

WHAT THIS IS

Build Your Own AI Executive Assistant

Most people use AI like a chatbot they ask questions. This gets you an assistant that runs your day — one that reads your inbox, plans your calendar, preps every meeting, turns transcripts into filed notes, and chases your follow-ups. It drafts; you approve. Follow this top to bottom and you'll have a working daily assistant — plus a morning routine that runs on its own — by the end of one sitting.

BY CORINA • THE CV METHOD • @THE_CV_METHOD

WHO IT'S FOR

Anyone drowning in the admin around their actual work — inbox, calendar, meetings, follow-ups. Works in the Claude app / Cowork (no code) or in Claude Code. You bring the tools you already use; the assistant sits on top.

WHAT YOU'LL HAVE BUILT

Five commands your assistant runs on request, plus one morning sweep that fires on its own — surfacing your day, drafting your prep, and creating your tasks before you open your laptop.

TIME

About 45 minutes to wire up the five commands and the context doc. The morning sweep is a second, optional sitting.

WHAT'S INSIDE

- The one rule that keeps an assistant safe to trust
- The context doc that makes it actually useful
- Putting it on a schedule so it runs itself
- The five daily commands, with the exact prompts
- The proactive morning sweep, assembled

The one rule

An assistant that sends emails, books meetings, or messages people on its own is a liability. An assistant that does all the work and then hands it to you to approve is leverage. That line is the whole thing.

RULE
ZERO

It drafts. You approve.

Every command below ends in a draft, a proposal, or a list — never a sent email or a booked invite. You stay the one who ships. That's what lets you hand off the work without handing off the judgment.

BEFORE YOU START

Two things to decide first

01

Which tools it plugs into.

Your assistant is only as good as what it can see. Decide which of your real tools to connect — your calendar, your email, and your notes or knowledge base (where your project context lives). Connect them through your AI's official connectors before you start; the prompts below assume it can read them.

02

Where it runs.

In the Claude app / Cowork (tied to your account, no code, and the only path that can run on a schedule while you're away), or in Claude Code on your machine. For the morning sweep in Part 4, you want the app/Cowork path — a scheduled task needs to run without you sitting there.

HONEST NOTE

AI tools and their connectors move fast. Everything here was written against how these tools worked in August 2026. If a connector or a menu has moved by the time you read this, trust the official docs (linked at the end) over this page.

THE STEP EVERYONE SKIPS

Write the assistant a context doc

The difference between a generic assistant and one that feels like yours is a single document it reads first — who you are, what you're working on, who's who, and how you like things done. Write it once; every command gets sharper for it.

Keep it in whatever your assistant reads first (a pinned note, a project file, or a context doc). Start small — you'll add to it as you notice it guessing.

A STARTER CONTEXT DOC — FILL IN THE BLANKS

```
# Who I am
Name, role, what I actually spend my week doing.

# What I'm working on
- Project A — one line: what it is, who's involved, where it stands.
- Project B — same.

# The people
- Name — how I know them, which project, how formal to be.

# How I like things done
- Voice: warm, direct, plain English. No hype.
- Never invent a number, date, or result. If it isn't in my notes, leave it out.
- Draft only. I send.
```

WHY THIS EARNS ITS KEEP

Every command below is really "do this task using what you know about me." The context doc is that knowledge. Skip it and you'll re-explain yourself every time — the exact thing you're trying to stop doing.

These are the five your assistant runs on request. Each is a prompt you can paste as-is — the [brackets] are where your real context goes. In your own setup, save each as a shortcut so a few words trigger the whole thing.

COMMAND	WHEN YOU REACH FOR IT	WHAT COMES BACK
Morning inbox brief	before you open your inbox	inbox sorted into reply / wait / noise, plus today's meetings
Plan my day	staring at a blank calendar	a time-blocked day around your real meetings
Meeting prep	right before a meeting	a one-page brief on the room, the history, your points
Meeting notes	the second a meeting ends	summary, decisions, tasks, drafted follow-ups
Who owes me a reply?	chasing loose ends	a ranked list with the nudges pre-drafted

01 • MORNING INBOX BRIEF

```
# say: "what came in overnight?"
Scan my inbox from the last 24 hrs.
Sort into: (1) needs my reply, (2) can wait, (3) noise.
For each reply-needed: sender, the ask in one line, a suggested next step.
Then list today's meetings with times.
Don't send anything — just brief me.
```

