

ADVANCED · REFERENCE-GRADE GUIDE

Mastering Claude Cowork

Skills, plugins, multi-agent systems, governance & QA.

The reference volume for power users and rollout owners

This is the reference volume — for power users and the people responsible for rolling Cowork out safely across a team or company. Use it to build repeatable house systems, govern access, and evaluate agent output rigorously.

Deployment help: nexushub.club/books-and-guides.

WHO THIS IS FOR

Power users ready to author Skills and plugins, not just run tasks.

Admins & rollout owners governing access, security, and cost across a team.

Anyone building house systems that need to outlive a single person's habits.

THE COWORK SERIES

Book 1 — Beginner Guide SETUP · PERMISSIONS

Book 2 — Intermediate Guide CONNECTORS · SKILLS

Book 3 — Advanced Guide YOU ARE HERE

Twelve chapters, start to finish

01	Authoring custom Skills (SKILL.md deep dive)	p. 04
02	Plugins & catalogs	p. 06
03	Advanced MCP / connector architecture	p. 08
04	Multi-agent & parallel workflow design	p. 09
05	Building repeatable house systems & SOPs	p. 10
06	Team standardization & governance	p. 11
07	Security, data handling & permissions at scale	p. 13
08	Evaluation & QA frameworks	p. 15
09	Failure modes & recovery playbooks	p. 16
10	Cost & usage optimization	p. 17
11	Advanced pattern library	p. 18
12	Appendices & reference	p. 20

HOW TO USE THIS GUIDE

This is reference-grade material — dip in by chapter as you build. Chapter 12’s appendices are designed to be revisited; items flagged [Unverified] should be checked before you distribute anything built on them.

Authoring custom Skills (SKILL.md deep dive)

1.1 THE MENTAL MODEL

A Skill is "onboarding for a new hire": procedural knowledge Claude loads on demand. It's a directory with a SKILL.md file; it may bundle reference files and executable scripts. Progressive disclosure keeps it efficient across three levels — so a skill's total content is effectively unbounded.

1.2 SKILL.MD ANATOMY

Required frontmatter: name (≤ 64 chars), description (≤ 200 chars) — write as trigger + scope ("Use when the user asks to...").

Optional: dependencies (e.g., python $\geq$ 3.8, pandas $\geq$ 1.5.0).

Body: overview, when-to-use, numbered steps, output format, examples.

Bundled resources: REFERENCE.md, templates, and scripts referenced by name.

EXAMPLE SKILL.MD

```
### Metadata
name: NDA Triage
description: Triage inbound NDAs against our
playbook. Use
when the user asks to review, red-line, or
assess an NDA or
confidentiality agreement.

### When to use
Any inbound NDA. For full contracts, defer to
Contract Review.

### Steps
1. Load references/nda-playbook.md.
2. Classify each clause: Standard, Negotiable,
or Blocker.
3. Produce a red-line table: Clause |
Assessment | Edit.
4. Draft a short cover note to the
counterparty.

### Output
A Word doc "NDA-Triage-[Counterparty]-YYYY-MM-
DD.docx".

### References
- references/nda-playbook.md
- references/fallback-language.md
```

1.3 SCRIPTS & DETERMINISM

Skills can include code Claude runs as tools. Use code for anything deterministic (sorting, parsing, PDF form-filling) — it's cheaper and more reliable than token generation. Claude/Claude Code can install packages from PyPI/npm when loading skills (API Skills require dependencies pre-installed).

1.4 AUTHORIZING BEST PRACTICES

- **Start with evaluation:** run representative tasks, find gaps, build skills to close them.
- **Structure for scale:** split large SKILL.md into referenced files.
- **Think from Claude's perspective:** tune name/description first when a skill doesn't trigger.
- **Iterate with Claude:** capture successful approaches and mistakes into the skill.
- Follow the open Agent Skills standard (agentskills.io) for portability.

