

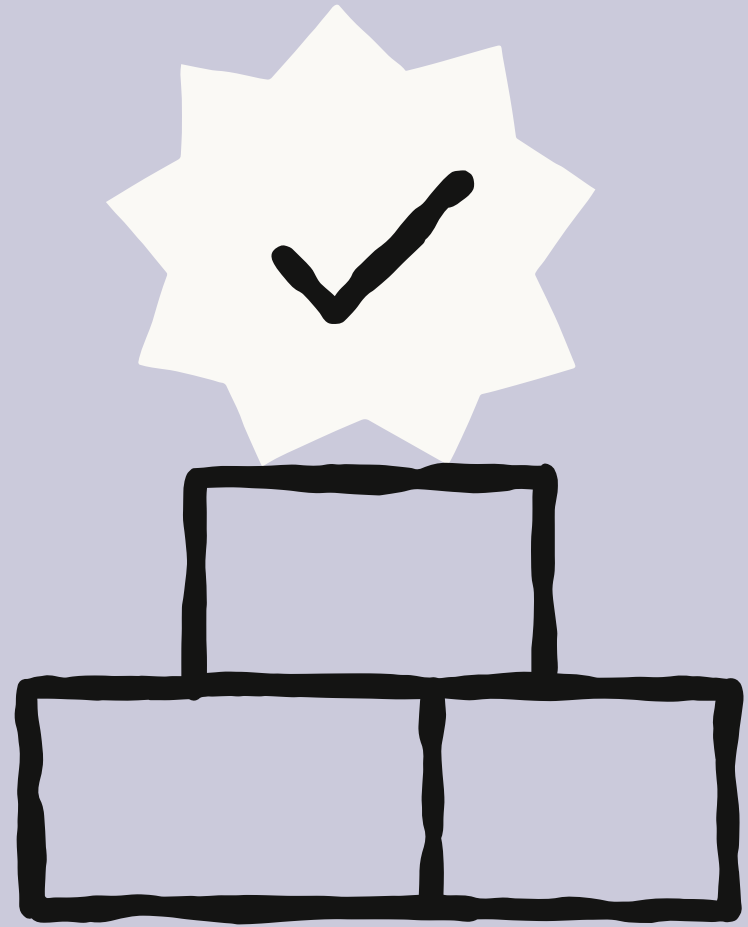
Claude Cowork product guide

Turn your ideas into decks, docs, and more with Claude



Table of contents

What is Claude Cowork?	3
Product overview	5
Getting started	7
Extending Claude Cowork with plugins	10
Popular use cases	13
Best practices and tips	17
Upleveling your work with Claude Cowork	20
Additional resources	22



What is Claude Cowork?

What is Claude Cowork?

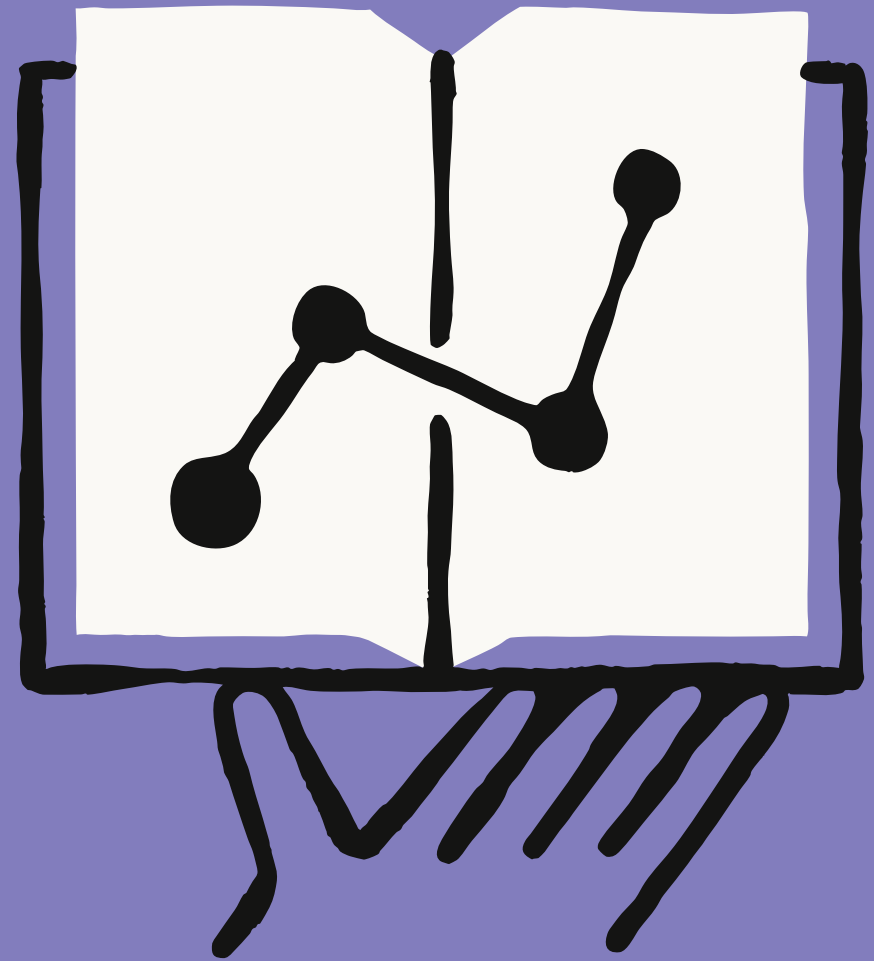
Most AI tools are conversational. You ask a question in a chat window, you get an answer back, and the work of turning that answer into something—a deck, a doc, a spreadsheet, a sent email—is still yours.

