

YouTube Autopilot

Auto-cross-post every video to YouTube with AI-optimized metadata. One prompt builds it.

Built by [@brycenwood.ai](#) · October 2026

This is the system that changed how I think about content. You're already filming videos. You're already posting on Instagram and TikTok. But if you're not on YouTube, you're leaving the biggest search engine in the world on the table.

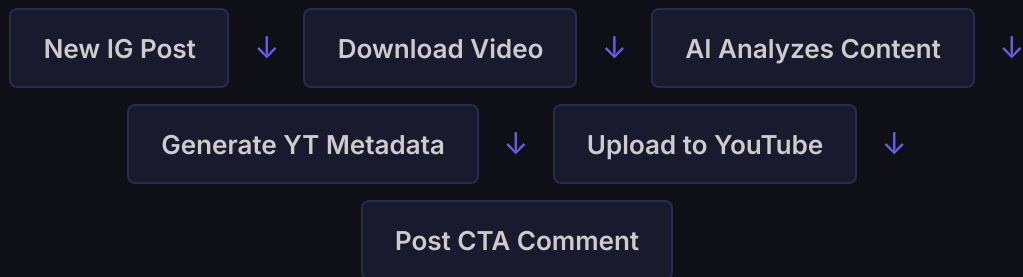
The problem is nobody wants to manually upload to another platform every day. And if you just paste your Instagram caption onto YouTube, it dies. YouTube's algorithm is search-based. Instagram's is engagement-based. Same caption doesn't work on both.

This system solves both problems. It watches your Instagram, grabs every new video, uses AI to generate a completely new title, description, and tags optimized for YouTube search, and uploads it as a Short. Automatically. You set it up once and never think about YouTube again.

OVERVIEW

What You're Building

Two Python scripts that work together. The first watches your Instagram via the official API, downloads new videos, and generates YouTube-optimized metadata using Claude. The second uploads them to your YouTube channel with the optimized titles, descriptions, and tags, and drops a CTA comment on every video.



WHY THIS IS DIFFERENT FROM REPURPOSE.IO OR MANUAL CROSS-POSTING

Apps like Repurpose.io just mirror your video with the same caption. That's not cross-posting, that's copy-pasting. YouTube's algorithm needs searchable titles, keyword-rich descriptions, and relevant tags to surface your content. This system generates all of that automatically using AI that actually analyzes your video content and researches what's ranking on YouTube for that topic.

BEFORE YOU START

What You Need

- **Claude Code** installed (the CLI tool)
- **An Instagram business or creator account** with access to the Instagram Graph API (Meta developer account, free)
- **A YouTube channel** (free, takes 2 minutes to create if you don't have one)
- **A Google Cloud project** with the YouTube Data API enabled (free tier covers everything)
- **An Anthropic API key** for Claude (usage-based, costs less than \$0.01 per video)

TIME + COST

Setup: about 1-2 hours including API credentials. Ongoing cost: under \$1/month for the AI metadata generation. The YouTube API and Instagram API are both free. Once it's running, it takes zero time because it's fully automated.

STEP 1

Set Up YouTube API Access

You need a Google Cloud project with the YouTube Data API enabled. This gives your script permission to upload videos and post comments on your channel.

PASTE THIS INTO CLAUDE CODE

I need to set up YouTube API access so I can upload videos programmatically. Walk me through:

1. Creating a Google Cloud project (or using an existing one)
2. Enabling the YouTube Data API v3
3. Creating OAuth 2.0 credentials (Desktop app type)
4. Downloading the credentials.json file
5. Where to save it in my project

I want to upload YouTube Shorts and post comments on my own channel. I don't need to access anyone else's channel.

THE OAUTH DANCE

The first time you run the upload script, it will open a browser window asking you to log into Google and authorize the app. Pick the correct YouTube channel on the consent screen. After that, it saves a token and never asks again (it auto-refreshes). If you have multiple YouTube channels, make sure you select the right one.

STEP 2

Set Up Instagram Graph API Access

Your script needs to read your Instagram posts to know when you've posted new videos. This uses Meta's official Graph API.

PASTE THIS INTO CLAUDE CODE

I need to set up Instagram Graph API access to read my posts programmatically. Walk me through:

1. Creating a Meta developer account and app (if I don't have one)
2. Adding Instagram Graph API to the app
3. Generating a long-lived access token for my Instagram account

4. Testing the token to make sure it can read my media

My Instagram handle is [YOUR HANDLE]. I only need to read my own posts, not manage other accounts.

Save the config to a JSON file with the access token and my Instagram user ID.

