20

AI AUTOMATIONS I RUN AS A FOUNDER

What each one does, what it replaced, a starter prompt to try it today, and an Autopilot prompt that sets it up to run hands-free.

ERIC KURBANOV

Founder & CEO, RKIN · erickurbanov.com

I run RKIN, a water filtration company. These are the 20 AI automations I run for myself right now. My team and the company have their own; this list is mine. Some took an afternoon to build. Some took weeks. Under each one is a starter prompt you can paste into Claude, ChatGPT, or whatever AI assistant you use. The two prompts work as a pair. Step 1, the starter prompt, is for today: paste it in, fill in the brackets, feed it your data by hand, and see if the result is useful. You can stop there. Step 2, the Autopilot prompt, is for when you want it to run without you: paste it into the same conversation, right after the starter prompt has worked. It turns your AI into a setup engineer that walks you through connecting every tool the automation needs, schedules it, tests it on real data, and hands you a runbook. Start with a small, redacted sample you have permission to share. Step 1 produces a result in the chat; it does not connect accounts or schedule work. Step 2 needs supported integrations and a persistent scheduler. You may need to approve access or paid features; a prompt alone cannot create those capabilities. Keep the human approvals on the final pages in place.



FIRST, READ THE IDEA

Each page is one automation: what it does for me and what it replaced.

1

STARTER PROMPT

Use it on its own. Paste it into your AI assistant, fill in the brackets, give it your data by hand, and judge the result. Stop here if manual is enough.

2

AUTOPILOT PROMPT

Use after Step 1, in the same chat. Connect the tools together, confirm a persistent schedule, test the output, and keep the agreed human approvals.

RUNNING MY DAY

01 The morning brief.

Before I'm up, an AI assistant reads my calendar, inbox, sales numbers, and open projects, then writes a one-page brief. Anything that needs action becomes a task automatically, and it checks the last week so I don't get duplicates.

BEFORE Tab-hopping every morning to figure out what today is.

STEP 1 · STARTER PROMPT use it alone to try this by hand today

```
You are my chief of staff. For this one-time trial, use the calendar, email excerpts, sales numbers, and project list I provide below. Write a one-page brief: today's schedule, the 3 things that need a decision from me, anything unusual in the numbers, and a short task list. Never invent data. If a source is unavailable, say so at the top.
```

STEP 2 · AUTOPILOT PROMPT paste into the same chat once step 1 works, to make it hands-free

```
You are my automation engineer. Goal: make the morning brief run automatically within the approval limits below. The job is the one I gave you earlier in this conversation. If you don't see it above, stop and ask me to paste the starter prompt first. Work with me one step at a time and wait for my answer before moving on.
1) Ask which tools I use for: calendar, email, sales or order system, and task or project list.
2) For each one, check whether you already have a connector or integration for it. If you don't, walk me through connecting it: where to click, which permissions to grant (the minimum needed), and then prove it works by pulling one real record. Never ask me to paste a password or API key into this chat; use the tool's official connect or sign-in flow.
3) Set the trigger: every weekday at 6 AM in my time zone.
4) Set the delivery: send the brief to me by email or chat, and create the tasks in my task tool.
5) Lock in the guardrail: read-only source access; only deliver the brief to me and create the agreed tasks. Never contact anyone else.
6) Confirm the schedule and time zone in a persistent scheduler, not just this chat. Run one test, verify its actual output, and prevent duplicate actions on retries.
7) Give me a one-page runbook: connected tools, trigger, delivery, failure alerts, and how to pause it. If a required capability is missing, say what is blocked; do not claim it is running.
```

02 Meeting prep that knows the customer.

At 5:30 AM, every meeting on my calendar gets a prep sheet. If the attendee is a customer, it pulls their order history and support conversations so I walk in knowing the full story.

BEFORE Looking people up five minutes before the call, or not at all.

STEP 1 · STARTER PROMPT use it alone to try this by hand today

```
Using the meeting list, order history, and support excerpts I provide, build
a one-page prep sheet for each meeting today. Summarize who they are, what
they bought and when, any open or recent support issues, and the 3 questions
I should ask. If an attendee has no record, say 'no history found' for that
person and continue with the others. Do not guess.
```

STEP 2 · AUTOPILOT PROMPT paste into the same chat once step 1 works, to make it hands-free

```
You are my automation engineer. Goal: make meeting prep run automatically
within the approval limits below. The job is the one I gave you earlier in
this conversation. If you don't see it above, stop and ask me to paste the
starter prompt first. Work with me one step at a time and wait for my answer
before moving on.
1) Ask which tools I use for: calendar, order or CRM system, and support
inbox.
2) For each one, check whether you already have a connector or integration
for it. If you don't, walk me through connecting it: where to click, which
permissions to grant (the minimum needed), and then prove it works by pulling
one real record. Never ask me to paste a password or API key into this chat;
use the tool's official connect or sign-in flow.
3) Set the trigger: every day at 5:30 AM, for that day's meetings.
4) Set the delivery: one prep sheet per meeting, delivered to me before my
first meeting.
5) Lock in the guardrail: read-only access; never contact an attendee.
6) Confirm the schedule and time zone in a persistent scheduler, not just
this chat. Run one test, verify its actual output, and prevent duplicate
actions on retries.
7) Give me a one-page runbook: connected tools, trigger, delivery, failure
alerts, and how to pause it. If a required capability is missing, say what is
blocked; do not claim it is running.
```

03 Podcast guest research.

When a podcast booking lands on my calendar, the assistant researches the show, the host, the audience, and the questions that host tends to ask. I get talking points and a follow-up checklist.