Claude Cowork is built for the work, not just the answer. It's a version of Claude that reads and writes files on your computer, uses your connected work apps, and runs multi-step tasks on your behalf. You describe an outcome ("build me a quarterly planning deck," "compare these five vendor proposals"), Claude plans the steps, executes them, and delivers something tangible across your files and tools.

In practice, this looks like:

- **Hours back in your week**, because Claude tackles the context-gathering and tedious work that you'd usually skip or push off.
- **Decisions grounded in everything relevant**, not just the three tabs you happened to have open.
- **Polished output you can trust**, with citations back to the real files and messages instead of plausible-sounding text assembled from nothing.
- **A shorter distance between "I need to do this" and "it's done,"** especially for the recurring work — weekly updates, meeting prep, catch-ups, triage — that never quite justifies its own tool.

This guide is for analysts, PMs, operators, researchers, marketers, recruiters, and anyone else whose day is spent moving information between systems. By the end, you'll have a working understanding of Claude Cowork, seven workflows to try today, and pointers for going deeper.



Product overview

Product overview

Claude Cowork is the place where work gets done. It meets your work where it already lives: local files and folders, connected cloud apps like Slack and Google Drive, and the browser via Claude in Chrome. Claude can also work across Excel, PowerPoint, and Word, carrying context from one to the other so an analysis and the deck that presents it happen in a single session.

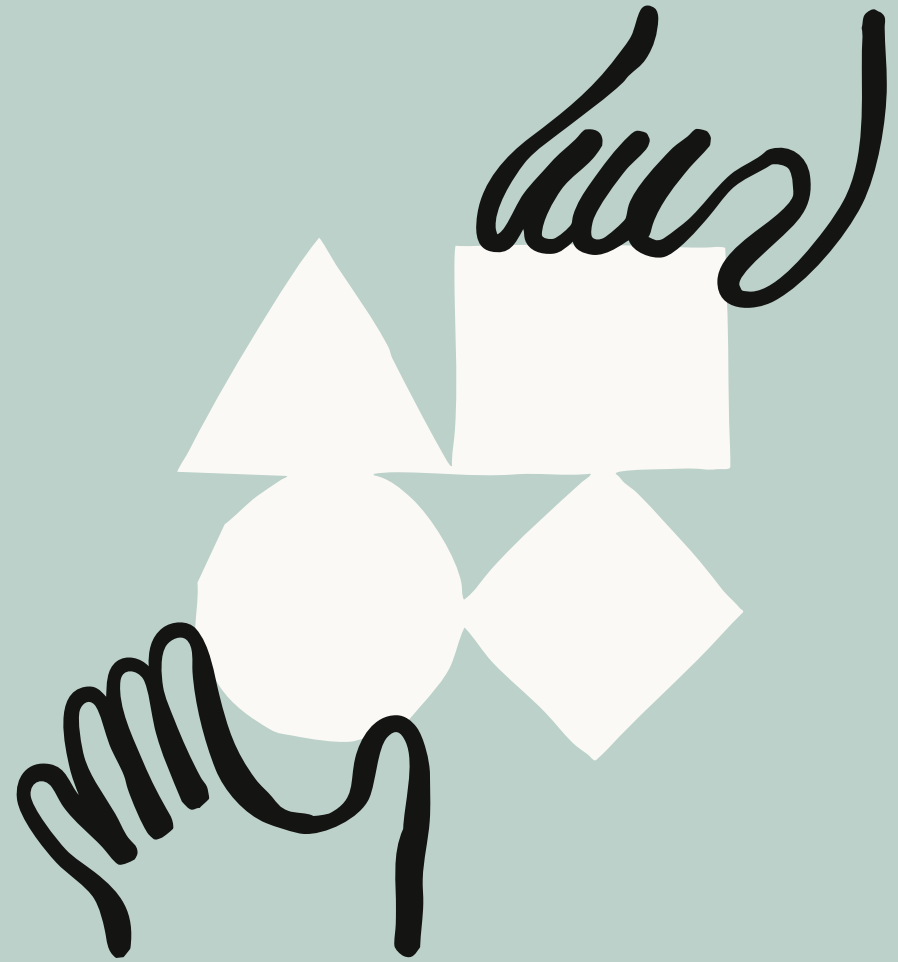
Key capabilities

- **Local file access.** Read from and write to files on your machine directly. No uploading, no downloading, no copying things into a chat window.

