

SEO GUIDE V2

The five things an SEO agency sells every month, as one setup you own.

The audit, the on-page work, the maintenance, the content strategy and the content creation, done by an agent you run yourself. Every step verified against the source documentation, every command ready to paste.

What is in here, and how to read it

This is a reference, not an essay. Read section 1 once. Paste the prompt in section 3 into Claude and let it guide you; sections 4 to 7 hold the same steps in full, with links, for when you want to check something yourself. Come back to sections 8 to 10 every month.

01 What we propose

The claim, why now, and the agency checklist as one table.

02 What we set up

The stack in five rows: agent, data, store, autonomy, you.

03 The setup prompt

Paste it into Claude. It asks where your site is and walks you through everything.

04 Getting the skills

Claude Code, claude-seo and the marketing skills, verified install lines.

05 The eight commands that do the month

What each one does, from the skill files, and when to run it.

06 Setting up the data layers

Google Cloud, Search Console, Analytics, DataForSEO, free backlink sources.

07 Connecting the store

Shopify, WordPress, or a static site on Cloudflare Pages. What the agent may read and write.

08 How the agent does the month

Seven action types and the loop, on a schedule or by hand.

09 The run, end to end

One month on yourstore.com: the prompt, what comes back, what goes live.

10 Appendix

Command list, configuration table, the printable monthly checklist, the prompt library.

HOW WE WRITE

Anything that is a model's estimate or a projection carries the label **est.** in this guide. A figure without the label is a fact we can point to. Where a step depends on your own account (a Google Cloud project, a Shopify app), we name the screen and the permission, and where we could not verify a detail on the day of writing, we say so and link the current documentation.

What we propose

The five things an SEO agency sells every month are one Claude setup: the audit, the on-site optimisation, the maintenance, the content strategy, the content creation. We are not selling you the work. We are selling the connection to a system that does the work, and this guide gives you the system.

Retainers for this work commonly run from the low four figures to the mid five figures a month. The setup in this guide runs on tools that are free or metered by the call, and on an agent that costs what your Claude plan costs. The part that stays with a human is judgment: which pages, which keywords, which changes go live.

Why now

AI overviews and answer engines changed the structure of content, not the quality bar. Unique, trustworthy content still decides who gets cited. What changed is the unit: the AI cites the one paragraph that carries the question and the answer together, while a human reads the whole article in order. Content that serves both is written in short, self-contained passages inside a longer piece that still flows. The tooling in this guide scores exactly that (section 5, the geo command), which is why we treat it as part of the monthly routine and not as a separate project.

The agency checklist

This is the card from the videos. Every line on it maps to a command in section 5 or a data layer in section 6.

BLOCK	WHAT IT COVERS
The audit	Technical crawl, indexing issues, page speed, broken links and redirects, cannibalization, backlink profile.
The on-page	Meta titles and descriptions, headings, internal linking, canonicals, schema, image alt text, LLM feeding.
The maintenance	Content refreshes, Search Console monitoring, indexing requests, rank tracking, rollbacks, monthly report.
The content strategy	Keyword research, search intent, content gaps, striking distance, topic clusters, content calendar.
The content creation	Pillar pages, articles and blog posts, landing pages, product copy, FAQ answers, backlink outreach.

Striking distance means pages already ranking on page two, where a small change moves them onto page one. Cannibalization means two of your own pages competing for the same query. LLM feeding means writing passages an AI can lift and cite in one piece.

What we set up

Five layers. Each one is free or metered, none of them is a subscription to us.

LAYER	WHAT IT IS	WHAT IT DOES FOR YOU
The agent	Claude Code with the <code>claude-seo</code> plugin (Daniel Agrici, MIT, open source) plus the <code>marketingskills</code> repository (Corey Haines).	Runs the audit, the page analysis, the schema work, the clustering, the briefs, the backlink review and the AI-search pass. Writes drafts and diffs. Never publishes on its own unless you tell it to.
The data layer	DataForSEO for SERP and keyword data. Google Search Console, Google Analytics 4 and Google's own page-speed data. Free backlink sources (Common Crawl, Moz, Bing Webmaster).	Gives the agent real numbers instead of guesses: what ranks, what gets clicked, what loads slowly, who links to you.
The store connection	Shopify: a scoped API access for the agent, or the repository if the site is static.	Lets the agent read your pages and propose changes as reviewable edits.
The schedule	The same loop, run every month. By you with the commands in section 5, or by an agent on a timer once you trust it.	Keeps the site maintained: refreshes, indexing checks, rollbacks, the monthly report.
The human	You, or whoever owns the site.	Decides which pages matter, which keywords are worth the fight, and which changes go live. This part does not go away.

How to set up, in order

Seven steps. Steps 1 and 2 take ten minutes. Steps 3 to 5 depend on your accounts and take an afternoon the first time. Step 6 is the first real run. Step 7 is what you keep doing.

- 1 Install Claude Code.** Section 4. One command, then log in.
- 2 Install the skills.** Section 4. Three lines inside Claude Code for `claude-seo`, one line for the marketing skills.
- 3 Create the Google Cloud project and enable the APIs.** Section 6. Search Console, Analytics, PageSpeed, the indexing API. Free.
- 4 Get the DataForSEO credentials.** Section 6. Pay as you go, minimum deposit 50 US dollars.
- 5 Connect the store.** Section 7. Scoped access, every write reviewable.
- 6 Run the first audit.** Section 9. Read the findings before you change anything.
- 7 Set the monthly cadence.** Section 8 and appendix C. Same order every month.