BEFORE Winging it, or listening to old episodes the night before.

STEP 1 · STARTER PROMPT use it alone to try this by hand today

```
I'm booked as a guest on [PODCAST NAME] hosted by [HOST]. Research the show: audience, format, episode length, and the last 10 episode topics. Research the host's background. List the 10 questions this host is most likely to ask me, with a confidence level for each, based on past episodes. Draft talking points from my bio below, and end with a post-recording checklist. [PASTE BIO]
```

STEP 2 · AUTOPILOT PROMPT paste into the same chat once step 1 works, to make it hands-free

```
You are my automation engineer. Goal: make podcast guest research run automatically within the approval limits below. The job is the one I gave you earlier in this conversation. If you don't see it above, stop and ask me to paste the starter prompt first. Work with me one step at a time and wait for my answer before moving on.
1) Ask which tools I use for: calendar, email, and web search.
2) For each one, check whether you already have a connector or integration for it. If you don't, walk me through connecting it: where to click, which permissions to grant (the minimum needed), and then prove it works by pulling one real record. Never ask me to paste a password or API key into this chat; use the tool's official connect or sign-in flow.
3) Set the trigger: whenever a calendar event containing 'podcast' or 'interview' is created or changed.
4) Set the delivery: a briefing document saved to my notes or docs tool, with a reminder 24 hours before recording.
5) Lock in the guardrail: research only; never email the host.
6) Confirm the schedule and time zone in a persistent scheduler, not just this chat. Run one test, verify its actual output, and prevent duplicate actions on retries.
7) Give me a one-page runbook: connected tools, trigger, delivery, failure alerts, and how to pause it. If a required capability is missing, say what is blocked; do not claim it is running.
```

04 Inbox triage with drafts, not sends.

An executive-assistant agent sorts my inbox, flags what matters, and drafts replies. It never sends anything. I approve or edit.

BEFORE Email ran my mornings.

STEP 1 · STARTER PROMPT use it alone to try this by hand today

```
Act as my executive assistant for email. Sort the email excerpts I provide into: needs my reply today, needs my reply this week, FYI, and ignore. For every message that needs a reply, write a draft in my voice (short, direct, friendly) and show it here for review. Never send anything or modify my inbox. Flag anything involving money, legal, or an upset customer at the very top.
```

STEP 2 · AUTOPILOT PROMPT paste into the same chat once step 1 works, to make it hands-free

```
You are my automation engineer. Goal: make inbox triage run automatically within the approval limits below. The job is the one I gave you earlier in this conversation. If you don't see it above, stop and ask me to paste the starter prompt first. Work with me one step at a time and wait for my answer before moving on.
1) Ask which tools I use for: email account, with permission to label and create drafts.
2) For each one, check whether you already have a connector or integration for it. If you don't, walk me through connecting it: where to click, which permissions to grant (the minimum needed), and then prove it works by pulling one real record. Never ask me to paste a password or API key into this chat; use the tool's official connect or sign-in flow.
3) Set the trigger: every 2 hours during my work day.
4) Set the delivery: labels applied in my inbox, drafts saved in my drafts folder, and one daily summary to me.
5) Lock in the guardrail: permission to draft only; sending must stay off.
6) Confirm the schedule and time zone in a persistent scheduler, not just this chat. Run one test, verify its actual output, and prevent duplicate actions on retries.
7) Give me a one-page runbook: connected tools, trigger, delivery, failure alerts, and how to pause it. If a required capability is missing, say what is blocked; do not claim it is running.
```

05 An accountability coach that checks in.

Five times a day a coaching agent checks me against my daily commitments: goals review, exercise, journaling, a social post, outreach. It keeps score. It's blunt on purpose.

BEFORE Good intentions.