Claude product matrix: when to use what

Surface	Best for	Primary users	Where it runs	Example task
Chat	Conversational drafting, research, and analysis in a chat interface	Anyone	Browser, desktop, mobile	"Summarize this report and draft a response."
Claude Code	Agentic coding inside a repo — building, refactoring, testing	Developers	Terminal, IDE	"Refactor this module and run the tests."
Claude Cowork	Cross-app knowledge work that touches files and multiple tools	Knowledge workers (analysts, PMs, operators, researchers, marketers, lawyers)	Claude desktop app	"Read the five vendor PDFs in my Downloads folder, compare them on price and SLAs, and put the result in a spreadsheet."

- **Sub-agents.** Break large tasks into pieces and run them in parallel to streamline work.
- **Real deliverables.** Produce Excel files with working formulas, PowerPoint decks, and formatted documents you can hand off without reformatting. Spreadsheets and slides can be opened and refined further in [Claude for Excel](#) and [Claude for PowerPoint](#).
- **Long-running work.** Work on a task for as long as it takes. Keep the app open and come back to finished work.
- **Scheduled tasks.** Save a task once and run it on demand, or set it to run on a cadence—every Monday morning, every quarter close, every Friday at five.
- **Projects.** Group related tasks into their own workspace with files, context, instructions, and memory that persist across sessions.



Getting started

Getting started

Getting Claude Cowork running is a one-time setup, but the first session is where you start to feel the difference. This section covers what you need installed, how to connect your files and apps, what Claude does (and asks before doing) once a task is underway, and a few starter prompts worth trying in your first ten minutes.

System requirements

- **Claude desktop app.** Claude Cowork runs in the desktop app for macOS or Windows,
- **Paid Claude subscription.** Claude Cowork is available on Pro, Max, Team, and Enterprise plans.
- **Active internet connection** throughout the session.

How to install

1. Install the [Claude desktop app](#).
2. Sign in. If you're on a Team or Enterprise plan, your admin may need to enable Claude Cowork for your org. See the [Claude Cowork IT admin guide].
3. Switch into Claude Cowork mode from the desktop app.

[Screenshot of the Claude Cowork interface]

Setting up your first task

Here are four things to do the first time you open Claude Cowork:

1. **Connect your apps.** Authenticate the MCP connectors you use most — Slack, Gmail, Drive, Asana, GitHub, and others. Claude Cowork will also suggest a connector whenever a task implies one.
2. **Review the permissions model.** You can adjust your preferred permissions model for each chat or in your settings. Even when you are in “Act without asking” mode, Claude Cowork asks for confirmation before sensitive actions like downloads, deletions, and sharing changes. See more guidance [here](#).
3. **[Optional] Grant folder access.** Claude Cowork sees no files until you explicitly connect a folder. You'll be prompted the first time you ask it to work with files, and you can connect more later.

What to expect during a task

When Claude is working on something in Claude Cowork, you can watch it work or step away. It walks through what it's doing at each step and surfaces its reasoning so you can follow along. If it's heading in the wrong direction, you can jump in mid-task and course-correct. For complex work, it may spin up multiple sub-agents running in parallel.

Two safeguards worth knowing about: Claude requires your permission to delete files and has strong guidance to require explicit confirmation when solving captchas, making purchases or engaging in financial transactions, handling sensitive data, deleting files, and modifying system files. Tasks can run for a while depending on complexity—monitor them if you want, or close the window and come back when Claude is done.

Global and folder instructions

You can give Claude standing instructions that apply to every Claude Cowork session. Use this for your preferred tone, output format, or background on your role and team.

To set global instructions:

1. Go to Settings > Claude Cowork in the Claude desktop app.
2. Click "Edit" next to Global instructions.
3. Type your instructions and save.

Folder instructions, which live in the Customize panel in Claude desktop, add project-specific context whenever you select a local folder. Claude can also update them on its own during a session as it learns more about how you work.

