3 and bottom 3 scoring videos:
  - What did the top hooks have in common? (numbers, questions, claims?)
  - Which content pillar drove the most engagement?
  - Did shorter or longer videos perform better this week?
  - Which CTAs drove the most keyword comments?

5. Generate 5 new topic ideas based on:
  - The best-performing pillar this week
  - Hook patterns from top performers
  - Comments on top videos that suggest follow-up topics
  Add these 5 topics to the Google Sheet with Priority = 1.

6. Send a Slack report to {SLACK_WEBHOOK_URL} with these sections:
  WEEKLY SNAPSHOT: videos posted, total views, total new followers
  TOP PERFORMER: topic, views, save rate, follow rate, why it worked
  HOOK ANALYSIS: what the top hooks had in common (specific, not vague)
```

```
   PILLAR RANKING: all 4 pillars ranked by average composite score
   AVOID NEXT WEEK: specific patterns from bottom performers
   NEW TOPICS ADDED: the 5 topics added and the data reason for each

Rules:
- Exclude videos live less than 48 hours from pattern analysis. Not enough data.
- Be specific in insights. 'Hooks with a dollar amount got 2× more saves' is useful.
  'Good hooks performed well' is useless. Never write a useless insight.
- If a platform API fails, note it and report what you could pull.
```

STEP
12

Enable the Schedule and Launch

Wire all agents to a daily schedule and go live.

All 9 agents are built. Now you connect them to a live daily schedule in OpenClaw and run your first full test before flipping the switch permanently.

The Full Daily Schedule

Time (ET)	Agent	What it does
9:00 AM	`IdeaIntakeAgent`	Pulls topics from Google Sheet, creates VideoJob records
9:15 AM	`ScriptWriterAgent`	Generates scripts for all 'draft' jobs
9:30 AM	`VoiceAgent`	Converts scripts to voice audio via ElevenLabs
9:45 AM	`AvatarVideoAgent`	Submits audio to HeyGen (submit phase)
10:30 AM	`AvatarVideoAgent`	Downloads completed HeyGen videos (retrieve phase)
10:45 AM	`CaptionAgent`	Adds captions, exports final.mp4
11:00 AM	`QAAgent`	Runs checks, sends Slack review package to you
12:00 PM	`PublisherAgent`	Schedules approved videos via Buffer
Mon 8 AM	`AnalyticsAgent`	Weekly performance report + updates content queue

First Week Launch Sequence

Do not enable the full schedule on day one. Validate each layer first:

Day 1	Verify every API key works. Make a raw test call to HeyGen, ElevenLabs, and Buffer from outside of OpenClaw. Fix authentication errors before building anything.
Day 2	Run CloneSetupAgent manually. Confirm both IDs stored. Listen to the verification audio. If the voice sounds off, adjust ElevenLabs settings or re-record.
Day 3	Trigger each agent manually in sequence for one test job. Monitor every step in the logs. Inspect the final.mp4. This is your debugging day — expect 2–3 issues.
Day 4	Fix what broke on Day 3. Run a second test job end-to-end. This one should complete cleanly.
Day 5	Enable the full schedule. Add 20+ topics to your Google Sheet. Watch the 9:00 AM run in the OpenClaw logs in real-time.

Days 6–7	Let it run. Review the QA packages that come in. Approve the best ones. Watch for any agent failures in the logs.
Week 2+	Read Monday's analytics report. Adjust ScriptWriterAgent voice prompt based on what performed. The system is now fully self-running.

Final Setup Checklist

Work through this before enabling the live schedule:

Accounts & Secrets

- ☐ HEYGEN_API_KEY stored in OpenClaw Secrets (HeyGen Studio plan confirmed)
- ☐ ELEVENLABS_API_KEY stored in OpenClaw Secrets (Creator plan confirmed)
- ☐ BUFFER_ACCESS_TOKEN stored (TikTok, Instagram, and YouTube channels connected)
- ☐ BUFFER_IG_CHANNEL_ID, BUFFER_TIKTOK_CHANNEL_ID, BUFFER_YT_CHANNEL_ID stored
- ☐ GOOGLE_SERVICE_ACCOUNT_JSON stored (service account has Editor access to your sheet)
- ☐ SLACK_WEBHOOK_URL stored and tested (send a test message to confirm it works)

Recordings

- ☐ Avatar training video saved at ~/content/avatar_training/avatar_v1.mp4 (10–20 min, 1080p)
- ☐ Voice training audio saved at ~/content/voice_training/voice_v1.mp3 (5–15 min, clean audio)
- ☐ Both files backed up to cloud storage

CloneSetupAgent

- ☐ CloneSetupAgent run manually and completed successfully
- ☐ HEYGEN_AVATAR_ID stored and non-empty in Secrets
- ☐ ELEVENLABS_VOICE_ID stored and non-empty in Secrets
- ☐ Verification audio played back — voice sounds natural

All 9 Agents Built

- ☐ CloneSetupAgent — manual only
- ☐ IdeaIntakeAgent — schedule: 0 9 * * *
- ☐ ScriptWriterAgent — schedule: 15 9 * * * (voice section personalized with your transcripts)
- ☐ VoiceAgent — schedule: 30 9 * * *
- ☐ AvatarVideoAgent — two schedules: 45 9 * * * and 30 10 * * *
- ☐ CaptionAgent — schedule: 45 10 * * *
- ☐ QAAgent — schedule: 0 11 * * *
- ☐ PublisherAgent — schedule: 0 12 * * *

- ☐ AnalyticsAgent — schedule: 0 8 * * 1

Test Run

- ☐ Single test job run manually end-to-end — all statuses progressed correctly
- ☐ final.mp4 reviewed — video plays, captions are accurate, duration is correct
- ☐ QA Slack notification received and readable
- ☐ Test job approved and scheduled in Buffer — post appeared in Buffer dashboard
- ☐ Google Sheet content queue has 30+ topics ready

Questions? Post in THE SPRINT Community.

@mattganzak on TikTok and Instagram

Tag your first automated post. I'll repost it.