STEP 1 · STARTER PROMPT use it alone to try this by hand today

```
You are my accountability coach. My five daily commitments are: [LIST THEM].
Run one check-in now: ask what I've finished, record my answer in this chat,
and score only the commitments I report. Be blunt and brief. Do not schedule
reminders yet.
```

STEP 2 · AUTOPILOT PROMPT paste into the same chat once step 1 works, to make it hands-free

```
You are my automation engineer. Goal: make the accountability coach run
automatically within the approval limits below. The job is the one I gave you
earlier in this conversation. If you don't see it above, stop and ask me to
paste the starter prompt first. Work with me one step at a time and wait for
my answer before moving on.
1) Ask which tools I use for: the chat or text app I actually check, and a
simple sheet or notes file for the score.
2) For each one, check whether you already have a connector or integration
for it. If you don't, walk me through connecting it: where to click, which
permissions to grant (the minimum needed), and then prove it works by pulling
one real record. Never ask me to paste a password or API key into this chat;
use the tool's official connect or sign-in flow.
3) Set the trigger: five check-ins a day at times I choose, plus a Sunday
scorecard.
4) Set the delivery: check-ins in chat, the log kept in the sheet.
5) Lock in the guardrail: it messages only me, never anyone else.
6) Confirm the schedule and time zone in a persistent scheduler, not just
this chat. Run one test, verify its actual output, and prevent duplicate
actions on retries.
7) Give me a one-page runbook: connected tools, trigger, delivery, failure
alerts, and how to pause it. If a required capability is missing, say what is
blocked; do not claim it is running.
```

SALES

06 Every sales call, analyzed overnight.

Each night a nine-step pipeline takes the day's recorded consultation calls, transcribes them, works out who said what, and updates the deal: stage, next step, follow-up date.

BEFORE Notes that got written if someone remembered.

STEP 1 · STARTER PROMPT use it alone to try this by hand today

```
Here is a transcript of a sales call. Identify who is the salesperson and who is the customer. Then extract: the customer's situation and needs, objections raised, what was promised, the deal stage (new, qualified, quoted, won, lost), the single next step, and a follow-up date. Output as structured fields I can paste into my CRM. Quote the customer's exact words for every objection. [PASTE TRANSCRIPT]
```

STEP 2 · AUTOPILOT PROMPT paste into the same chat once step 1 works, to make it hands-free

```
You are my automation engineer. Goal: make sales call analysis run automatically within the approval limits below. The job is the one I gave you earlier in this conversation. If you don't see it above, stop and ask me to paste the starter prompt first. Work with me one step at a time and wait for my answer before moving on.
1) Ask which tools I use for: call recording or meeting tool, transcription, and CRM.
2) For each one, check whether you already have a connector or integration for it. If you don't, walk me through connecting it: where to click, which permissions to grant (the minimum needed), and then prove it works by pulling one real record. Never ask me to paste a password or API key into this chat; use the tool's official connect or sign-in flow.
3) Set the trigger: every night, for that day's recorded calls.
4) Set the delivery: CRM fields updated and a one-line summary per call sent to me.
5) Lock in the guardrail: for the first two weeks, show me every CRM change for approval before writing it.
6) Confirm the schedule and time zone in a persistent scheduler, not just this chat. Run one test, verify its actual output, and prevent duplicate actions on retries.
7) Give me a one-page runbook: connected tools, trigger, delivery, failure alerts, and how to pause it. If a required capability is missing, say what is blocked; do not claim it is running.
```

07 Weekly sales coaching from real calls.

Once a week the same data becomes a coaching report for the team: what worked, what stalled, which questions we skipped.

BEFORE Coaching by gut feel.

STEP 1 · STARTER PROMPT use it alone to try this by hand today

```
Here are this week's sales call summaries. Write a one-page coaching report
for the team: what our best calls had in common, where deals stalled,
discovery questions we skipped, and one specific thing each rep should
practice next week. Use real examples from the calls. Be specific and kind.
No generic sales advice. [PASTE SUMMARIES]
```

STEP 2 · AUTOPILOT PROMPT paste into the same chat once step 1 works, to make it hands-free

```
You are my automation engineer. Goal: make weekly sales coaching run
automatically within the approval limits below. The job is the one I gave you
earlier in this conversation. If you don't see it above, stop and ask me to
paste the starter prompt first. Work with me one step at a time and wait for
my answer before moving on.
1) Ask which tools I use for: the call summaries from automation 6 and the
place my team reads updates.
2) For each one, check whether you already have a connector or integration
for it. If you don't, walk me through connecting it: where to click, which
permissions to grant (the minimum needed), and then prove it works by pulling
one real record. Never ask me to paste a password or API key into this chat;
use the tool's official connect or sign-in flow.
3) Set the trigger: every Friday afternoon.
4) Set the delivery: one coaching report posted to the team channel after I
approve it.
5) Lock in the guardrail: nothing goes to the team without my OK.
6) Confirm the schedule and time zone in a persistent scheduler, not just
this chat. Run one test, verify its actual output, and prevent duplicate
actions on retries.
7) Give me a one-page runbook: connected tools, trigger, delivery, failure
alerts, and how to pause it. If a required capability is missing, say what is
blocked; do not claim it is running.
```

08 Win/loss review.

A nightly job looks at deals that closed or died and records why, so patterns show up in weeks instead of years.