What to try in your first ten minutes

Pick one real task you were about to do yourself and hand it to Claude Cowork. Good starters:

- Point it at a folder of PDFs and ask for a one-page brief.
- Ask it to draft a status update from last week's messages in one Slack channel.
- Give it a messy spreadsheet and ask it to clean the columns and produce a chart.

Don't start with a toy prompt. You'll learn more about Claude Cowork's strengths, weaknesses, and what kind of context it needs from one real task than from ten demos.



Extending Claude Cowork with plugins

Extending Claude Cowork with plugins

Out of the box, Claude Cowork can read your files, run code in a sandbox, browse the web, and connect to a growing list of apps through MCP connectors. Plugins go further: they bundle skills, sub-agents, and connectors into a single installable package built around a specific job.

A plugin is a pre-built toolkit for a role or workflow. Instead of wiring up connectors one at a time and re-explaining how your team works every session, you install a plugin and Claude knows the vocabulary, the steps, and the outputs your workflow expects.

What's inside a plugin

A plugin can include any combination of:

- **Skills:** step-by-step playbooks Claude loads automatically when a task matches. A "contract review" skill might tell Claude which clauses to flag, what your standard fallback positions are, and how to format the redline.
- **Subagents:** purpose-built assistants for specific kinds of work. Each runs in its own context window with a custom system prompt, specific tool access, and independent permissions.
- **Connectors:** the MCP integrations the plugin depends on (Salesforce, Asana, Slack, internal tools), bundled so you don't have to hunt for them.

When to reach for a plugin

Plugins earn their keep when your request depends on your org's specific context — your pipeline, your playbooks, your templates, your customers. Good signals:

- You keep re-explaining the same workflow to Claude every session.

- The task touches several tools in sequence (pull from CRM, draft in Docs, send via Slack).
- There's a "right way" your team does this thing, and you want Claude to follow it consistently.

For one-off questions or general knowledge ("what's MEDDIC?", "draft a cold email"), skip the plugin and just ask.

Installing a plugin

Browse available plugins from the [Claude Cowork plugin marketplace](#), or wait for Claude to suggest one when it notices your request would benefit. Installation is a single click: approve the plugin and from there, you can use individual skills or chained skills in a workflow. You can disable or remove a plugin at any time from settings.