1.5 PACKAGING & LIFECYCLE

- Zip with the skill folder as the root.
- Upload via Customize > Skills; test with several trigger prompts.
- On Team/Enterprise, provision org-wide (Personal / Shared / Organization tiers).

VISUAL 1.2 – SKILL AUTHORIZING QA CHECKLIST

- ☐ Description written as trigger + scope
- ☐ Single, focused workflow
- ☐ Examples included
- ☐ References split out into their own files
- ☐ Scripts safe & reviewed
- ☐ No hardcoded secrets
- ☐ Tested against 3+ trigger prompts
- ☐ Validated on the target surface

Ship skills that trigger reliably and behave predictably.

Plugins & catalogs

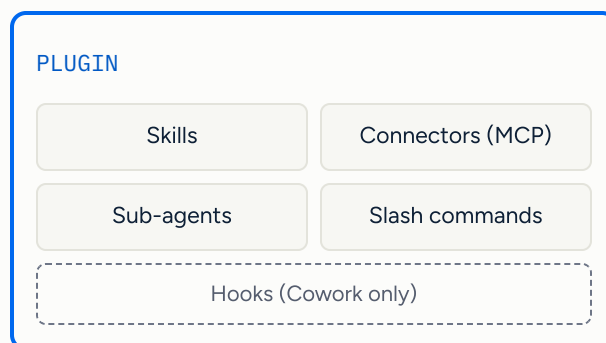
2.1 WHAT A PLUGIN IS

A plugin bundles, per Anthropic's official knowledge-work-plugins README, "the skills, connectors, slash commands, and sub-agents for a specific job function." Cowork's install docs also list hooks as a possible plugin component (scripts that run at defined points; hooks/sub-agents run only in Cowork, not chat). Every component is file-based (Markdown + JSON) — easy to build, edit, and share.

2.2 THE OFFICIAL CATALOGS

Anthropic open-sourced 11 knowledge-work plugins (github.com/anthropics/knowledge-work-plugins). The Feb 24, 2026 enterprise release added HR, Design, Engineering, Operations, Financial analysis, plus partner plugins like Brand voice (Tribe AI). Five finance plugins live in a separate repo (github.com/anthropics/financial-services-plugins), alongside partner plugins S&P Global (Capital IQ Pro), LSEG, and Slack by Salesforce.

VISUAL 2.1 — ANATOMY OF A PLUGIN



Marketplace / Git repo / File → [Install into Cowork](#)

One install turns Claude into a role specialist.

2.3 TWO PLUGIN SHAPES

1. End-to-end process — sequential skills for one workflow (e.g., monthly-close: pull actuals → variance table → board memo).

2. A team's most-used skills bundled (e.g., a finance plugin with variance analysis, modeling, memo drafting, quarterly reports).

2.4 INSTALLING, CUSTOMIZING, BUILDING

Install: Customize > Plugins > Browse; approve connectors; enable/disable components.

From file/Git repo: upload a package or add a marketplace by repo URL.

Customize: Customize > Plugins > [name] > Customize opens a task to tailor it.

Build: use the cowork-plugin-management plugin / /setup-cowork, or Plugin Create from scratch.

PLUGIN LIMITS

200 MB/package (uncompressed) · 5,000 files/package · 500 plugins/marketplace · 25 marketplaces addable · 1 MB in-app preview.

VISUAL 2.2 — THE OFFICIAL PLUGIN CATALOG

PANEL A · KNOWLEDGE-WORK (JAN/FEB 2026)

Productivity, Sales, Customer support, Product management, Marketing, Legal, Finance, Data, Enterprise search, Bio-research, Cowork-plugin-management, +HR, Design, Engineering, Operations, Financial analysis.

PANEL B · FINANCIAL SERVICES REPO

Financial analysis, Investment banking, Equity research, Private equity, Wealth management + partner plugins: S&P Global, LSEG, Slack by Salesforce, Brand voice.

Footnote: Zoom appears in third-party reports but not Anthropic's official Feb 24 posts — [Unverified].