BEFORE We didn't do it.

STEP 1 · STARTER PROMPT use it alone to try this by hand today

```
Here is a list of deals that closed or were lost this month, with call notes. For each one, state the main reason it was won or lost in one sentence, using evidence from the notes. Then group the reasons and tell me the top 3 patterns and one change we should test next month. [PASTE DEALS]
```

STEP 2 · AUTOPILOT PROMPT paste into the same chat once step 1 works, to make it hands-free

```
You are my automation engineer. Goal: make win/loss review run automatically within the approval limits below. The job is the one I gave you earlier in this conversation. If you don't see it above, stop and ask me to paste the starter prompt first. Work with me one step at a time and wait for my answer before moving on.
1) Ask which tools I use for: CRM and call notes.
2) For each one, check whether you already have a connector or integration for it. If you don't, walk me through connecting it: where to click, which permissions to grant (the minimum needed), and then prove it works by pulling one real record. Never ask me to paste a password or API key into this chat; use the tool's official connect or sign-in flow.
3) Set the trigger: on the first of each month, plus any time a deal is marked won or lost.
4) Set the delivery: a monthly patterns report to me.
5) Lock in the guardrail: read-only access.
6) Confirm the schedule and time zone in a persistent scheduler, not just this chat. Run one test, verify its actual output, and prevent duplicate actions on retries.
7) Give me a one-page runbook: connected tools, trigger, delivery, failure alerts, and how to pause it. If a required capability is missing, say what is blocked; do not claim it is running.
```

09 Abandoned carts go to a person.

When a high-intent cart is abandoned, the system creates a task for our customer service team with the context attached. A human reaches out. No robot emails pretending to be me.

BEFORE A generic email sequence and nothing else.

STEP 1 · STARTER PROMPT use it alone to try this by hand today

```
Using the abandoned-cart and customer-history data I provide, draft a task
for our customer service team for each cart worth more than $[AMOUNT]. Show
the task here; do not create it in another tool yet. Include: customer name,
what was in the cart, whether they are a past customer, any prior support
conversations, and a suggested opening line for a personal, helpful outreach.
Do not email the customer yourself. A person makes the contact.
```

STEP 2 · AUTOPILOT PROMPT paste into the same chat once step 1 works, to make it hands-free

```
You are my automation engineer. Goal: make abandoned-cart handoff run
automatically within the approval limits below. The job is the one I gave you
earlier in this conversation. If you don't see it above, stop and ask me to
paste the starter prompt first. Work with me one step at a time and wait for
my answer before moving on.
1) Ask which tools I use for: store platform, customer and order history,
support inbox, and the task tool my service team uses.
2) For each one, check whether you already have a connector or integration
for it. If you don't, walk me through connecting it: where to click, which
permissions to grant (the minimum needed), and then prove it works by pulling
one real record. Never ask me to paste a password or API key into this chat;
use the tool's official connect or sign-in flow.
3) Set the trigger: whenever a cart above my threshold has been abandoned for
60 minutes.
4) Set the delivery: a task assigned to the service team with full context.
5) Lock in the guardrail: never email or text the customer; a person makes
contact.
6) Confirm the schedule and time zone in a persistent scheduler, not just
this chat. Run one test, verify its actual output, and prevent duplicate
actions on retries.
7) Give me a one-page runbook: connected tools, trigger, delivery, failure
alerts, and how to pause it. If a required capability is missing, say what is
blocked; do not claim it is running.
```

10 Local partner outreach, researched and drafted.

Three jobs run overnight: find local trade businesses that fit, research each one, draft a personal note for me to approve.

BEFORE A spreadsheet I never got to.

STEP 1 · STARTER PROMPT use it alone to try this by hand today

```
Find 10 [TYPE OF BUSINESS] within [CITY/REGION] that would be a good referral partner for a company that sells [WHAT YOU SELL]. For each: business name, owner's name if public, website, and why they fit. Then draft a short, personal outreach note for each one that mentions something specific about their business. Show the notes here as drafts for my approval. Do not save to another tool or send.
```

STEP 2 · AUTOPILOT PROMPT paste into the same chat once step 1 works, to make it hands-free

```
You are my automation engineer. Goal: make partner outreach run automatically within the approval limits below. The job is the one I gave you earlier in this conversation. If you don't see it above, stop and ask me to paste the starter prompt first. Work with me one step at a time and wait for my answer before moving on.
1) Ask which tools I use for: web search, a spreadsheet or CRM to hold prospects, and email drafts.
2) For each one, check whether you already have a connector or integration for it. If you don't, walk me through connecting it: where to click, which permissions to grant (the minimum needed), and then prove it works by pulling one real record. Never ask me to paste a password or API key into this chat; use the tool's official connect or sign-in flow.
3) Set the trigger: every night, 10 new prospects.
4) Set the delivery: prospects added to the sheet, and outreach notes saved as drafts.
5) Lock in the guardrail: drafts only; I review and send.
6) Confirm the schedule and time zone in a persistent scheduler, not just this chat. Run one test, verify its actual output, and prevent duplicate actions on retries.
7) Give me a one-page runbook: connected tools, trigger, delivery, failure alerts, and how to pause it. If a required capability is missing, say what is blocked; do not claim it is running.
```

CUSTOMERS AND SUPPORT

11 A support brain built only from verified facts.

Our support team can ask an AI a product question and get an answer drawn only from our verified product documents. It refreshes daily, and when it doesn't know, it says so.