The setup prompt

Copy the dark block and paste it into Claude, in Claude Code or in the Claude app. It asks where your site is hosted, then walks you through the setup one confirmed step at a time. It holds no secrets and never asks you to paste one into the chat.

You are my SEO setup guide and, once the setup is done, my SEO operator. Work like a careful senior consultant: one question at a time, short answers, exact commands, and never a step I have not confirmed.

WHAT WE ARE BUILDING

The five things an SEO agency sells every month, as one setup I own: the audit, the on-page optimisation, the maintenance, the content strategy, the content creation. Tools: Claude Code with the `claude-seo` plugin (github.com/AgriciDaniel/claude-seo, MIT) and the `marketingskills` repo (github.com/coreyhaines31/marketingskills), Google Search Console, Google Analytics 4, Google's PageSpeed data, DataForSEO for SERP and keyword data, and a scoped connection to my site.

HOW TO RUN THIS CONVERSATION

1. First ask me: "Where is your site hosted: Shopify, WordPress, Cloudflare Pages or another git-based static host, or something else?" Then ask whether I am talking to you inside Claude Code (a terminal) or in the Claude app. In Claude Code you run the commands yourself and show me the output. In the app you give me the exact command to run and wait for my paste.
2. Walk me through the steps below in order. After each step, confirm it worked before moving on. When a step depends on my account, name the exact screen and the permission.
3. Never ask me to paste an API key, password or token into this chat. Tell me where the value lives on my machine instead: a config file with owner-only permissions, or an environment variable. Never write a secret into a file that could be committed.
4. Every change to my site is a draft, a diff or a commit for me to review. Nothing is published without my explicit yes.
5. Label every projection or estimate "est.". Keep a changelog: date, page, what changed, why, how to revert.

STEP 1: CLAUDE CODE

macOS, Linux, WSL: `curl -fsSL https://claude.ai/install.sh | bash`
Windows PowerShell: `irm https://claude.ai/install.ps1 | iex`
Check with: `claude --version`. Then run: `claude`, and log in (needs a paid Claude plan).

STEP 2: THE SKILLS (inside Claude Code, version 1.0.33 or newer)

```
/plugin marketplace add AgriciDaniel/claude-seo
/plugin install claude-seo@agricidaniel-claude-seo
/seo setup
```

That creates an isolated Python runtime and Chromium under `~/.claude/skills/seo/`; `/seo doctor` checks it any time. In the terminal, outside Claude Code: `npx skills add coreyhaines31/marketingskills`

STEP 3: CONNECT MY SITE (pick the branch that matches my answer)

Shopify: admin > Settings > Apps > Develop apps > Build apps in Dev Dashboard > Create app. In the app version set the Admin API scopes under Access (start read-only: products, content, themes; add write scopes one at a time), click Release, then Installs > Install app. Token: in the app's Settings copy Client ID and Client secret, then POST `https://STORE.myshopify.com/admin/oauth/access_token` with `grant_type=client_credentials`, `client_id` and `client_secret`. The token lasts 24 hours and only works when app and store belong to the same Shopify organization. Keep id and secret in environment variables.

WordPress: wp-admin > Users > the editor user > Application Passwords > add one and copy it once. Needs WordPress 5.6 or newer and HTTPS. Requests use Basic auth: `curl --user "USER:APP_PASSWORD" https://SITE/wp-json/wp/v2/posts?status=draft`. Everything we write goes in as a draft.

Cloudflare Pages or any git-based static site: the repository is the source of truth. Work on a branch, open a pull request, I read the diff and merge. Never push to the production branch.

Something else: ask me what it is, then find the smallest scoped access that lets you read pages and create drafts.

STEP 4: GOOGLE (free, one Cloud project)

`console.cloud.google.com` > new project. APIs & Services > Library: enable Google Search Console API, PageSpeed Insights API, Chrome UX Report API, Web Search Indexing API, Google Analytics Data API. Credentials > Create Credentials > API key, restricted to PageSpeed Insights, Chrome UX Report and Knowledge Graph. IAM & Admin > Service Accounts > create one > Keys > Add key > JSON, saved as `~/.config/claude-seo/service_account.json`. Search Console: Settings > Users and permissions > Add user > the service account's `client_email`, permission Full. GA4: Admin > Property access management > Add users > the same email, role Viewer; note the property id from Admin > Property details. Write `~/.config/claude-seo/google-api.json` with `service_account_path`, `api_key`, `default_property` ("`sc-domain:SITE`") and `ga4_property_id` ("`properties/NUMBER`"). Run `/seo google setup` and confirm it reports tier 2. Note: Google's Indexing API only accepts pages with JobPosting or BroadcastEvent data. For a normal site use `/seo google inspect`, keep the sitemap clean, and use IndexNow through `/seo bing`.