Start from an official plugin; customize to your house.

Advanced MCP / connector architecture

3.1 HOW CONNECTORS RUN

Connectors reach services from Anthropic's cloud infrastructure, not your local machine — true even in Cowork/Desktop. A custom connector (remote MCP server) must be reachable over the public internet from Anthropic's IP ranges. Local MCP servers run only through the desktop app and not in cloud sessions.

3.2 CREDENTIALS & ISOLATION

- Sandbox holds only short-lived, session-scoped tokens (expire within hours).
- Connector authorization tokens never enter the sandbox — calls are made server-side.
- Tenant isolation at the data layer: every record scoped to your org/account.
- Egress enforced outside the sandbox via a mandatory proxy it can't bypass.

WARNING

Network egress permissions don't apply to web fetch, web search, or MCPs (including Claude in Chrome). Owners can disable web search for Cowork/Chat in Organization settings > Capabilities.

VISUAL 3.1 — WHERE A SESSION REACHES

Isolated sandbox (Anthropic servers)

Claude + session-scoped tokens

↓ egress proxy (allow-list only)

Remote MCP connectors (server-side auth)

↔ Anthropic-brokered connection

User's device: local files/browser (Desktop app open)

Local MCP servers: desktop only, never in cloud sessions

Isolation protects your machine; it doesn't change what you've granted.

3.3 DESIGN GUIDANCE

- Prefer first-party connectors; reserve custom MCP for gaps.
- Annotate read-only tools so they're exempt from write-approval gating.
- Keep the connector set minimal per Project to reduce token cost and attack surface.

Multi-agent & parallel workflow design

Cowork can coordinate sub-agents working in parallel. Design for it deliberately.

PATTERNS

Fan-out / fan-in: split independent research streams across sub-agents, then synthesize.

Pipeline: ordered phases where each output feeds the next.

Specialist sub-agents: a research sub-agent, a drafting sub-agent — each in its own context.

Orchestrator + workers: one lead task coordinates; workers handle scoped pieces.

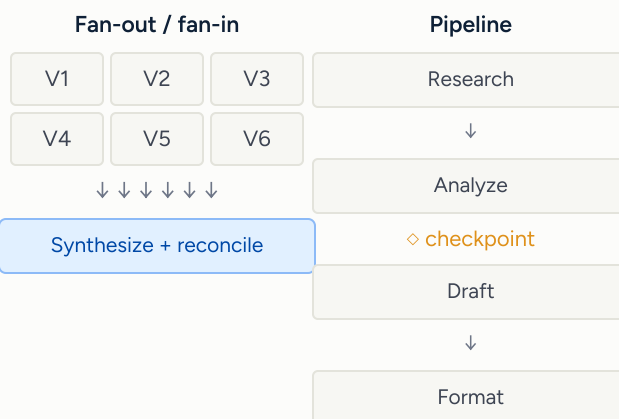
COPY-PASTE PROMPT

Research these 6 vendors in parallel – one sub-agent each.
For every vendor return: pricing, positioning, notable customers, and recent changes (cite sources). Then merge into one ranked comparison table with a recommendation. Flag conflicts between sub-agents' findings rather than silently resolving them.

TIP

Parallelism speeds throughput but multiplies token spend and the chance of inconsistent sub-results. Always add a synthesis + conflict-reconciliation step.

VISUAL 4.1 – FAN-OUT/FAN-IN VS. PIPELINE



Choose parallel for breadth, pipeline for dependency.