BEFORE "let me check and get back to you."

STEP 1 · STARTER PROMPT use it alone to try this by hand today

```
You are a support assistant for [COMPANY]. Answer questions using ONLY the attached product documents. Quote the document and section for every answer. If the documents don't contain the answer, reply exactly: 'I don't have a verified answer for that. Please check with a team lead.' Never guess specs, warranty terms, or compatibility. [ATTACH PRODUCT DOCS]
```

STEP 2 · AUTOPILOT PROMPT paste into the same chat once step 1 works, to make it hands-free

```
You are my automation engineer. Goal: make the support answer assistant run automatically within the approval limits below. The job is the one I gave you earlier in this conversation. If you don't see it above, stop and ask me to paste the starter prompt first. Work with me one step at a time and wait for my answer before moving on.
1) Ask which tools I use for: our product documents folder and the chat tool my support team uses.
2) For each one, check whether you already have a connector or integration for it. If you don't, walk me through connecting it: where to click, which permissions to grant (the minimum needed), and then prove it works by pulling one real record. Never ask me to paste a password or API key into this chat; use the tool's official connect or sign-in flow.
3) Set the trigger: on demand, whenever a team member asks a question; refresh the documents daily.
4) Set the delivery: answers in the team's chat with the source document cited.
5) Lock in the guardrail: answer only from the documents; say 'no verified answer' otherwise; never talk to customers directly.
6) Confirm the schedule and time zone in a persistent scheduler, not just this chat. Run one test, verify its actual output, and prevent duplicate actions on retries.
7) Give me a one-page runbook: connected tools, trigger, delivery, failure alerts, and how to pause it. If a required capability is missing, say what is blocked; do not claim it is running.
```

12 Filter reminders timed to each customer.

A daily sync works out who is due for replacement filters based on what they bought and when, and queues the reminder.

BEFORE One blast to everyone on the same schedule.

STEP 1 · STARTER PROMPT use it alone to try this by hand today

```
Here is an export of customer orders with product and purchase date. Using the replacement schedule below, list every customer who is due for a replacement in the next 30 days, which item they need, and the date it's due. Output a list I can upload to my email tool. Skip anyone who already reordered. [PASTE SCHEDULE AND ORDER EXPORT]
```

STEP 2 · AUTOPILOT PROMPT paste into the same chat once step 1 works, to make it hands-free

```
You are my automation engineer. Goal: make timed reorder reminders run automatically within the approval limits below. The job is the one I gave you earlier in this conversation. If you don't see it above, stop and ask me to paste the starter prompt first. Work with me one step at a time and wait for my answer before moving on.
1) Ask which tools I use for: store order history and email marketing tool.
2) For each one, check whether you already have a connector or integration for it. If you don't, walk me through connecting it: where to click, which permissions to grant (the minimum needed), and then prove it works by pulling one real record. Never ask me to paste a password or API key into this chat; use the tool's official connect or sign-in flow.
3) Set the trigger: every morning.
4) Set the delivery: an updated 'due for replacement' segment in my email tool.
5) Lock in the guardrail: it builds the segment only; I control the campaign that sends.
6) Confirm the schedule and time zone in a persistent scheduler, not just this chat. Run one test, verify its actual output, and prevent duplicate actions on retries.
7) Give me a one-page runbook: connected tools, trigger, delivery, failure alerts, and how to pause it. If a required capability is missing, say what is blocked; do not claim it is running.
```

13 Review moderation.

New product reviews are read and sorted overnight. Anything that needs a human reply is flagged to the team.

BEFORE Checked whenever someone thought of it.

STEP 1 · STARTER PROMPT use it alone to try this by hand today

```
Here are this week's new product reviews. Sort them into: publish as is,
publish and reply, and needs a human (complaint, safety issue, or order
problem). For the ones that need a reply, draft a short, genuine response in
our brand voice. For the ones that need a human, summarize the issue in one
line. Do not publish or reply yourself. [PASTE REVIEWS]
```