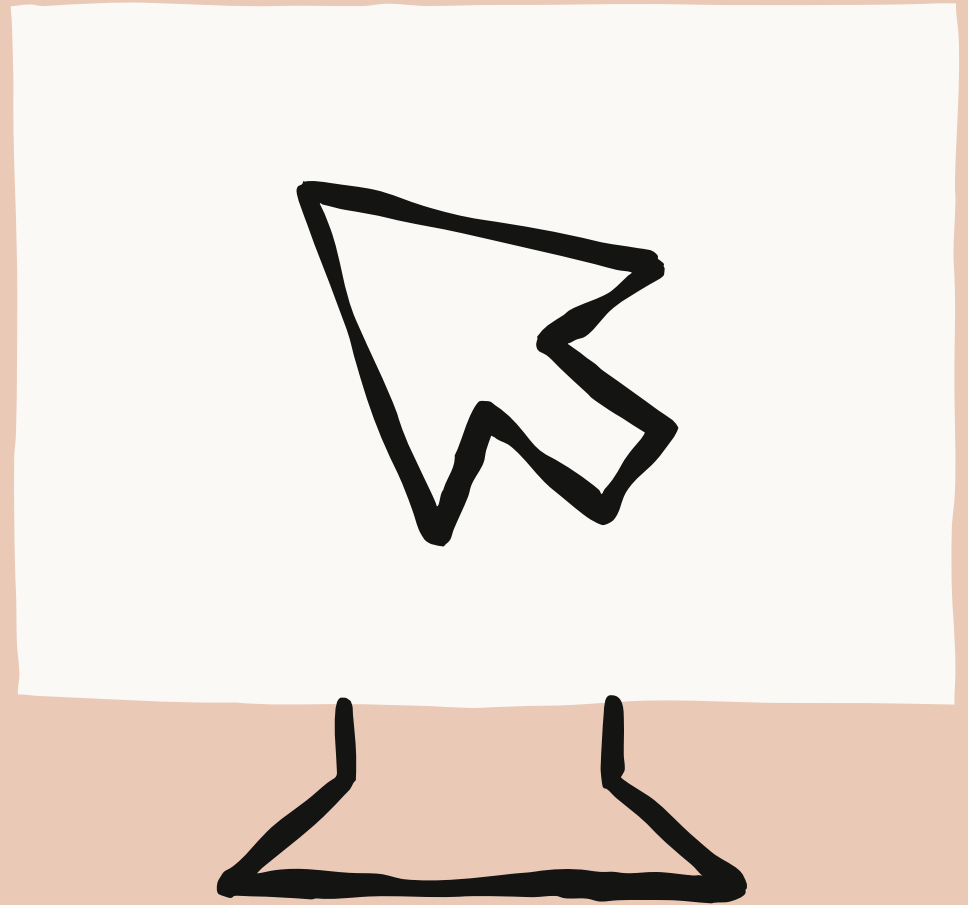
Building your own plugins

If your team has a workflow that doesn't exist as a plugin yet, you can build one. A plugin is a folder that can include skill files (Markdown with instructions), slash commands, subagents for specialized tasks that need their own context and tool permissions, and a manifest listing any MCP servers it depends on. Most teams start with a single skill for their most repetitive task and grow from there—adding a subagent when a job is big enough to warrant its own isolated context. Plugins can stay private to your org or be published to a marketplace for others to install.

A quick example

Say you run customer success and every renewal call follows the same prep ritual: pull the account from Salesforce, check recent Zendesk tickets, skim the last three Gong calls, and drop a briefing into a Google Doc.

Without a plugin, you explain that sequence to Claude every time. With a "Renewal prep" plugin installed, you type `/prep-renewal Acme Corp` and Claude runs the whole chain — connectors already wired up, format already defined, briefing landed in the right folder.



Popular use cases

Popular use cases

In this section, we share seven of the most common ways knowledge workers use Cowork, from research synthesis and meeting prep to inbox triage and catching up after time away. Each follows the same shape: the situation, what to ask Cowork, what you'll get back, and tips to get better results. The named roles are illustrative, but the patterns generalize across a wide range of work.

Turning a messy folder of research into a synthesized brief

A competitive intelligence analyst at a mid-market SaaS company has spent three weeks collecting articles, PDFs, screenshots, and notes about the competitive landscape. She needs a two-page brief for the product leadership meeting tomorrow.

What to ask Claude Cowork

I've connected the `~/Research/competitive-landscape` folder. Read everything in it, then write a two-page brief for our product leads covering: (1) the three most important trends, (2) what each competitor is doing differently, (3) where we have the biggest gaps, and (4) two recommendations. Cite the source file for each claim. Save as `competitive-brief.md` in the same folder.

What you'll get back

A structured markdown brief in the folder you specified, with inline citations tying each point back to a source file. Claude Cowork will usually ask a clarifying question or two — about audience, length, or which sources to prioritize — before starting.

Tips

Tell Claude Cowork the *reader* and the *decision* the brief should support, not just the topic. Ask for citations from the start; retrofitting them is painful. If the folder is large, ask Claude Cowork to share its plan before diving in so you can correct course early.

Preparing for a meeting using scattered context

A partnerships lead has a quarterly review with a customer in 90 minutes. The relevant context is spread across a Gmail thread, two Slack DMs, a co-edited Google Doc, and her notes from the last meeting.

What to ask Claude Cowork

I have a meeting at 2pm with Sarah Chen from Acme about the Q3 partnership review. Pull together: (1) the most recent thread with her in Gmail, (2) our last two Slack exchanges, (3) the shared doc titled "Acme partnership Q3," and (4) my calendar notes from our previous meeting. Give me a one-page prep doc with the three things I should go in knowing and the two open questions to raise.

What you'll get back

A prep doc that threads together the relevant context from each source, with links back to the originals. Claude Cowork will tell you if any source was inaccessible — for example, if Gmail isn't connected yet.

Tips

Name the people, the topic, and the timebox. The more specific you are about *what decision* the meeting is about, the more useful the synthesis. Asking for the "top three" rather than a dump forces better prioritization.

Drafting a recurring report from source files

An engineering manager writes a weekly update every Friday. The content comes from the same places each time: an Asana board, a metrics CSV the data team publishes, and a blockers channel in Slack.

What to ask Claude Cowork

Draft this week's eng update in the format of `~/Reports/weekly-template.md`. Pull shipped items from my Asana "Done this week" section, key metrics from the CSV at `~/Reports/metrics.csv`, and blockers from the `#eng-blockers` Slack channel since Monday. Save as `weekly-update-2026-04-10.md`.

What you'll get back

A filled-in draft matching your template's structure, grounded in real files and real messages.

Tips

Build the template once and reuse it every week. Use **scheduled tasks** to have Claude Cowork start the draft automatically every Friday morning so it's waiting for you.

Inbox triage with context

A chief of staff returns from a long weekend to 120 unread emails. Some need immediate responses, some can wait, some can be archived, and a few need to be escalated.

What to ask Claude Cowork

Go through my unread Gmail from the last 72 hours. Categorize each thread as "needs a response today," "can wait a week," "FYI only," or "escalate to [exec]." For each "needs a response today," draft a two-sentence reply in a separate file I can review before sending. Don't send anything — just draft.