STEP 5: DATAFORSEO (pay as you go, 50 USD minimum)

`app.dataforseo.com/register`, then API Access in the sidebar: the login is the email, the password is the API password. In the plugin folder run `./extensions/dataforseo/install.sh` and enter both there. Check: `/seo dataforseo serp "MY MAIN KEYWORD"`. Free backlink sources: `/seo backlinks setup` (Common Crawl, Moz, Bing Webmaster).

STEP 6: THE FIRST RUN, THEN EVERY MONTH, IN THIS ORDER

```
/seo audit SITE, /seo technical SITE, /seo page on the five pages with the most impressions (get them with /seo google gsc sc-domain:SITE), /seo schema SITE, /seo cluster "MAIN TOPIC", /seo content-brief "the best gap from the cluster", /seo backlinks SITE, /seo geo on the page with many impressions and few clicks. Stop after every step and show me the findings. Then the review: for every proposed change, the expected effect (est.), what could break, the rollback. At month end: a one-page report from Search Console, GA4 and the changelog, plus a rollback check on every page changed in the last 28 days.
```

Start now with the first question.

The prompt is the compressed version of sections 4 to 7. When Claude and this guide disagree, the current documentation linked in those sections wins, and we would like to hear about it.

Getting the skills

Step 1: Claude Code

Claude Code is Anthropic's agent for the terminal. It needs a Claude Pro, Max, Team, Enterprise or Console account; the free plan does not include it. Install it, then run `claude` once and follow the browser prompt to log in. Verified against the setup page at code.claude.com/docs/en/setup on 02 September 2026.

macOS, Linux, WSL

```
> curl -fsSL https://claude.ai/install.sh | bash
```

macOS with Homebrew

```
> brew install --cask claude-code
```

Windows PowerShell

```
> irm https://claude.ai/install.ps1 | iex
```

Check the install, then start

```
> claude --version
> claude
```

Requirements: macOS 13 or newer, Windows 10 1809 or newer, Ubuntu 20.04 or newer, 4 GB RAM. The native install updates itself in the background.

Step 2: claude-seo

claude-seo is an open-source SEO plugin for Claude Code by Daniel Agrici, MIT licence, public on GitHub at github.com/AgriciDaniel/claude-seo. It runs 25 sub-skills and 18 specialist agents. The plugin install needs Claude Code 1.0.33 or newer. Three lines, typed inside a Claude Code session:

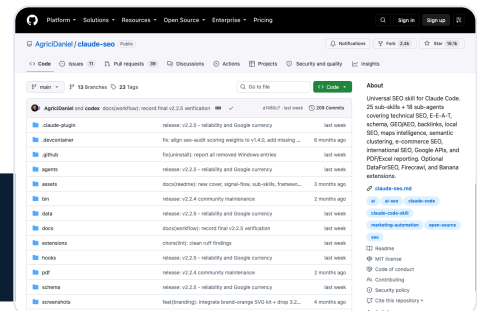
```
> /plugin marketplace add AgriciDaniel/claude-seo
> /plugin install claude-seo@agricidaniel-claude-seo
> /seo setup
```

`/seo setup` creates an isolated Python runtime and installs a Playwright Chromium inside Claude's plugin data (`~/.claude/skills/seo/.venv/`). Nothing global on your machine changes. Chromium is only needed for rendering JavaScript sites and taking screenshots; if that part fails, the rest still works. `/seo doctor` checks the runtime any time without changing anything.

```
> /seo doctor
```

Manual install, if you prefer a clone (Python 3.10 or newer needed):

```
> git clone --depth 1 https://github.com/AgriciDaniel/claude-seo.git
> bash claude-seo/install.sh
```



github.com/AgriciDaniel/claude-seo, captured on 02 September 2026.

Step 2b: the marketing skills

marketingskills by Corey Haines (github.com/coreyhaines31/marketingskills) is the general marketing skill set the videos show next to claude-seo: copywriting, conversion, analytics, ads, launch and pricing skills, 60 and more. The install line from its README, run in your terminal, not inside Claude Code:

```
> npx skills add coreyhaines31/marketingskills
```

To install only a few: `npx skills add coreyhaines31/marketingskills --skill cro copywriting`. To see the list: add `--list`.

The skills inside claude-seo

25 skills at the time of writing. One line each, taken from the plugin's own command table. You will use eight of them every month (section 5); the rest are there when a specific question comes up.

SKILL	WHAT IT DOES
seo	The entry point. <code>/seo</code> plus a sub-command routes to the skill below.
seo-audit	Full website audit with parallel subagent delegation, crawls up to 500 pages, returns a health score.
seo-technical	Technical SEO audit across nine categories.
seo-page	Deep single-page analysis.
seo-schema	Detect, validate and generate Schema.org markup.
seo-content	E-E-A-T and content quality analysis.
seo-content-brief	Detailed content brief: target keywords, outline, internal links.
seo-cluster	SERP-based semantic clustering.
seo-competitor-pages	Competitor comparison page generation.
seo-backlinks	Backlink profile analysis (Moz, Bing, Common Crawl).
seo-dataforseo	Live SEO data through the DataForSEO extension.
seo-drift	SEO drift monitoring with SQLite snapshots: baseline, compare, history.
seo-ecommerce	E-commerce SEO and marketplace intelligence.
seo-flow	FLOW framework prompts, evidence-led.
seo-geo	AI Overviews and Generative Engine Optimization.
seo-google	Google's own APIs: Search Console, PageSpeed, CrUX, Indexing, GA4, PDF reports.
seo-hreflang	Hreflang and international SEO audit and generation.
seo-image-gen	AI image generation for SEO assets (extension).
seo-images	Image optimization analysis.
seo-local	Local SEO: Business Profile, citations, reviews, map pack.
seo-maps	Maps intelligence: geo-grid, Business Profile audit, reviews, competitors.
seo-plan	Strategic SEO planning by site type: saas, local, ecommerce, publisher, agency.
seo-programmatic	Programmatic SEO analysis and planning.
seo-sitemap	Analyze or generate XML sitemaps.
seo-sxo	Search experience optimization: page type, user stories, personas.