STEP 2 · AUTOPILOT PROMPT paste into the same chat once step 1 works, to make it hands-free

```
You are my automation engineer. Goal: make review moderation run
automatically within the approval limits below. The job is the one I gave you
earlier in this conversation. If you don't see it above, stop and ask me to
paste the starter prompt first. Work with me one step at a time and wait for
my answer before moving on.
1) Ask which tools I use for: product review platform and the team's task or
chat tool.
2) For each one, check whether you already have a connector or integration
for it. If you don't, walk me through connecting it: where to click, which
permissions to grant (the minimum needed), and then prove it works by pulling
one real record. Never ask me to paste a password or API key into this chat;
use the tool's official connect or sign-in flow.
3) Set the trigger: every night.
4) Set the delivery: a sorted list with draft replies, and a task for
anything that needs a person.
5) Lock in the guardrail: never publish, reply, or hide a review on its own.
6) Confirm the schedule and time zone in a persistent scheduler, not just
this chat. Run one test, verify its actual output, and prevent duplicate
actions on retries.
7) Give me a one-page runbook: connected tools, trigger, delivery, failure
alerts, and how to pause it. If a required capability is missing, say what is
blocked; do not claim it is running.
```

MARKETING AND MONEY

14 Ad spend guardrails.

A daily review marks each campaign keep or kill, based on rules I set. A separate watcher checks spend every minute on our most sensitive campaign and pauses it if it breaks the limit.

BEFORE Finding out on Monday what went wrong on Saturday.

STEP 1 · STARTER PROMPT use it alone to try this by hand today

```
Here is yesterday's performance for each ad campaign: spend, clicks,
purchases, revenue. My rules: [YOUR RULES, e.g. pause if spend exceeds $X
with zero purchases; keep if return is above Y]. Apply the rules and give me
a keep / watch / kill list with a one-line reason for each. Recommend only. I
make the changes.
```

STEP 2 · AUTOPILOT PROMPT paste into the same chat once step 1 works, to make it hands-free

```
You are my automation engineer. Goal: make ad spend guardrails run
automatically within the approval limits below. The job is the one I gave you
earlier in this conversation. If you don't see it above, stop and ask me to
paste the starter prompt first. Work with me one step at a time and wait for
my answer before moving on.
1) Ask which tools I use for: each ad platform I use and my sales data.
2) For each one, check whether you already have a connector or integration
for it. If you don't, walk me through connecting it: where to click, which
permissions to grant (the minimum needed), and then prove it works by pulling
one real record. Never ask me to paste a password or API key into this chat;
use the tool's official connect or sign-in flow.
3) Set the trigger: a daily keep/watch/kill report, plus a spend check every
15 minutes on campaigns I mark as sensitive.
4) Set the delivery: the daily report to me, and an immediate alert when a
limit is broken.
5) Lock in the guardrail: start in recommend-only mode; turn on auto-pause
only for the specific limits I approve in writing.
6) Confirm the schedule and time zone in a persistent scheduler, not just
this chat. Run one test, verify its actual output, and prevent duplicate
actions on retries.
7) Give me a one-page runbook: connected tools, trigger, delivery, failure
alerts, and how to pause it. If a required capability is missing, say what is
blocked; do not claim it is running.
```

15 True margin, every night.

Overnight jobs pull sales, fees, and ad costs from every channel and rebuild contribution margin by product and channel.

BEFORE A month-end surprise.

STEP 1 · STARTER PROMPT use it alone to try this by hand today

```
Here are yesterday's sales by product and channel, plus my costs: product cost, shipping, marketplace fees, payment fees, and ad spend by channel. Calculate contribution margin per product and per channel. Show the 5 most and 5 least profitable, and flag anything where margin dropped more than 5 points versus last week. Show your math. [PASTE DATA]
```

STEP 2 · AUTOPILOT PROMPT paste into the same chat once step 1 works, to make it hands-free

```
You are my automation engineer. Goal: make nightly true margin run automatically within the approval limits below. The job is the one I gave you earlier in this conversation. If you don't see it above, stop and ask me to paste the starter prompt first. Work with me one step at a time and wait for my answer before moving on.
1) Ask which tools I use for: store and marketplace sales, payment and marketplace fees, shipping costs, ad spend, and product costs.
2) For each one, check whether you already have a connector or integration for it. If you don't, walk me through connecting it: where to click, which permissions to grant (the minimum needed), and then prove it works by pulling one real record. Never ask me to paste a password or API key into this chat; use the tool's official connect or sign-in flow.
3) Set the trigger: every night after midnight.
4) Set the delivery: a margin dashboard or sheet, plus an alert when any product's margin drops more than 5 points.
5) Lock in the guardrail: read-only access to every financial system.
6) Confirm the schedule and time zone in a persistent scheduler, not just this chat. Run one test, verify its actual output, and prevent duplicate actions on retries.
7) Give me a one-page runbook: connected tools, trigger, delivery, failure alerts, and how to pause it. If a required capability is missing, say what is blocked; do not claim it is running.
```

16 A revenue leak detector.

Every four hours the system tests the steps customers use to buy and compares sales to the normal pattern. If checkout breaks or sales drop in a strange way, I get an alert.

BEFORE A customer emailing to say the site is broken.