Building repeatable house systems & SOPs

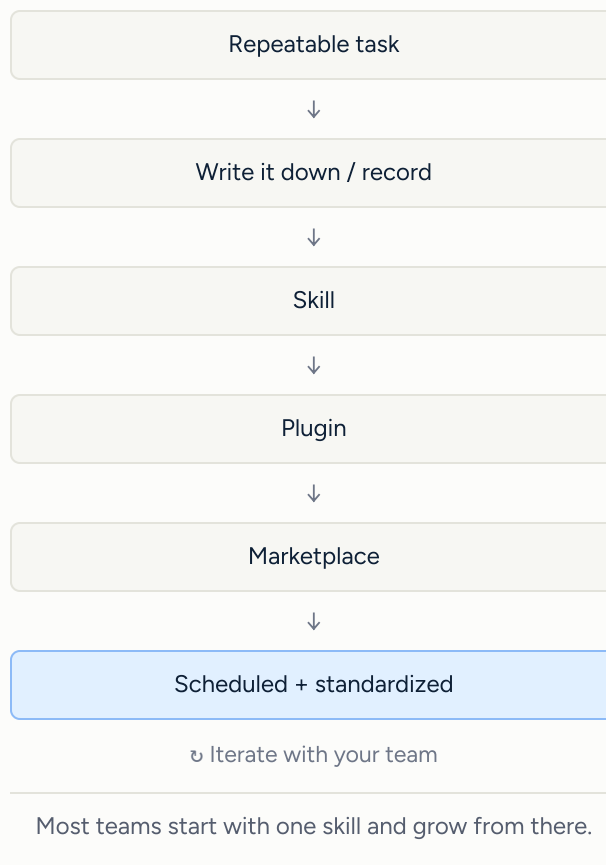
Turn ad-hoc wins into a durable operating system.

1. **Identify** tasks done at least monthly.
2. **Document** how you do them (or record a skill on Mac).
3. **Encode** each as a Skill with a precise trigger description.
4. **Bundle** related skills into a Plugin for the role/process.
5. **Standardize context** with global + project instructions and shared memory files.
6. **Automate** cadence with scheduled tasks.
7. **Distribute** via a plugin marketplace and iterate.

NOTE

A common team practice is to keep skill/plugin source in a Git repo used as a private marketplace. [Unverified]
Some third-party sources describe per-skill version history and JSON export in Settings; confirm current behavior in-product.

VISUAL 5.1 — FROM ONE-OFF TASK TO HOUSE SYSTEM



Team standardization & governance

6.1 ADMIN CONTROLS (TEAM/ENTERPRISE)

- Cowork on/off (Organization settings > Cowork; on by default).
- Cloud sessions toggle (Team: on by default; Enterprise: off by default, granted via custom roles).
- Built-in browser and Claude in Chrome toggles.
- Auto mode availability toggle (on by default).
- "Always allow" for connector write-tools (off by default).
- Plugin marketplaces & per-plugin preferences: Installed by default / Available / Required / Not available.
- Company branding on the Cowork home.
- Skills + Cowork must both be enabled to use plugin marketplaces.

VISUAL 6.1 – GOVERNANCE CONTROL STACK

Org-wide Cowork toggleon

Cloud sessionsTeam on / Ent. off

Browser controlsvaries

Permission-mode availabilityAuto on

Connector write-approvaloff

Plugin marketplace policyper-plugin

Per-group custom roles (Enterprise)most specific

Where each control lives — and its default.

6.2 ACCESS MODEL

The Cowork toggle is org-wide on Team (all-or-nothing). Enterprise can use groups and custom roles for per-team enablement and for the "Run Cowork in the cloud" capability. The most restrictive layer wins; role grants can't override the connector-approval setting.

VISUAL 6.2 – PLUGIN DISTRIBUTION STATES	
Installed by default	Auto-added to every workspace; users can remove it.
Available	Self-service — users browse and install it themselves.
Required	Auto-installed and non-removable.
Not available	Hidden from the catalog entirely.
How admins stage plugins across the org.	

A QUICK GOVERNANCE READ

On Team, one switch governs everyone. On Enterprise, groups and roles let you stage a rollout — pilot with one team on "Available," then flip to "Required" once you trust it. Connector write-approval settings sit outside role grants entirely; no role can loosen them.