What you'll get back

A triage summary plus a file of draft replies. Claude Cowork will never send email on your behalf without explicit confirmation on each send.

Tips

Give Claude Cowork a quick rundown of the people and projects that matter before it starts triaging. A line like "anything from the board or referencing a sales pipeline dashboard should escalate to me first" helps it sort signal from noise far more accurately than category rules alone. Without that context, Claude has to guess at importance from sender domains and subject lines, which works for obvious cases but misses the nuance.

Building a project plan from a kickoff doc

A marketing ops lead just finished a website-redesign kickoff. The notes are messy, the scope is half-defined, and she needs a plan by tomorrow morning.

What to ask Claude Cowork

Read `~/Projects/website-redesign/kickoff-notes.md`. Turn it into a project plan with milestones, task breakdown under each milestone, owners (use the names mentioned in the notes), a rough timeline assuming a six-week delivery, and a risks section. Output as both a markdown doc and a CSV.

What you'll get back

Two files: a readable plan for humans and a structured CSV for your tracker. Claude Cowork will flag anywhere the kickoff notes were ambiguous rather than inventing details.

Tips

Ask for the ambiguity list explicitly ("flag anything you had to guess"). Resolving five clarifying questions up front is faster than catching invented details later.

Comparing options across documents

A finance analyst has five vendor proposals for a new analytics tool. She needs a head-to-head comparison and a short recommendation for her director.

What to ask Claude Cowork

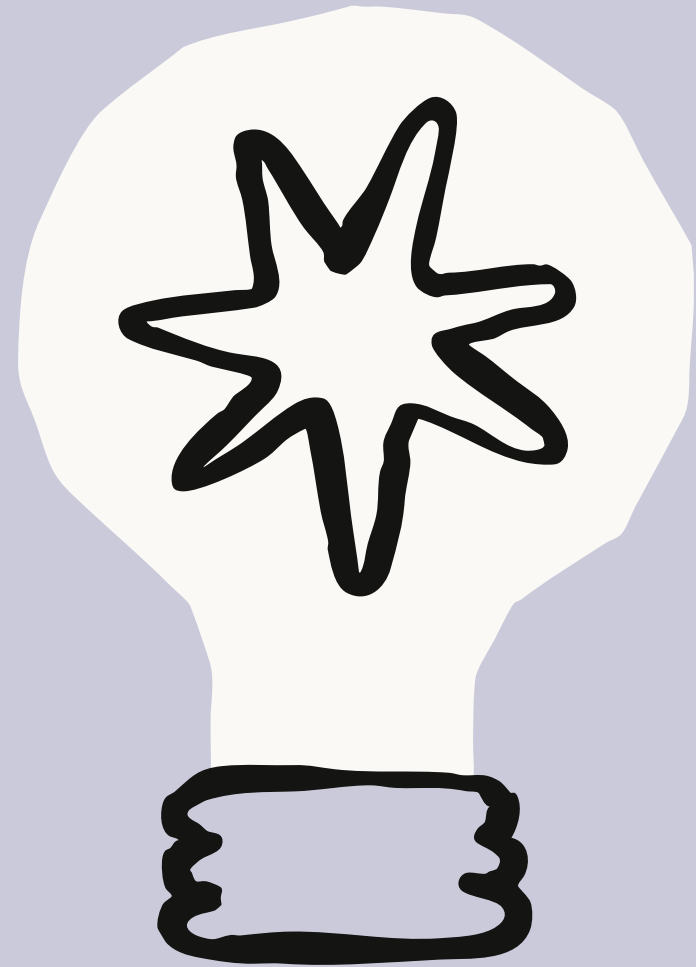
In `~/Vendors/proposals` there are five PDFs from different analytics vendors. Extract from each: price, implementation timeline, what's included, what's extra, SLAs, and references. Produce a comparison table as an .xlsx file and a short written recommendation explaining which two I should shortlist and why.

What you'll get back

A spreadsheet with one row per option and one column per criterion, plus a written rationale. Claude Cowork will note where a proposal was silent on a criterion rather than guessing.

Tips

Define your criteria up front. If you don't, Claude Cowork will pick reasonable ones—but they may not be *your* reasonable ones.



Best practices and tips

Best practices and tips

The tips below are what separate "useful" from "I can't believe how much time this saves me." The people who get the most out of Claude Cowork aren't using a secret feature; they're front-loading context, being specific about what they want, and knowing when to stop and hand off.