STEP 3

Build the Cross-Post System

This is the main prompt. It builds the entire system: the Instagram watcher, the AI metadata generator, and the YouTube uploader. One prompt, two scripts, fully automated.

PASTE THIS INTO CLAUDE CODE

I want to build a system that automatically cross-posts my Instagram Reels to YouTube Shorts with AI-optimized metadata. Build me two Python scripts:

SCRIPT 1: Instagram to Queue

- Check my Instagram account for new video posts using the Graph API
- Download any new videos it finds
- For each video, read my Instagram caption and use the Anthropic API (Claude) to generate:
 - A YouTube-optimized title (searchable, keyword-rich, under 80 characters, front-load the main keyword)
 - A YouTube description rewritten from scratch for search (NOT a copy of the IG caption). Include relevant keywords, a hook, key takeaways, and a CTA section at the bottom with links to my playbook and social profiles
 - 15-20 YouTube tags mixing exact-match search phrases, related topics, and broad category tags
- Save everything to a JSON queue file
- Track which posts have already been processed so it skips duplicates on re-run

SCRIPT 2: Queue to YouTube

- Read the JSON queue
- Upload each video as a YouTube Short (public, category: Education)
- Post a pinned comment with a CTA on each upload
- Handle YouTube quota limits gracefully (pause if exceeded)
- Save the YouTube video ID back to the queue so we know what's

uploaded

Both scripts should be re-runnable and safe to call from a cron job. Add a `--dry-run` flag to both. The Instagram script should have a `--backfill` flag to reprocess everything.

My Anthropic API key is in my `.env` file as `ANTHROPIC_API_KEY`.
My Instagram config is at `_config/instagram.json`.
My YouTube credentials are at `credentials.json` with token at `_config/youtube_token.json`.

STEP 4

Automate It

The scripts work manually, but the whole point is to never think about YouTube again. Set up a cron job (Linux) or launchd agent (Mac) to run them on a schedule.

PASTE THIS INTO CLAUDE CODE

I want to automate my YouTube cross-posting system so it runs every 6 hours without me doing anything. Set up:

1. A scheduled task that runs the Instagram-to-queue script every 6 hours
2. Immediately after, runs the YouTube uploader script
3. Logs output to a file so I can check what happened
4. Doesn't run if the previous run is still going

I'm on [Mac/Linux]. Use launchd for Mac or cron for Linux. Make sure it runs even if I'm not logged in.

MY RESULTS

What Happened After I Shipped This

- **5,000+ views on YouTube** across 12 videos without ever opening YouTube to upload a single one
- **Every 6 hours** the system checks for new Instagram posts and uploads them automatically

- **Each video gets a unique YouTube-optimized title** instead of a pasted Instagram caption
- **Total time spent on YouTube after setup:** zero
- **Setup time:** about 2 hours including API credentials
- **Ongoing cost:** under \$1/month for the AI metadata generation

REAL TALK

The biggest unlock here isn't the automation. It's that YouTube is a search engine, not a feed. Your Instagram videos have a 24-48 hour shelf life. YouTube Shorts can get discovered months later through search. Every video you post is a long-term asset on YouTube and a short-term play on Instagram. This system makes sure you get both without any extra work.

TIPS

Things I Learned

1. **YouTube titles and Instagram captions are different languages.** Instagram captions are conversational ("Comment MIRROR and I'll DM you"). YouTube titles are searchable ("How to Make Your Website Readable by AI"). The AI rewrite handles this but understand why it matters.
2. **YouTube tags have a 500-character total limit.** Don't generate 30 long-tail tags. 15-20 concise tags is the sweet spot.
3. **The YouTube Data API has a daily quota.** You get 10,000 units/day for free. Each upload costs ~1,600 units. That's about 6 uploads per day, which is more than enough for cross-posting.
4. **Post a CTA comment on every video.** YouTube comments with links actually work because YouTube doesn't suppress them like Instagram does. Use this for your playbook or website link.
5. **Don't delete and re-upload.** If you want to update metadata on an existing video, use the API to update in place. This preserves your view count, comments, and watch history.
6. **YouTube Shorts get discovered through search for months.** Unlike Instagram where a post dies in 48 hours, YouTube Shorts surface in search results indefinitely. Every video is a long-term asset.

That is the whole system.

Zero ad spend. Zero coding experience. The other playbooks
live at brycenwood.com/playbook, one download each.

Built by Brycen Wood · [@brycenwood.ai](https://twitter.com/brycenwood.ai) · brycenwood.com