Security, data handling & permissions at scale

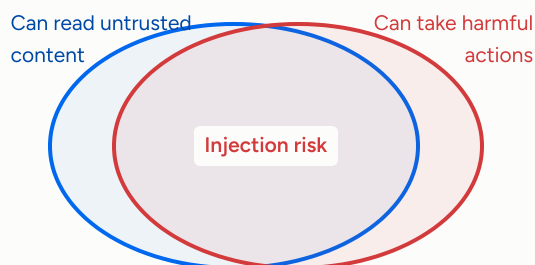
7.1 THE THREAT THAT DEFINES THE MODEL: PROMPT INJECTION

For an injection to succeed, two things must be true at once: Claude can read untrusted content, and Claude can act in a way that harms you. Break either and the attack gets much harder. Anthropic reports strong (not perfect) robustness — e.g., self-reported ~1% attack success against its internal Best-of-N attacker for Claude in Chrome; the PromptArmor PoC (Jan 14, 2026) reused a Files API upload path.

CONTROLS TO COMBINE

- Keep sensitive files in folders Cowork never touches.
- Vet files, skills, and plugins from unknown sources before use.
- Limit browser/web to trusted sites; be deliberate about open tabs in the side panel.
- Prefer Manual mode for high-consequence work; Auto adds screening but isn't a substitute for review.
- Monitor for off-task behavior; stop immediately if it appears; report to Anthropic.

VISUAL 7.1 – THE TWO-CONDITION INJECTION MODEL



Remove either condition to shrink the risk.

7.2 DATA HANDLING & RETENTION

Cloud sessions: saved to the Claude account; captured in the Compliance API.

Local sessions: stored locally, not subject to standard retention policies, and not centrally managed/exportable by admins (Enterprise can retrieve local session content via Compliance API; deletion endpoints aren't available yet).

Task deletion: delete any time; removed from history immediately, deleted from backend storage within ~30 days.

7.3 MONITORING & MDM

OpenTelemetry streams Cowork events (tool calls, file access, approval decisions) to SIEM — but doesn't replace audit logging for compliance.

MDM keys: `sc-camel-is-local-dev-mcp-enabled=false` disables local MCP servers; `sc-camel-is-desktop-extension-enabled=false` blocks MCPB/DXT extensions.

WARNING

As of the source material, Cowork activity was not yet captured in audit logs or the Compliance API for local sessions the same way as cloud sessions. Confirm current coverage before deploying at scale.

VISUAL 7.2 – ENTERPRISE DEPLOYMENT SECURITY CHECKLIST

- ☐ Review egress before enabling
- ☐ Set cloud-session policy
- ☐ Decide Auto-mode availability
- ☐ Keep connector write-approval off unless justified
- ☐ Curate plugin marketplace
- ☐ Stream OTel to SIEM
- ☐ Set MDM keys
- ☐ Confirm Compliance API coverage
- ☐ Publish an acceptable-use SOP

Configure before you invite users in.

Evaluation & QA frameworks

Evaluate agents the way you'd evaluate a new hire's output — against explicit criteria.

FRAMEWORK

1. **Define success criteria** (measurable: accuracy, completeness, format, sourcing).
2. **Build representative test tasks** with known-good answers.
3. **Score** each run: pass/fail per criterion, plus error type.
4. **Trace** failures to a cause (bad description, missing context, ambiguous scope, tool gap).
5. **Fix at the right layer** (prompt, skill description, instructions, or connector).
6. **Regression-test** after each change.

COPY-PASTE PROMPT

Evaluate the deliverable against this rubric and return a scored table:

- Accuracy: every figure matches source (0-5)
- Completeness: all required sections present (0-5)
- Sourcing: each external claim cited (0-5)
- Format: matches the requested structure (0-5)

List every item scoring below 4 and how you'd fix it.

TIP

Not every failure is a prompting problem — sometimes a different model, a connector, or a deterministic script is the right fix.

