

BEGINNER GUIDE · NO EXPERIENCE NEEDED

Your Own YouTube Research Agent

Set up an AI assistant that reads any YouTube video for you — the full transcript, chapters, and key details — on command. From a blank computer to a working agent, step by step.

Built for a total beginner. If you can copy-and-paste and follow a recipe, you can do this in about an hour.

What is this, in plain English?

You're going to rent a small always-on computer on the internet (a "VPS"), put an AI assistant on it (**Claude Code** — the same AI as the Claude app, but it can run tools and commands), and give it one special ability (a "**skill**"): pulling the entire written transcript and details out of any YouTube video in seconds. Instead of watching a 30-minute video, you paste the link and get the whole thing as text you can search, summarize, or feed to the AI.

Note: This is the *efficient* version. The old way used a "swarm" of 10 AI agents and cost a fortune in usage. This single skill does the same job for roughly **30× less** by trying the cheapest method first and only escalating if YouTube blocks it.



The shopping list (what you'll sign up for)

- ☐ **A VPS** — a small Linux server. ~\$6/month (DigitalOcean, Hetzner, or Vultr).
- ☐ **A Claude account** — either a Claude **Pro/Max subscription** (~\$20+/mo, easiest) *or* an Anthropic **API key** (pay-per-use).
- ☐ **A Bright Data account** — the service that reliably fetches YouTube transcripts. Pay-as-you-go, pennies per video.
- ☐ **(Optional) A Webshare account** — free tier, used as the cheap first attempt.

Total to get started: about \$6–\$26/month plus a few cents per video.

Part 1 — Get your server (the VPS)

1 Create the server

Go to **DigitalOcean.com** (or Hetzner / Vultr). Create an account, then create a new "Droplet" / "Server" with these settings:

- **Operating System:** Ubuntu 24.04 (or 22.04)
- **Size:** the basic ~\$6/month option (1 GB RAM is fine to start; 2 GB is smoother)
- **Authentication:** set a password (simplest) or an SSH key

When it finishes, the provider shows you an **IP address** (like `203.0.113.45`). Write it down.

2 Connect to it

On Windows: open **PowerShell** (search for it in the Start menu). **On Mac:** open **Terminal**. Then type this, using your server's IP:

```
ssh root@YOUR_SERVER_IP
```

Type `yes` if asked to trust it, then enter the password. You're now "inside" your server — anything you type runs on it.

Part 2 — Install the AI (Claude Code)

3 Install the basics

Copy-paste these lines one block at a time (press Enter after each):

```
# update the server + install Node.js and Python
apt update && apt install -y nodejs npm python3 python3-pip git curl
```

4 Install Claude Code

```
npm install -g @anthropic-ai/claude-code
```

Now start it by typing `claude`. The first time, it will ask you to log in:

- If you have a **Claude Pro/Max subscription** → choose the login option and follow the link.
- If you have an **Anthropic API key** → paste it when asked (get one at console.anthropic.com).

Once you see the Claude prompt, it works. Type `/exit` for now — we have one more thing to add.

Part 3 — Add the YouTube skill

What's a "skill"? A skill is just a folder of code that gives Claude a new ability. They live in a folder called `~/.claude/skills/`. You drop a skill folder in there, and Claude can use it. We're adding the `yt_research` skill.

5 Put the skill in place

You were given a file `yt_research_skill.zip`. Upload it to the server (drag-and-drop in your provider's web console, or use `scp`), then:

```
mkdir -p ~/.claude/skills
cd ~/.claude/skills
unzip ~/yt_research_skill.zip # creates the yt_research folder
```

6 Install what the skill needs

```
pip3 install youtube-transcript-api playwright requests
python3 -m playwright install chromium
python3 -m playwright install-deps
```

(That last one installs the headless browser the skill uses as a backup method.)

Part 4 — The API keys

The skill tries several methods to fetch a transcript, fastest-and-cheapest first. It reads its keys from one small file. Create it:

```
mkdir -p /root/surflines
nano /root/surflines/.env
```

Paste these lines (with **your own** keys — see the table), then save with `Ctrl+O`, `Enter`, `Ctrl+X`:

```
BRIGHTDATA_API_KEY=your_bright_data_key_here
WEBSHARE_API_TOKEN=your_webshare_token_here
```

Key	Where to get it	Cost	Why
BRIGHTDATA_API_KEY	brightdata.com → sign up → API settings	Pay-as-you-go, ~pennies/video	The reliable one. Works even when YouTube blocks the server.
WEBSHARE_API_TOKEN	webshare.io → free plan → API	Free tier	The cheap first attempt (may get blocked; that's OK, it falls back to Bright Data).

Keep these keys private. They're like passwords for paid services. Never post them publicly or put them in a shared document.

Part 5 — Use it

Option A — the simple way (just ask Claude)

Type `claude` to start the AI, then just say:

```
get this youtube video: https://www.youtube.com/watch?v=VIDEO_ID
```

Claude recognizes the link, runs the skill, and hands you the transcript + summary.

Option B — run the skill directly

```
python3 ~/.claude/skills/yt_research/scrape.py --url "https://www.youtube.com/watch?v=VIDEO_ID"
```

The result is saved to a folder: `~/wc/outreach/yt_research/<video_id>/` — open `transcript.md` to read the whole video as text.

That's it. You now have an AI on an always-on server that can read any YouTube video on command. Ask Claude to "summarize it," "pull the 5 key points," or "list every tool mentioned" — it already has the transcript.

Costs & troubleshooting

Thing	Roughly
VPS	\$6/month
Claude (subscription route)	\$20+/month · or API pay-per-use
Bright Data per video	a few cents

- **"It says blocked / no transcript."** Run it once more — the skill rotates methods and the Bright Data layer almost always succeeds on the retry.
- **Different folder paths?** This skill was built to run as the `root` user with its key file at `/root/surflines/.env`. If you set it up somewhere else, open `scrape.py` and change the `ENV_PATH` line near the top to wherever your `.env` lives.
- **Playwright errors?** Re-run `python3 -m playwright install-deps`.

Cheat sheet — the whole thing in 6 commands

```
apt update && apt install -y nodejs npm python3 python3-pip git curl unzip
npm install -g @anthropic-ai/claude-code
mkdir -p ~/.claude/skills && cd ~/.claude/skills && unzip ~/yt_research_skill.zip
pip3 install youtube-transcript-api playwright requests && python3 -m playwright
install chromium
mkdir -p /root/surflines && nano /root/surflines/.env # paste your 2 keys
claude # then: "get this youtube video: <link>"
```

Made by Wave for a friend. Questions? The person who gave you this can help — they have the same setup running.