STEP 1 · STARTER PROMPT use it alone to try this by hand today

```
Run one check of [STORE URL]: load the homepage, open a product page, add to cart, and reach checkout. Never submit an order or payment. Using the order totals I provide, compare the last 4 hours with the same window over the past 4 weeks. Report failures or a drop above [X]% here, with evidence. If browser access or data is missing, say which check you could not complete.
```

STEP 2 · AUTOPILOT PROMPT paste into the same chat once step 1 works, to make it hands-free

```
You are my automation engineer. Goal: make the revenue leak detector run automatically within the approval limits below. The job is the one I gave you earlier in this conversation. If you don't see it above, stop and ask me to paste the starter prompt first. Work with me one step at a time and wait for my answer before moving on.
1) Ask which tools I use for: my website (as a test shopper), store order data, and the alert channel I check fastest.
2) For each one, check whether you already have a connector or integration for it. If you don't, walk me through connecting it: where to click, which permissions to grant (the minimum needed), and then prove it works by pulling one real record. Never ask me to paste a password or API key into this chat; use the tool's official connect or sign-in flow.
3) Set the trigger: every 4 hours.
4) Set the delivery: silence when healthy, an immediate alert when something breaks.
5) Lock in the guardrail: use test mode; never place a real order or charge a card.
6) Confirm the schedule and time zone in a persistent scheduler, not just this chat. Run one test, verify its actual output, and prevent duplicate actions on retries.
7) Give me a one-page runbook: connected tools, trigger, delivery, failure alerts, and how to pause it. If a required capability is missing, say what is blocked; do not claim it is running.
```

17 Competitor price watch.

Every Monday morning a job checks competitor pricing on comparable products and reports what changed.

BEFORE Checking by hand a few times a year.

STEP 1 · STARTER PROMPT use it alone to try this by hand today

```
Check the current price once for each of these competitor products: [LIST WITH URLS]. Compare to last week's prices in the attached sheet. Report only what changed: product, old price, new price, percent change, and any new promotion you see on the page. Show the proposed sheet updates here; do not edit it yet. State facts only. No commentary about the competitors.
```

STEP 2 · AUTOPILOT PROMPT paste into the same chat once step 1 works, to make it hands-free

```
You are my automation engineer. Goal: make competitor price watch run automatically within the approval limits below. The job is the one I gave you earlier in this conversation. If you don't see it above, stop and ask me to paste the starter prompt first. Work with me one step at a time and wait for my answer before moving on.
1) Ask which tools I use for: web browsing and a spreadsheet.
2) For each one, check whether you already have a connector or integration for it. If you don't, walk me through connecting it: where to click, which permissions to grant (the minimum needed), and then prove it works by pulling one real record. Never ask me to paste a password or API key into this chat; use the tool's official connect or sign-in flow.
3) Set the trigger: every Monday at 6 AM.
4) Set the delivery: a changes-only report to me and the sheet updated.
5) Lock in the guardrail: public pages only; facts, no commentary.
6) Confirm the schedule and time zone in a persistent scheduler, not just this chat. Run one test, verify its actual output, and prevent duplicate actions on retries.
7) Give me a one-page runbook: connected tools, trigger, delivery, failure alerts, and how to pause it. If a required capability is missing, say what is blocked; do not claim it is running.
```

18 Inventory and reorder checks.

Nightly inventory checks flag what's running low against how fast it's selling, so reorders happen before stockouts.

BEFORE Noticing when we ran out.

STEP 1 · STARTER PROMPT use it alone to try this by hand today

```
Here is current inventory by SKU and the last 60 days of sales by SKU.
Calculate days of stock remaining for each SKU at the current sales rate. My
supplier lead time is [X] days. List every SKU that will run out before a new
order could arrive, how many to reorder to cover [Y] days, and any SKU with
more than 180 days of stock. [PASTE DATA]
```

STEP 2 · AUTOPILOT PROMPT paste into the same chat once step 1 works, to make it hands-free

```
You are my automation engineer. Goal: make inventory and reorder checks run
automatically within the approval limits below. The job is the one I gave you
earlier in this conversation. If you don't see it above, stop and ask me to
paste the starter prompt first. Work with me one step at a time and wait for
my answer before moving on.
1) Ask which tools I use for: inventory or warehouse system and sales
history.
2) For each one, check whether you already have a connector or integration
for it. If you don't, walk me through connecting it: where to click, which
permissions to grant (the minimum needed), and then prove it works by pulling
one real record. Never ask me to paste a password or API key into this chat;
use the tool's official connect or sign-in flow.
3) Set the trigger: every night.
4) Set the delivery: a reorder list to me and whoever places purchase orders.
5) Lock in the guardrail: it recommends quantities; a person places every
order.
6) Confirm the schedule and time zone in a persistent scheduler, not just
this chat. Run one test, verify its actual output, and prevent duplicate
actions on retries.
7) Give me a one-page runbook: connected tools, trigger, delivery, failure
alerts, and how to pause it. If a required capability is missing, say what is
blocked; do not claim it is running.
```

19 An AI project manager.

A project-manager agent runs a daily standup summary, pings owners when a project stalls, tracks deadlines, and sends me a Friday digest of what moved and what didn't.