VISUAL 8.1 — AGENT OUTPUT QA RUBRIC

Accuracy	
Completeness	
Sourcing	
Format	
Safety	

Each scored 0-5; a "how to test" note travels with every row.

VISUAL 8.2 — DIAGNOSING A BAD RESULT

Skill didn't trigger	→ fix description
Wrong scope	→ tighten prompt/instructions
Missing data	→ add connector/context
Inconsistent numbers	→ add verification step
Off-task/unsafe	→ stop, review permissions

Match the fix to the failure.

Failure modes & recovery playbooks

Failure mode	Signs	Recovery
Vague-prompt drift	Claude does the wrong thing confidently	Stop; restate goal with the prompt skeleton; add a checkpoint
Silent number errors	Figures don't match source	Add "verify against source" step; spot-audit; require citations
Prompt injection	Off-task actions, unexpected file/site access	Stop task; revoke permissions; report; quarantine the file
Stalled local session	Progress halts	Ensure Desktop app open & machine awake; use cloud sessions
Usage exhaustion mid-task	Hits cap	Batch/scope tighter; switch simple parts to Chat; monitor Usage
Skill won't trigger	Claude ignores your skill	Rewrite description as trigger+scope; test prompts
Overwritten originals	Source file changed	Always request new copies + change logs; keep /sources read-only
Connector over-broad	Unneeded write access	Set to Needs approval/Blocked; prune unused connectors

WARNING

Treat any off-task behavior as a potential injection until proven otherwise. Stop first, investigate second.

VISUAL 9.1 – INCIDENT RESPONSE

Notice off-task behavior

→

Stop the task

→

Revoke access

→

Quarantine file

→

Report to Anthropic

→

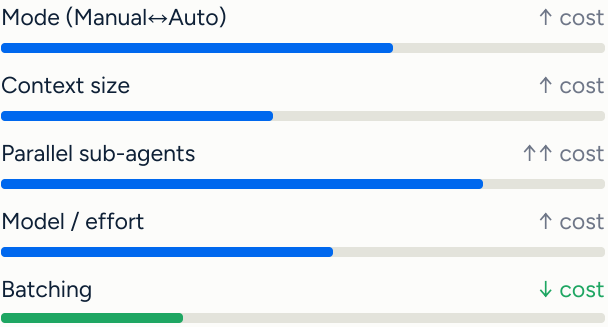
Review scope

A fast, rehearsed stop beats a clever recovery.

Cost & usage optimization

- Chat, Cowork, and Claude Code share one usage pool (rolling 5-hour + weekly caps). Verify current values at claude.com/pricing.
- Auto mode costs more than Manual — use it where hands-off speed pays for itself.
- Right-size the model and effort where those controls exist; trim context to only needed files.
- Batch related work; reserve Cowork for multi-step, file-touching tasks; use Chat for simple ones.
- Schedule heavy recurring work for off-peak if promotional off-peak boosts are active.
- Track team usage via analytics/OpenTelemetry.

VISUAL 10.1 – FIVE COST LEVERS



Five levers that move your usage the most.

Advanced pattern library

A · RESEARCH-TO-DELIVERABLE PIPELINE

Phase 1 verify facts (browser) → Phase 2 analyze (Excel),
pause for review → Phase 3 draft (Word/PowerPoint on
template) → Phase 4 self-verify vs. sources
→ deliver. Cite
everything; flag unverifiable claims.

B · PARALLEL COMPETITIVE SCAN

Fan-out per competitor, fan-in to one ranked table
with conflict flags (see Chapter 4).

C · MONTHLY-CLOSE PLUGIN

Skills: pull actuals → variance table (>threshold flags)
→ board memo; scheduled monthly; human
checkpoint before the memo.

D · CLIENT-DELIVERABLE FACTORY

Project per client; global voice instructions; /sources
read-only, /working, /final; email drafted not sent;
verification pass mandatory.

E · INBOX + CRM TRIAGE

Gmail/CRM connectors set to Needs approval; draft-only outputs; daily scheduled brief with decisions surfaced for the human.