02 • PLAN MY DAY

```
# say: "plan my day"
Here's today's calendar: [paste your events].
Build a time-blocked day around these fixed meetings:
15-min buffers, one protected 90-min focus block, a slack block for overflow.
Return a clean timeline for me to approve before booking.
```

03 • MEETING PREP

```
# say: "prep me for the 2pm"
Meeting: [name, time]. Attendees: [names].
Context: [paste the last thread or notes].
Make a 1-page prep: who each person is, where we left off,
project status, 3 talking points, and what to bring. Keep it skimmable.
```

04 • MEETING NOTES

```
# say: "here's the transcript – notes please"  
Here's the transcript: [paste it].  
Give me: a 3-sentence summary, the decisions,  
and action items as "owner → task → due date."  
List my action items separately.  
Draft any follow-up emails I owe – don't send them.
```

05 • WHO OWES ME A REPLY?

```
# say: "who owes me a reply?"  
Review my sent emails and open threads.  
Find every one where I'm waiting on a reply that hasn't come back yet.  
Rank them by how overdue + how important.  
Draft a short, friendly nudge for each – for me to review and send.
```

NOTICE THE LAST LINE

Every prompt ends by telling the assistant to stop at a draft. That single line is Rule Zero, enforced. Keep it on every command you add later.

WHERE IT STOPS BEING A TOOL AND STARTS BEING AN ASSISTANT

Assemble the morning sweep

The five commands are reactive — you ask, it answers. The real unlock is chaining a few of them into one routine that runs before you're even up: read the day, pre-build the prep, and create the tasks — so the work is waiting for you, not waiting on you.

It's one instruction that ties Parts 1 and 2 together. It still ends where everything ends: a brief for you to approve.

THE MORNING SWEEP — ONE INSTRUCTION

```
# the routine, run once each morning
Using my context doc:
1. Read today's calendar.
2. For each real meeting, find the matching project note and pull the context.
3. Draft a prep one-pager per meeting (attendees, last thread, status, points).
4. Create a task for anything that needs action today.
5. Send me ONE morning brief with it all.
Draft and create in my own tools only — book nothing, send nothing.
```

THE ONE TO GET RIGHT

How the assistant maps a calendar meeting to the right note is the whole game. Give it a rule it can follow — match on the project name in the event title, on an attendee, or on a link you drop in the invite. When it can't find context, it should say "no prior context found" and flag it — never invent history.

SO IT RUNS WITHOUT YOU

Wire the sweep to a morning trigger

A routine you have to remember to run isn't automation. Set the morning sweep as a scheduled task in the Claude app / Cowork so it fires every weekday morning on its own and leaves the brief waiting for you.

Start it on weekdays a little before your day begins. The output is a draft you open with coffee — nothing has left your system without you.

Run it read-only for the first week: let it produce the brief and show you which note it would use for each meeting, before you let it create a single task. Trust is earned by watching it behave.

[SCREENSHOT PLACEHOLDER — CORINA TO ADD]

Your scheduled task set to fire on weekday mornings, and the morning brief it produces — a real shot of it running makes this land.

DURABILITY

Use your AI's real scheduled-task feature (the one that survives you closing the app), not a one-off timer inside a single chat. A scheduled task starts a fresh run each morning — which is exactly why the context doc in Part 1 matters: it's how each fresh run knows your world.

KEEP IT HONEST & SAFE

It drafts — you approve

Books nothing, sends nothing on its own

Your data stays in your own tools

No invented meetings, numbers, or history

Anything ambiguous is flagged, not guessed

THE ONE-SITTING CHECKLIST

- 01 Calendar, email, and notes connected to your AI.
- 02 Context doc written: who you are, your projects, the people, how you like things done.
- 03 The five commands saved as shortcuts and tested on a real day.
- 04 Morning sweep assembled — with a clear meeting-to-note matching rule.
- 05 Ran read-only for a week; every drift patched back into the context doc.
- 06 Sweep on a weekday morning schedule; the brief lands before you do.

Start from the source — connectors, scheduling, and skills all move fast, so verify here if anything above has changed:

Connectors & the Claude apps

support.claude.com

Build-with-Claude docs

docs.claude.com

Skills + skill-creator (to save each command)

github.com/anthropics/skills

RATHER HAVE THE WHOLE THING WIRED UP FOR YOU?

I build these assistants end to end.

Connected to your tools, tuned to your voice, gated so it drafts and you approve, and scheduled so it runs on its own. That's the CV Method. Comment **VEE** or reach out.

hello.thecvmethod.com →