Extensions that plug in

Eight optional connectors, each installed with its own script from the plugin folder and each needing its own account: **dataforseo** (live SERP, keyword, backlink and AI-mention data), **ahrefs** (official server, backlinks and organic keywords), **seranking** (AI share of voice across ChatGPT, Gemini, Perplexity, AI Overviews), **profound** (LLM citation tracking), **firecrawl** (full-site crawling), **unlighthouse** (multi-page Lighthouse, runs locally), **bing-webmaster** (Bing Webmaster Tools plus IndexNow), **banana** (image generation). Section 6 covers the one we use every month, DataForSEO.

The eight commands that do the month

Each description below is taken from the skill file that runs the command, so what you read is what the tool does. Replace `yourstore.com` with your site. Run them inside Claude Code, in the project folder of your site if you have one.

1. The audit

```
> /seo audit yourstore.com
```

Full website SEO audit with parallel subagent delegation. Crawls up to 500 pages, detects the business type, delegates to up to 15 specialists (8 always, 7 depending on what it finds), generates a health score from 0 to 100. Run it on day one, then monthly. The findings land in a folder per domain, one file per category.

2. The technical pass

```
> /seo technical yourstore.com
```

Technical SEO audit across nine categories: crawlability, indexability, security, URL structure, mobile, Core Web Vitals, structured data, JavaScript rendering, and the IndexNow protocol. This is the part of the agency audit that used to need a crawler licence.

3. The single page

```
> /seo page yourstore.com/about
```

Deep single-page SEO analysis covering on-page elements, content quality, technical meta tags, schema, images and performance. Use it on the pages that matter most: the five with the most impressions in Search Console, and any page you are about to rewrite.

4. Schema

```
> /seo schema yourstore.com
```

Detects, validates and generates Schema.org structured data as JSON-LD. Product, Organization, FAQ, Article, BreadcrumbList: the markup that makes rich results and AI citations possible.

5. The cluster

```
> /seo cluster "your niche"
```

SERP-based semantic topic clustering for content architecture planning. Groups keywords by actual Google SERP overlap, not by text similarity, designs hub-and-spoke content clusters with internal link matrices, and generates visualizations. This is the content strategy block of the checklist.

6. The brief

```
> /seo content-brief "your topic"
```

Competitive SEO content brief with per-section word counts, competitor scoring, keyword density guidance and page-type templates. Supports both new-page briefs and improve-existing-page briefs. One brief per piece you intend to write this month.

7. Backlinks

```
> /seo backlinks yourstore.com
```

Backlink profile analysis: referring domains, anchor text distribution, toxic link detection, competitor gap analysis. Works with free APIs (Moz, Bing Webmaster, Common Crawl) and with the DataForSEO extension for richer data. Common Crawl is always available; Moz and Bing need a free sign-up, and `/seo backlinks setup` shows how.

8. The AI-search pass

```
> /seo geo yourstore.com/blog/post
```

Generative Engine Optimization for AI Overviews, ChatGPT web search, Perplexity and other AI-powered search: brand mention signals, AI crawler accessibility, llms.txt compliance, passage-level citability scoring and platform-specific optimization. Run it on the page from step 3 with the most impressions and the lowest click-through rate.

Also from the README

COMMAND	WHAT IT DOES
<code>/seo setup</code>	Create or refresh the isolated Python runtime and Chromium.
<code>/seo doctor</code>	Check runtime readiness without changing the system.
<code>/seo sitemap yourstore.com</code>	Analyze the XML sitemap; <code>/seo sitemap generate</code> builds one.
<code>/seo dataforseo serp "query"</code>	Live Google organic results for a query (extension).
<code>/seo firecrawl crawl yourstore.com</code>	Full-site crawl and URL discovery (extension).
<code>/seo drift baseline yourstore.com</code>	Snapshot the site now; <code>compare</code> later shows what drifted.
<code>/seo google gsc sc-domain:yourstore.com</code>	Search Console clicks, impressions, CTR, position (needs section 6).

Setting up the data layers

The agent is only as good as its numbers. Google's data is free and needs one Cloud project. DataForSEO is metered. The backlink sources are free. Do these once; the credentials live in one folder on your machine.

Google Cloud: Search Console, Analytics, PageSpeed, Indexing

These are the free tools from Google the videos show. The `claude-seo` plugin reads them through the `seo-google` skill. The steps below are from the skill's own setup reference (`skills/seo-google/references/auth-setup.md`) and Google's documentation, checked on 02 September 2026. Inside Claude Code, `/seo google setup` walks you through the same steps interactively.