Give Claude Cowork the context it needs up front. A good Claude Cowork prompt front-loads context; a bad one makes Claude Cowork guess. Here are some examples of effective (and less effective) prompting:

Research and synthesis

- Less effective: "Summarize what's happening with the pricing project."
- Effective: "Read the docs in my Drive folder 'Pricing 2026' and the last two weeks of #pricing-wg in Slack. Summarize where we've landed on the enterprise tier debate and what's still open. Skip anything older than April 1."

Inbox and messaging

- Less effective: "Find anything important in Slack."
- Effective: "Check #eng-leads, #exec-staff, and any DMs from Priya or Marcus from the last 72 hours. Surface anything where I was @-mentioned, anything tagged urgent, and anything where a decision is pending on me. Ignore standups and bot posts."

Document creation

- Less effective: "Draft a board update."
- Effective: "Draft a Q2 board update using the template at /Templates/Board Update Q1.docx as the structure. Pull metrics from the 'Q2 KPIs' sheet in my Finance folder, and use the narrative from my last three weekly updates in /Updates/2026 for tone and recent context. One page, no jargon."

Calendar and meeting prep

- Less effective: "Help me get ready for tomorrow."
- Effective: "I have a 10am with the Acme Corp team tomorrow. Pull the meeting notes from our last two calls (in /Customers/Acme), the open feedback items tagged 'Acme' in the feedback hub, and any Slack threads from #acme-account in the past month. Give me a one-page brief: where we left off, what they're likely to ask, what I should push on."

Be specific about the output. Format (markdown, docx, xlsx, pdf), length (one page, three bullets, a full brief), audience (your manager, an exec, your team), and tone. "Write a summary" is ten times weaker than "Write a one-page summary for our VP of Product that leads with the recommendation and keeps the background to a single paragraph." Check out our [best practices for prompt engineering](#) to learn more.

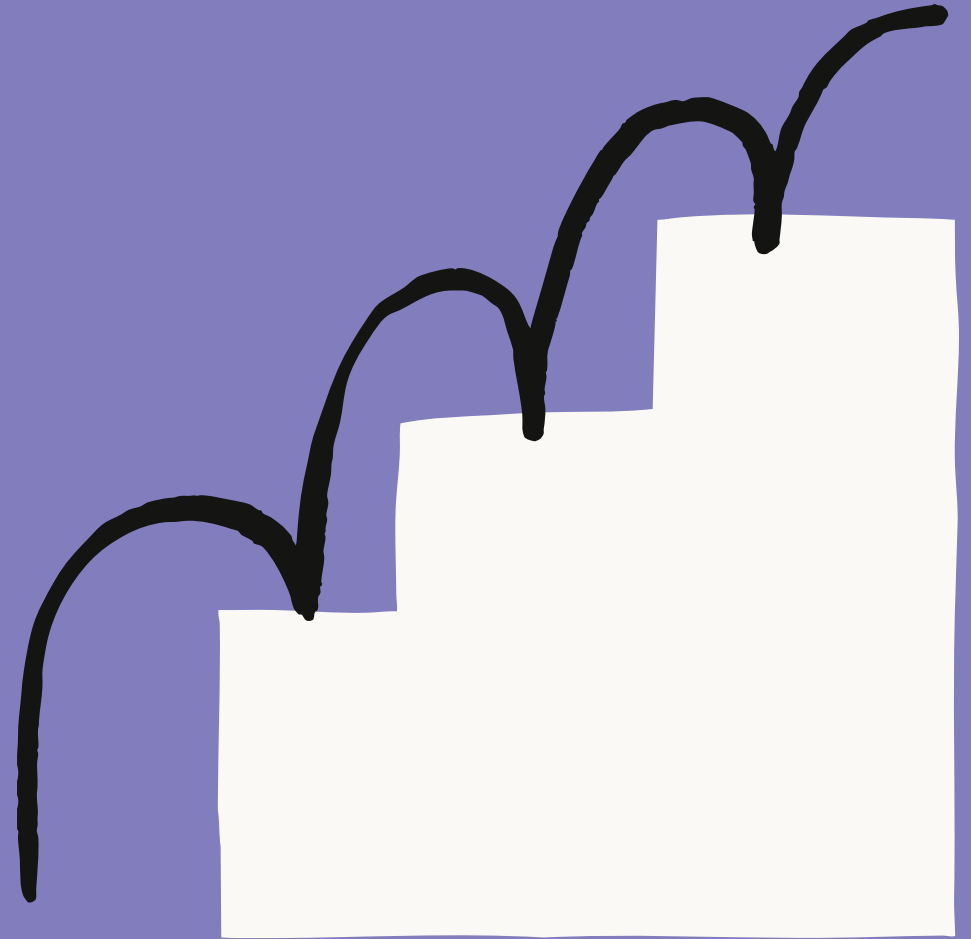
Iterate in place rather than starting over. If the first draft is 80% right, tell Claude Cowork what to change. It remembers the conversation and edits faster than it regenerates from scratch.

