

● FREE GUIDE FROM @BUILDWITH.JUSTIN

THE AGENT LOOP.

How to stack Claude agents so they run your content every single day.



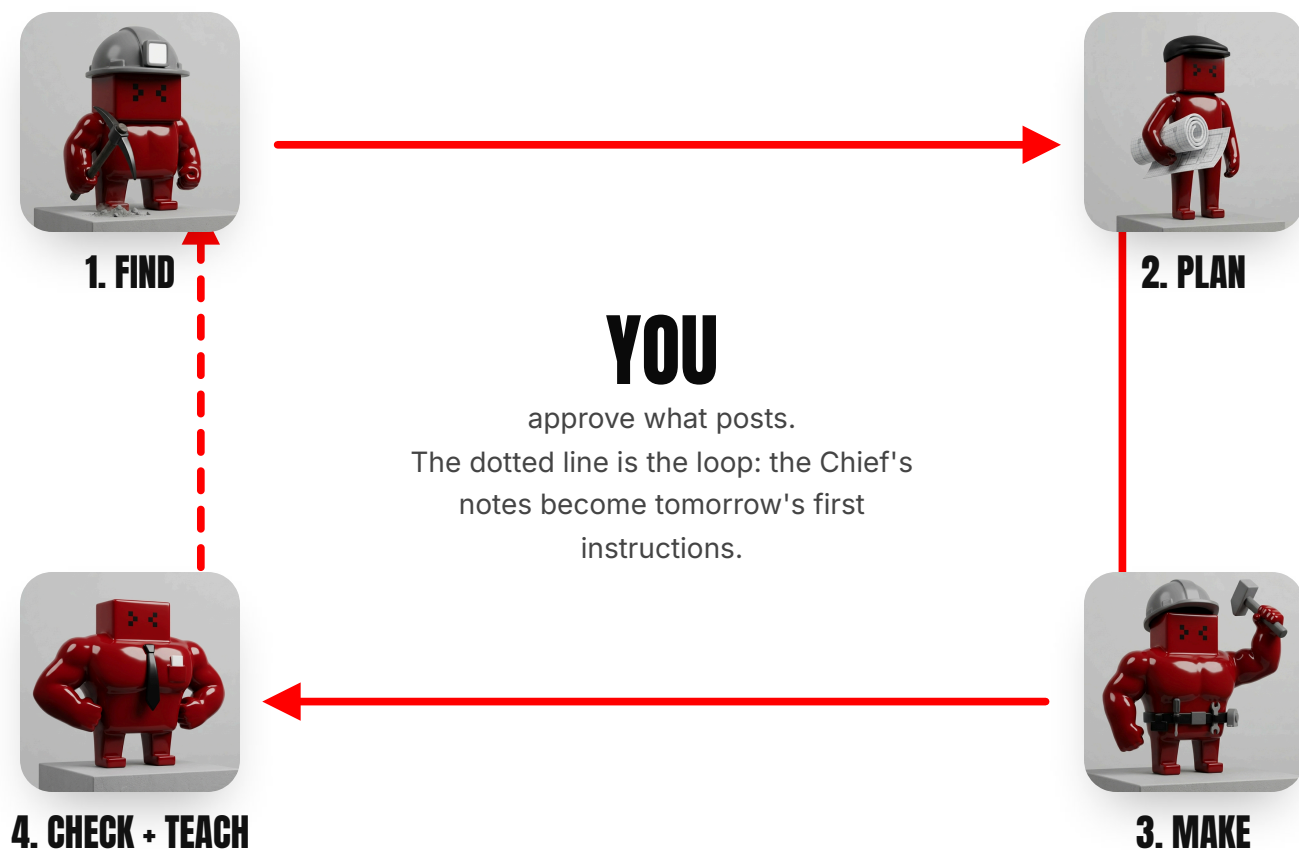
The exact setup behind my 10-agent team:
who does what, how they hand work to
each other, and how the loop feeds itself.



START HERE

MOST PEOPLE BUILD **ONE** AGENT. THE TRICK IS THE LOOP.

One agent that writes posts is a tool. A loop is a team: each agent does one job, hands its work to the next one, and the last one writes the notes the first one reads tomorrow. Every day the team starts smarter than the day before.



1 ONE AGENT, ONE JOB.

The fastest way to break a team is one agent that "does everything." Give each agent one job, one thing it reads, and one file it writes. Here is my team:

	AGENT	ITS ONE JOB	READS	WRITES
	MINER	Finds what is selling, not just getting views	The web + my notes	<code>miner/today.md</code>
	ARCHITECT	Turns findings into today's plan	<code>miner/today.md</code>	<code>architect/today.md</code>
	CREATOR	Writes hooks, slides and captions in my voice	<code>architect/today.md</code>	<code>creator/today.md</code>
	BUILDER	Designs the posts and saves them as drafts	<code>creator/today.md</code>	<code>builder/drafts/</code>
	CHIEF	Scores the work, picks the best, writes tomorrow's notes	Everything above	<code>directives.md</code>
	PACKAGER	Weekly: turns the best designs into templates	<code>builder/drafts/</code>	<code>kit/</code>

Start with the first 4 plus the Chief. Add the others once the loop runs clean for a week.

2 ONE SHARED FOLDER.

Agents can't talk to each other. They hand work over through files. Put everything in one folder (a GitHub repo works best for scheduled cloud agents, because each run can pull it, read and save back).