- 1 **Create a project.** Go to `console.cloud.google.com`, create a new project (any name, for example `seo-agent`).
- 2 **Enable the APIs.** APIs & Services > Library. Enable: **Google Search Console API, PageSpeed Insights API, Chrome UX Report API, Web Search Indexing API, Google Analytics Data API**. Optional: Knowledge Graph Search API. All of them are free to call.
- 3 **Create an API key.** APIs & Services > Credentials > Create Credentials > API key. Under API restrictions, limit it to PageSpeed Insights, Chrome UX Report and Knowledge Graph. This key covers page speed and field data (tier 0 in the skill's terms).
- 4 **Create a service account.** IAM & Admin > Service Accounts > Create Service Account. Open it, Keys > Add Key > Create new key > JSON. Save the file as `~/.config/claude-seo/service_account.json` . The `client_email` inside it is what you add to Search Console and Analytics next.
- 5 **Grant Search Console access.** In Search Console, pick the property. Settings > Users and permissions > Add user. Paste the `client_email` . Permission **Full** for reading (search analytics, URL inspection, sitemaps). **Owner** only if you want to use the Indexing API, see the note below.
- 6 **Grant Analytics access.** In GA4: Admin > Property access management > Add users (the plus icon). Paste the `client_email` , role **Viewer**. Note your property id from Admin > Property details; the skill wants it as `properties/123456789` .
- 7 **Write the config file.** `~/.config/claude-seo/google-api.json` , then let the skill check it.

```
{
  "service_account_path": "~/.config/claude-seo/service_account.json",
  "api_key": "YOUR_GOOGLE_API_KEY",
  "default_property": "sc-domain:yourstore.com",
  "ga4_property_id": "properties/123456789"
}
```

```
> /seo google setup
```

The skill reports a credential tier: 0 with the API key only, 1 with the service account (Search Console, indexing), 2 once the GA4 property id is set (organic traffic, landing pages). Aim for tier 2.

A NOTE ON THE INDEXING API, SO YOU ARE NOT SURPRISED

Google's Indexing API is free and the skill supports it (`/seo google index <url>` , 200 publish requests a day by default), but Google restricts it to pages carrying JobPosting or BroadcastEvent structured data. For a store or a blog the honest path is: keep the sitemap clean (`/seo sitemap`), check a page's index status with `/seo google inspect <url>` , request indexing in the Search Console UI when it matters, and use IndexNow through the Bing extension (`/seo bing`) for Bing and the engines that share it. Current wording at developers.google.com/search/apis/indexing-api/v3/quota-pricing.

DataForSEO and the free sources

DataForSEO

DataForSEO sells raw SERP, keyword, backlink and on-page data by the call. It is what replaces the agency's tool subscriptions. Pay as you go, minimum payment 50 US dollars, no monthly fee. Prices from the DataForSEO pricing pages on 02 September 2026 (they change, check dataforseo.com/pricing before you budget):

DATA	STANDARD QUEUE	LIVE
Google SERP, one query, 10 results	0.0006 USD (0.60 per 1,000)	0.002 USD (2.00 per 1,000)
Keyword search volume, one task of up to 1,000 keywords	0.06 USD	0.09 USD

For a store with a few hundred pages, one full monthly cycle of the eight commands with DataForSEO enabled tends to stay in the single-digit dollars **est.**; the plugin's own setup notes list typical per-call costs between a tenth of a cent and five cents. Setup, from the extension's documentation:

- 1 Register at app.dataforseo.com/register and verify the email. New accounts come with a small trial balance.
- 2 Log in, open **API Access** in the left sidebar. Your API login is your registered email; the API password is the one set during registration.
- 3 Run the extension installer from the plugin folder. It asks for the login and password and writes the MCP configuration for you.

```
> ./extensions/dataforseo/install.sh
```

```
> /seo dataforseo serp "your main keyword"
```

Other commands the extension adds: `keywords`, `volume`, `difficulty`, `intent`, `trends`, `backlinks`, `competitors`, `ranked`, `intersection`, `traffic`, `onpage`, `ai-mentions`. Once installed, the audit and the backlinks command use the live data on their own.

Free backlink sources

The backlinks skill works without any paid data. Common Crawl is always available and gives domain-level link graph metrics. Two free sign-ups add depth: **Moz** (Domain Authority, Page Authority, spam score, linking root domains) and **Bing Webmaster Tools** (inbound links with anchor text for a site you have verified). The skill shows the exact steps:

```
> /seo backlinks setup
```

Optional extensions

Only if you already pay for them: **Ahrefs** (official server, your account token), **SE Ranking** (AI share of voice), **Profound** (LLM citations over time), **Firecrawl** (full-site crawl for very large sites), **Unlighthouse** (Lighthouse on every page, runs locally, free). Each has an `install.sh` under `extensions/` and a setup document next to it.

WHERE THE CREDENTIALS LIVE

Everything goes to `~/.config/claude-seo/` with owner-only file permissions. Nothing is sent to us. If you ever hand the machine to someone else, delete that folder and rotate the keys in Google Cloud and DataForSEO.