F · LIVE OPS DASHBOARD

Desktop live artifact pulling from connected tools; reopen to refresh; pair with an action ("kick off the follow-up task").

G · ONBOARDING SYSTEM

SOP notes → onboarding Skill → role Plugin → distributed via marketplace → new hires install once.

VISUAL 11.1 – THE PATTERN LIBRARY AT A GLANCE

A · Research pipeline

B · Parallel scan

C · Monthly close

D · Client factory

E · Inbox/CRM triage

F · Live dashboard

G · Onboarding system

VISUAL 11.2 – A QUARTER OF HOUSE-SYSTEM ROLLOUT

WK1–2

Identify tasks

WK3–4

Author first skills

WK5–6

Bundle first plugin

WK7–8

Pilot small group

WK9–10

Governance +
marketplace

WK11–12

Scheduled
automation +
review

A realistic path from pilot to standard practice.

Appendices & reference

12.1 ROLLOUT TIMELINE (VERIFY CURRENT STATUS)

JAN 12

Cowork launches (research preview, macOS, Max)

JAN 16–23

Expands to Pro, then Team & Enterprise

FEB 5–24

Excel/PowerPoint preview; Windows; plugins, admin controls, marketplaces, enterprise connectors

FEB 25–MAR 23

Scheduled tasks; Excel/PPT shared context; Projects, Dispatch; computer use preview

APR 17–20

Claude Design (Anthropic Labs); live artifacts

JUL–AUG

Web + mobile beta, cloud sessions; built-in browser; Chrome side panel full session

12.2 CAPABILITIES QUICK-REFERENCE

Computer use: research preview, Pro/Max, desktop, macOS/Windows; needs the machine awake and app open; slower than connectors; Team/Enterprise excluded at time of source.

Live artifacts: persistent, self-refreshing dashboards; desktop only, local to that machine; paid plans; not synced to web/mobile.

Scheduled tasks: all paid plans; run in the cloud (no device online).

Dispatch: assign tasks from your phone to run on your desktop (needs app open); beta, Pro/Max.

12.3 LIMITS & DEFAULTS QUICK-REFERENCE

Skill metadata: name ≤ 64 , description ≤ 200 . Zip root = skill folder.

Plugins: 200 MB, 5,000 files, 500/marketplace, 25 marketplaces; 1 MB in-app preview.

Connector approvals: "Always allow" for write tools off by default (Team/Enterprise).

Deletion always requires explicit approval.

12.4 SOURCE ANCHORS TO KEEP THIS GUIDE CURRENT

support.claude.com – Cowork collection

claude.com/pricing · Settings > Usage

github.com/anthropics/knowledge-work-plugins

github.com/anthropics/financial-services-plugins

github.com/anthropics/skills ·

agentskills.io

docs.claude.com – prompt-engineering guides

[UNVERIFIED] FLAGS TO RE-CHECK BEFORE DISTRIBUTION

Zoom connector; any specific per-skill version-history/JSON-export UI; exact current usage-limit numbers; whether local-session audit-log coverage has changed.

NEXUS HUB CLOSING

This reference pairs with our Beginner and Intermediate guides as a progression. For a governed, team-wide Cowork rollout — skills, plugins, policy, and QA — reach us at nexushub.club/books-and-guides.

You've mastered Cowork, end to end.

From your first delegated task to authoring Skills and governing a team-wide rollout — you've covered the whole Cowork series. Keep this volume close as a working reference, and revisit the [Unverified] flags before you distribute anything built on them.

BOOK 1

Beginner Guide

Setup, permissions,
first tasks.

BOOK 2

Intermediate Guide

Connectors, Skills,
workflows.

BOOK 3

Advanced Guide

You are here — complete.

[Browse the Nexus Hub library](#)

[NEXUSHUB.CLUB/BOOKS-AND-GUIDES](https://nexushub.club/books-and-guides)