FOLDER LAYOUT

```
my-team/  
  CLAUDE.md      <- the rules every agent reads  
  directives.md   <- the Chief's notes for tomorrow  
  my-notes.md     <- YOUR overrides (always win)  
  .claude/skills/ <- the shared brain (rule 4)  
  agents/         <- one file per agent  
  outputs/miner/  outputs/architect/  outputs/creator/  
  outputs/builder/ outputs/chief/
```

3 STAGGER THE CLOCK.

Each agent needs the last one's file to exist before it starts. So schedule them an hour apart, in order. Mine run as scheduled Claude Code routines:

6:00 AM	MINER	9:00 AM	BUILDER
7:00 AM	ARCHITECT	10:00 AM	CHIEF OF STAFF
8:00 AM	CREATOR	Sundays	PACKAGER

If a file is missing, the agent should stop and write "waiting on Miner" instead of guessing. That one line saves you from garbage days.

4 SKILLS ARE THE SHARED BRAIN.

This is where most setups fall apart. If every agent has its own idea of your brand, your posts sound like five different people. Write your rules once, as skills, and make every agent load the same ones.

BRAND SKILL

Who you are, who you sell to, your voice, words you never use, your colors and fonts. Every agent loads this first.

HOOK SKILL

How you write openers, plus a grader that scores each hook before it ships. The Creator writes with it. The Chief checks with it.

FORMAT SKILL

The post layouts you've approved: slide count, sizes, where text goes. The Builder follows it so every post looks like you.

HONESTY SKILL

No invented numbers, no income claims, label every guess. This is the one that keeps your account safe.

A SKILL IS JUST A FILE

```
.claude/skills/hook-grader/SKILL.md
---
name: hook-grader
description: Score any hook before it posts. Use whenever
  you write or review a hook or first line.
---
Score each 0-2: stop power, about THEM, one emotion,
specific, open loop, right audience, honest.
8+ ships. 6-7 rewrite. Under 6: say why, then rewrite.
```

5 CLOSE THE LOOP.

Without this step you just have a line of agents doing the same thing every day. The loop is what makes them get better.

THE CHIEF WRITES TOMORROW

After scoring the day's work, the Chief writes `directives.md`: what worked, what flopped, and one direction for each agent. Example: "Miner, find 3 more products like the one that sold. Creator, shorter hooks."

EVERYONE READS IT FIRST

The first line of every agent file says: read `my-notes.md`, then `directives.md`, before doing anything. That's the dotted line on page 2.

YOUR NOTES ALWAYS WIN

Your file sits above the Chief's. If you say "no more sales posts this week," that beats anything the Chief wrote.

LOG EVERY CHANGE

When the Chief changes how an agent works, it adds a line to a change log with what it changed and why. If something breaks, you can roll it back.

6 GUARDRAILS FIRST.

Agents that run on their own need fences. These are mine, and I'd set them before you run anything:

DRAFTS ONLY

Agents save posts to a drafts folder. Nothing goes live until you move it. You stay the one who presses post.

NO SENDING, NO SPENDING

Only connect the tools an agent truly needs. A planner doesn't need email. A writer doesn't need your ad account.

NEVER INVENT PROOF

No made-up numbers, followers or results. "Not visible" beats a guess every time.

A REPORT YOU CAN READ IN 2 MINUTES

The Chief sends one short daily brief: what was made, the top pick, and one decision it needs from you.

COPY AND PASTE

BUILD YOUR FIRST TWO AGENTS TODAY.

Paste this into Claude Code inside an empty folder. It sets up the folder, your rules and the first two agents of the loop.

STARTER PROMPT

```
Set up an agent team in this folder for my content.
1. Create CLAUDE.md with my rules. Ask me 5 questions
   first: who I help, what I sell, my voice, words I
   never use, and where I post.
2. Create my-notes.md (my overrides, always read first)
   and directives.md (written by the Chief).
3. Create two agents in agents/:
   Miner: finds what is selling in my niche, writes
   outputs/miner/YYYY-MM-DD.md with a source for every line.
   Architect: reads the Miner's file and writes today's
   plan to outputs/architect/YYYY-MM-DD.md.
4. Every agent reads my-notes.md, then directives.md,
   before anything else. If its input file is missing,
   it stops and says what it is waiting on.
5. Drafts only. Never post, send or spend.
Then show me how to schedule them an hour apart.
```

AGENT FILE TEMPLATE

```
---
name: creator
description: Stage 3. Writes today's posts from the plan.
---
FIRST: read my-notes.md, then directives.md.
INPUT: outputs/architect/YYYY-MM-DD.md (stop if missing)
SKILLS: brand, hook-grader, formats
JOB: write the hooks, slides and captions for each post.
OUTPUT: outputs/creator/YYYY-MM-DD.md
NEVER: post, invent numbers, use words in my never-list.
```

THAT'S THE LOOP

START WITH TWO. CLOSE THE LOOP. THEN ADD ONE A WEEK.

My team didn't start with 10. It started with a Miner and an Architect, and grew one agent at a time as each one proved useful. Yours can too.



WANT MORE LIKE THIS?

Follow [@buildwith.justin](https://www.instagram.com/buildwithjustin) on Instagram. I share what this team builds, what breaks, and the prompts behind it. Free starter guide: justinleefleming.com/free

This is how I set up my own system. Results depend on your effort and consistency; nothing here promises income. The mascot-style characters are my own designs.