Connecting the store

The analysis commands only need a URL. The moment the agent should change something, it needs scoped access to the store. One rule for all three platforms: every write is a reviewable change, a draft, a commit, or a scheduled edit, until you have watched the agent long enough to trust it.

Shopify

Shopify's current way is a custom app in the Dev Dashboard (verified on 02 September 2026 at help.shopify.com/en/manual/apps/install-setup-apps). Apps created before 1 January 2026 could reveal a token in the admin; new ones use a client-credentials token instead.

- 1 In the Shopify admin: **Settings > Apps**, then **Develop apps** and **Build apps in Dev Dashboard**. Click **Create app** and name it, for example `seo-agent`. Staff need the permission App development > Develop.
- 2 Create a version: in the **Access** section enter the Admin API scopes the agent needs. For SEO work: read and write products, read and write content (pages and blogs), read and write themes only if the agent is allowed to touch templates. Exact scope strings are listed in Shopify's access scope reference; start with read scopes and add write scopes one at a time.
- 3 Click **Release**, then in the app's **Installs** section **Install app** on your store.
- 4 Get a token: in the Dev Dashboard open the app, **Settings**, copy **Client ID** and **Client secret**. The token comes from one request and lasts 24 hours (Shopify: `expires_in` is always 86399), so the agent requests a fresh one when it starts:

```
curl -X POST https://YOURSTORE.myshopify.com/admin/oauth/access_token \
  -d client_id=YOUR_CLIENT_ID -d client_secret=YOUR_CLIENT_SECRET \
  -d grant_type=client_credentials
```

The client-credentials grant only works when the app and the store belong to the same Shopify organization, which is the case for your own store. Keep the client secret in your shell environment (we use `SHOPIFY_STORE`, `SHOPIFY_CLIENT_ID`, `SHOPIFY_CLIENT_SECRET`), never in a file the agent could commit. **Reads:** products, collections, pages, blog articles, theme files. **Writes:** titles, descriptions, meta fields, article drafts. Publish stays a human click until you decide otherwise.

WordPress

WordPress speaks REST out of the box. Create an **Application Password** for a user with editor rights: wp-admin > Users > edit that user > Application Passwords, give it a name, copy the password once. Needs WordPress 5.6 or newer and HTTPS. Requests use Basic authentication with the username and that password (verified at developer.wordpress.org/rest-api/using-the-rest-api/authentication/).

```
curl --user "USERNAME:APPLICATION_PASSWORD" https://yourstore.com/wp-json/wp/v2/posts?status=draft
```

Yoast or RankMath can stay: they are where the meta title and description are stored and shown, and the agent writes the same fields. Do not remove anything in month one; let the changelog show you what the agent covers first. **Reads:** posts, pages, media, categories. **Writes:** drafts and scheduled posts, meta fields. Publishing stays with you.

A static site on Cloudflare Pages

If the site is a repository (Astro, Hugo, Next, anything that builds to files), the repository is the source of truth. Connect it once in the Cloudflare dashboard: Create application > Pages > Connect to Git, choose the production branch (usually `main`), set the build command and the build output directory. Every push to the production branch deploys; every other branch gets a preview URL. The agent works in a branch, opens a pull request, you read the diff, you merge. That is the cleanest review loop of the three. **Reads:** the whole repository. **Writes:** commits on a branch. Nothing reaches production without a merge.

How the agent does the month

Once the skills run on a schedule instead of on demand, the work settles into a loop. An agent looks at every page against the queries the page could serve and files each finding as one of seven action types. Whether a scheduled agent does this or you do it by hand once a month, the types and the order are the same.

ACTION	WHAT IT MEANS
Improve	Titles, content, meta descriptions, tags on a page that ranks but underperforms.
Write	New articles for keywords the site has no page for yet.
Agitate	Refresh content that is getting old, every 3 to 6 months, even for pages that perform. Search engines reward pages that are visibly maintained.
Cannibalization	Two similar pieces compete: update one, remove one, or set a canonical.
Off-brand	Content that does not fit the brand any more. Flagged for a human, never changed on its own.
Indexing	In the sitemap but not indexed. Submit, and check again next run.
Rollback	A change did not improve performance in its window. Revert to the previous version, which is kept for every edit.

Autonomy, if you automate it. Let an agent act on its own only above a confidence threshold and within daily limits you set, for example a handful of refreshes a day. Every edit is listed as manual or automatic with the new title, what changed, and a link to the page. Any figure for potential clicks or impressions is a projection **est.** and stays labelled that way.

None of this is plug-and-play in the sense that a subscription does the work for you. The system does the work. Somebody still has to want the outcome and read the changelog.

The loop, by hand, once a month

The eight commands in this order give you the full cycle. Do not change anything until the review step; the point of the order is that later steps read the output of earlier ones.

- 1 `/seo audit` : the month's baseline and health score.
- 2 `/seo technical` : fix what blocks crawling and indexing first; everything else is wasted while a page cannot be read.
- 3 `/seo page` on the top five pages by impressions, from Search Console.
- 4 `/seo schema` : apply the generated JSON-LD where it is missing.
- 5 `/seo cluster` on your main topic: this decides what gets written.
- 6 `/seo content-brief` for the one gap with the best intent match.
- 7 `/seo backlinks` : toxic links to disavow, gaps to pursue.
- 8 `/seo geo` on the page with many impressions and few clicks.
- 9 **Review.** Read every proposed change with prompt B from the appendix. Approve, edit, or reject. Log what went live.