Know when to hand work off. Claude Cowork is best for file-and-context work across tools. For production code across a repo, hand off to Claude Code; Claude Cowork itself will offer to launch a Claude Code session when it detects a coding task. For quick factual questions or conversational thinking, Claude.ai is lighter weight.

Review before you ship. Claude Cowork accelerates your work; it doesn't replace your judgment. Read what it produces before sending, publishing, or acting on it, especially anything with numbers, names, citations, or financial implications.

Start with a real task, not a demo. The instinct to test Claude Cowork on something trivial is strong. Resist it. You'll learn more from one real task than from five toy ones.

Use skills early and often. Skills are pre-written playbooks for common deliverables and team workflows, and they materially improve output quality. If your team has recurring workflows, consider building a custom skill for them.



Upleveling your work with Claude Cowork

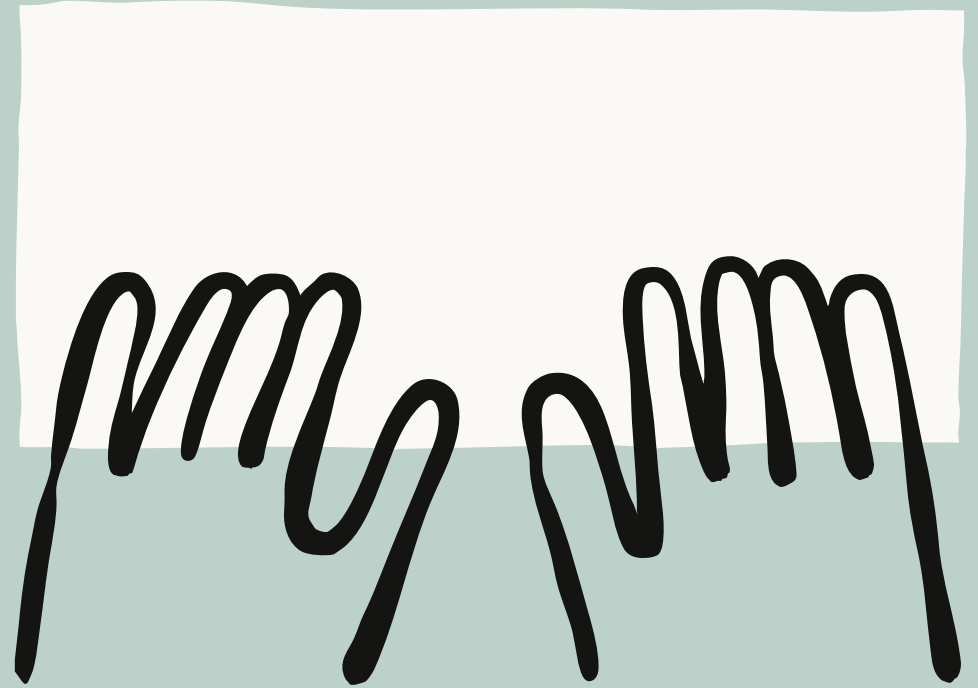
Upleveling your work with Claude Cowork

A weekly update takes ninety minutes, and eighty of them are spent finding things. Meeting prep means opening the invite, the last three docs the attendees touched, and whatever Slack channel the project lives in. Catching up after a day off is its own half-day project. None of this is the work you were hired to do, but it's where the day goes.

Claude Cowork shifts the balance. Once it's reading your folders and connected to your apps, the gathering stops being something you do and becomes something that's already done by the time you sit down. Your mornings open up. The recurring work that never justified its own tool—Friday updates, Monday catch-ups, meeting prep, inbox triage—happens in the background. And the work that needs you gets the version of you that hasn't already spent four hours hunting for context.

Start with one real task this week. Hand it to Claude Cowork and see what comes back. The habits in this guide compound from there.

What will you create?



Additional resources

Additional resources

- **Get started with Claude Cowork** — A step-by-step setup guide covering how to switch into Claude Cowork, assign tasks, and use core features like projects, scheduled tasks, and professional outputs.
- **Let Claude use your computer in Claude Cowork** — How computer use works in Claude Cowork, including per-app permissions, blocked sensitive apps, safeguards, and which workflows to avoid.
- **Introduction to Claude Cowork** — A hands-on course covering the Claude Cowork task loop, plugins and skills, file and research workflows, and how to steer multi-step work responsibly.
- **Use Claude Cowork safely** — Safety guidance for Claude Cowork's agentic capabilities: prompt injection defenses, scheduled task precautions, file access best practices, and recommendations for sensitive workflows.