BEFORE Me, chasing.

STEP 1 · STARTER PROMPT use it alone to try this by hand today

```
Act as my project manager. Here is my project list with owners, deadlines,
and last update date. For this one-time trial, write a standup summary: what
moved yesterday, what is stalled (no update in 5+ days), and what is due this
week. Draft a polite nudge to the owner of each stalled project for my
approval. Then write a draft weekly digest from the history I provide. Do not
schedule or send anything.
```

STEP 2 · AUTOPILOT PROMPT paste into the same chat once step 1 works, to make it hands-free

```
You are my automation engineer. Goal: make the AI project manager run
automatically within the approval limits below. The job is the one I gave you
earlier in this conversation. If you don't see it above, stop and ask me to
paste the starter prompt first. Work with me one step at a time and wait for
my answer before moving on.
1) Ask which tools I use for: project or task tool and the team chat.
2) For each one, check whether you already have a connector or integration
for it. If you don't, walk me through connecting it: where to click, which
permissions to grant (the minimum needed), and then prove it works by pulling
one real record. Never ask me to paste a password or API key into this chat;
use the tool's official connect or sign-in flow.
3) Set the trigger: standup summary every weekday morning, stall check daily,
digest every Friday.
4) Set the delivery: summaries to me; nudges saved as drafts.
5) Lock in the guardrail: nudges to teammates go out only after I approve
them, until I say otherwise.
6) Confirm the schedule and time zone in a persistent scheduler, not just
this chat. Run one test, verify its actual output, and prevent duplicate
actions on retries.
7) Give me a one-page runbook: connected tools, trigger, delivery, failure
alerts, and how to pause it. If a required capability is missing, say what is
blocked; do not claim it is running.
```

20 The system that watches the systems.

Automations break. So other automations check that last night's jobs ran, refresh expiring logins, back everything up nightly, run a practice restore, and do a weekly self-audit. When something fails, I hear about it before it matters.

BEFORE Silent failures I found weeks later.

STEP 1 · STARTER PROMPT use it alone to try this by hand today

```
Here is a list of my scheduled automations and where each one logs its
output. For this one-time trial, use the logs I provide to check each job
against its expected schedule and verify its actual output, not just a
success label. List any failed or overdue jobs, with evidence and the error
message if present. Do not flag weekly jobs just because they did not run
today. Also tell me about any login or API token that expires in the next 14
days. If everything is fine, say so in one line.
```

STEP 2 · AUTOPILOT PROMPT paste into the same chat once step 1 works, to make it hands-free

```
You are my automation engineer. Goal: make the watcher run automatically
within the approval limits below. The job is the one I gave you earlier in
this conversation. If you don't see it above, stop and ask me to paste the
starter prompt first. Work with me one step at a time and wait for my answer
before moving on.
1) Ask which tools I use for: the scheduler or logs for every other
automation, my backup location, and the alert channel I check fastest.
2) For each one, check whether you already have a connector or integration
for it. If you don't, walk me through connecting it: where to click, which
permissions to grant (the minimum needed), and then prove it works by pulling
one real record. Never ask me to paste a password or API key into this chat;
use the tool's official connect or sign-in flow.
3) Set the trigger: every morning at 7:30 AM, plus a weekly self-audit on
Sunday.
4) Set the delivery: one line when healthy, details when not.
5) Lock in the guardrail: it reports and restarts only what I've listed as
safe to restart.
6) Confirm the schedule and time zone in a persistent scheduler, not just
this chat. Run one test, verify its actual output, and prevent duplicate
actions on retries.
7) Give me a one-page runbook: connected tools, trigger, delivery, failure
alerts, and how to pause it. If a required capability is missing, say what is
blocked; do not claim it is running.
```

THE ONE RULE

AI DRAFTS. PEOPLE DECIDE.

In my company, AI never sends a message to a customer, spends money, changes a price, or makes a product claim without a person approving it. It prepares, checks, watches, and reminds. That's where nearly all the time savings are anyway, and it's the reason I trust it.

IF YOU'RE STARTING FROM ZERO, BUILD THESE THREE FIRST

1. **The morning brief (#1).** It forces you to connect your calendar, email, and sales data once. Everything else builds on that.
2. **Call analysis (#6).** If you sell by phone or video, your calls are the most valuable data you own and you're throwing it away.
3. **The watcher (#20).** Build it early. The first silent failure will cost more than the build.

BUILT ONE? STUCK ON ONE?

Reply to any Operator Notes email or write to **hello@erickurbanov.com**. I read every reply. More at **erickurbanov.com**.