The run, end to end

One month on yourstore.com, a Shopify site. This is the shape of a real run: the prompt that starts it, what each step returns, and what reaches the site. Where you see no number, that is on purpose. Your numbers come from your run.

The monthly prompt you paste into Claude Code

```
You are running this month's SEO cycle for yourstore.com (Shopify).
Work in this order and stop after every step to show me the findings before you change anything:
1. /seo audit yourstore.com
2. /seo technical yourstore.com
3. /seo page on the five pages with the most impressions (pull them with /seo google gsc sc-domain:yourstore.com)
4. /seo schema yourstore.com
5. /seo cluster "OUR MAIN TOPIC"
6. /seo content-brief "the one gap from step 5 with the best intent match"
7. /seo backlinks yourstore.com
8. /seo geo on the page from step 3 with the most impressions and the lowest CTR
Rules: propose every change as a diff or a draft, never publish. Label every projection "est.".
Keep a changelog with date, page, what changed, why, and how to revert.
```

What the audit returns

A folder named after the domain with a health score (0 to 100) and one findings file per category: technical, content, schema, performance, visual, and the conditional specialists the site type triggered (e-commerce for a store). Each finding has a severity, the affected URLs, and a fix. The first run usually lists more than you can do in a month; the score tells you where the weight is, because the audit weights technical SEO highest.

The on-page changes it proposes

For each of the five pages: a rewritten title and meta description with the reason (the query it targets, the current position, the gap), missing or duplicated headings, internal links to add with the anchor text, and image alt text. Proposed as a diff per page. You accept some, edit some, and the accepted ones go to the store as drafts or a pull request (section 7).

One content brief

The brief names the target query and the secondary ones, gives an outline with a word count per section, scores the top-ranking competitors, and lists the internal links the new page should carry. With prompt C from the appendix, the agent writes the draft against that brief. The draft is a draft. Somebody who knows the product reads it before it is published.

One geo pass

For the chosen page: a citability score per passage, which passages an AI could lift as a complete answer and which cannot, whether AI crawlers can reach the page, whether `llms.txt` exists, and the rewrite for the three weakest passages so each carries the question and the answer in one paragraph.

The monthly report

One page, produced with prompt E: what changed this month (from the changelog), what moved in Search Console and GA4 (clicks, impressions, CTR, position, organic sessions, before and after), what the rollback rule caught, and the plan for next month. Anything in it that projects forward carries **est.**

WHAT REACHES THE SITE IN MONTH ONE, TYPICALLY

The technical fixes that block crawling. New titles and descriptions on the five pages. Schema where it was missing. One new article from the brief. A disavow list if the backlink pass found toxic links. That is a normal first month. The compounding starts in month two, when the agitate and rollback actions have something to compare against.

The complete command list

COMMAND	WHAT IT DOES
/seo setup	Create or refresh the isolated Python runtime and Chromium
/seo doctor	Check runtime readiness without changing the system
/seo audit <url>	Full website audit with parallel sub-agent delegation
/seo page <url>	Deep single-page analysis
/seo technical <url>	Technical SEO audit across 9 categories
/seo content <url>	E-E-A-T and content quality analysis
/seo content-brief <topic>	Detailed content brief: target keywords, outline, internal links
/seo schema <url>	Detect, validate, and generate Schema.org markup
/seo geo <url>	AI Overviews / Generative Engine Optimization
/seo sitemap <url generate>	Analyze or generate XML sitemaps
/seo images <url>	Image optimization analysis
/seo plan <type>	Strategic SEO planning (saas, local, ecommerce, publisher, agency)
/seo programmatic <url>	Programmatic SEO analysis and planning
/seo competitor-pages <url>	Competitor comparison page generation
/seo local <url>	Local SEO analysis (GBP, citations, reviews, map pack)
/seo maps [command]	Maps intelligence (geo-grid, GBP audit, reviews, competitors)
/seo hreflang <url>	Hreflang / i18n SEO audit and generation
/seo google [command]	Google SEO APIs (GSC, PageSpeed, CrUX, Indexing, GA4, PDF reports)
/seo backlinks <url>	Backlink profile analysis (Moz, Bing, Common Crawl)
/seo cluster <keyword>	SERP-based semantic clustering
/seo sxo <url>	Search Experience Optimization (page-type, user stories, personas)
/seo drift baseline compare history <url>	SEO drift monitoring with SQLite snapshots
/seo ecommerce <url>	E-commerce SEO and marketplace intelligence
/seo flow [stage]	FLOW framework prompts (evidence-led)
/seo firecrawl [command] <url>	Full-site crawling (extension)
/seo dataforseo [command]	Live SEO data (extension)
/seo ahrefs [command] <url>	Backlinks, organic keywords and content data via the official Ahrefs server
/seo seranking [command]	AI share of voice across ChatGPT, Gemini, Perplexity, AI Overviews, AI Mode
/seo profound [command]	LLM citation tracking with time-series data (extension)
/seo bing [command] <url>	Bing Webmaster Tools plus IndexNow URL submission (extension)
/seo unighthouse <url>	Multi-page Lighthouse runner, runs locally (extension)
/seo image-gen [use-case]	AI image generation for SEO assets (extension)
/seo google <sub>	Sub-commands: pagespeed, crux, crux-history, gsc, inspect, inspect-batch, sitemaps, index, index-batch, ga4, ga4-pages, entities, nlp, keywords, volume

Configuration and credentials

What lives where. Names in the first group are set by the plugin and must match exactly; names in the second group are the environment variables we use in our own prompts and scripts, and you can rename them as long as you are consistent.

ITEM	WHERE	USED BY
~/.config/claude-seo/google-api.json	service_account_path, api_key, default_property, ga4_property_id	/seo google, the audit
~/.config/claude-seo/service_account.json	The JSON key from Google Cloud	Search Console, Indexing, GA4
DataForSEO login and password	Entered in ./extensions/dataforseo/install.sh, stored in the MCP configuration	/seo dataforseo, audit, backlinks
Moz and Bing Webmaster keys	Via /seo backlinks setup	/seo backlinks
~/.claude/skills/seo/.venv/	The isolated Python runtime created by /seo setup	Every skill
SHOPIFY_STORE, SHOPIFY_CLIENT_ID, SHOPIFY_CLIENT_SECRET	Your shell environment (our naming)	Store writes on Shopify
WP_URL, WP_USER, WP_APP_PASSWORD	Your shell environment (our naming)	Store writes on WordPress
Git remote and branch	The repository itself	Static sites on Cloudflare Pages

IF SOMETHING DOES NOT WORK

`/seo doctor` for the runtime. `/seo google setup` reports the credential tier and which API failed. "Service account lacks GSC access" means the `client_email` is missing under Settings > Users and permissions. "GA4 property not found" means the property id in the config is wrong; it is shown under Admin > Property details.

The monthly checklist

Print this page. One line per step, in order. Date it, tick it, keep it with the changelog.

WEEK 1: MEASURE

- ☐ /seo audit, note the health score
- ☐ /seo technical, fix crawl and index blockers
- ☐ /seo google gsc, export top pages and queries
- ☐ /seo page on the top five pages
- ☐ /seo drift compare against last month

WEEK 2: DECIDE

- ☐ /seo schema, apply missing JSON-LD
- ☐ /seo cluster on the main topic
- ☐ /seo content-brief for one gap
- ☐ Review proposed on-page diffs (prompt B)

WEEK 3: BUILD

- ☐ Write the article from the brief (prompt C)
- ☐ /seo geo on the high-impression, low-CTR page (prompt D)
- ☐ /seo backlinks, disavow and outreach list
- ☐ Agitate one older page that still performs

WEEK 4: SHIP AND REPORT

- ☐ Approved changes go live as drafts, merges, schedules
- ☐ Rollback check on last month's changes (prompt F)
- ☐ Monthly report (prompt E), one page
- ☐ /seo drift baseline for next month

The prompt library

Ready to paste into Claude Code. Replace the capitalised parts. These are ours; the commands inside them are the plugin's.

A. The monthly run

```
You are running this month's SEO cycle for yourstore.com (Shopify).
Work in this order and stop after every step to show me the findings before you change anything:
1. /seo audit yourstore.com
2. /seo technical yourstore.com
3. /seo page on the five pages with the most impressions (pull them with /seo google gsc sc-domain:yourstore.com)
4. /seo schema yourstore.com
5. /seo cluster "OUR MAIN TOPIC"
6. /seo content-brief "the one gap from step 5 with the best intent match"
7. /seo backlinks yourstore.com
8. /seo geo on the page from step 3 with the most impressions and the lowest CTR
Rules: propose every change as a diff or a draft, never publish. Label every projection "est.".
Keep a changelog with date, page, what changed, why, and how to revert.
```

B. Review one proposed change

```
Here is the proposed change for URL. Before I approve it, answer in five lines:
what is the expected effect (est.), what could break, what is the rollback,
which query does it target, and what is the page's position for that query today.
```

C. From brief to draft

```
Using the brief for "TOPIC", write the article. Keep every section within the brief's word counts.
Put the direct answer to the search query in the first paragraph of the section that carries it,
so the passage stands on its own. No claims without a source we control. Draft only, do not publish.
```

D. The AI-search pass

```
/seo geo URL
Then rewrite the three lowest-scoring passages so each carries the question and the answer in one paragraph.
Keep the facts, change the structure. Show me before and after side by side.
```

E. The monthly report

```
Compile this month's SEO report for yourstore.com from Search Console (/seo google gsc sc-domain:yourstore.com),
GA4 (/seo google ga4-pages) and the changelog: what changed, what moved (before and after, same 28-day windows),
what we roll back, what we do next month. Mark every projection "est.". One page, plain language.
```

F. The rollback rule

```
For every page changed in the last 28 days, compare clicks and impressions with the 28 days before the change.
List the pages that got worse with both figures. Prepare the revert for each one from the changelog.
Do not apply anything until I say so.
```

ABOUT THIS GUIDE

Maintained by Orvus, orvus.net. Every step was checked against the linked documentation on the date on the cover. If something is wrong or out of date, write to service@orvus.net and it gets fixed in the next version.