

Optional Features

The kit ships with 14 optional features that are OFF by default. None are required to ship a working store. Each one solves a specific problem — turn it on when you have that problem.

For each feature: **what it is**, **when you'd want it**, **how to turn it on**, and **what breaks if you turn it on without doing the prerequisite work**.

1. Buy Now Pay Later (BNPL) badge

What it is. A small badge on every product page (PDP) that says "or 4 interest-free payments of \$X — Klarna · Affirm · Afterpay". Renders only on products priced \geq \$35.

When you'd want it. When you're selling products that buyers might be tempted to put off because of the price. The badge is purely informational — the actual BNPL payment methods come from Stripe Hosted Checkout.

Setup.

1. In Stripe Dashboard → Settings → Payment Methods, enable **Klarna** and **Affirm** (one click each)
2. In Vercel env vars: `NEXT_PUBLIC_BNPL_ENABLED=true`
3. Redeploy

What breaks if you turn this on without enabling Klarna/Affirm in Stripe. Buyers see the badge promising 4 payments, then click checkout, then don't see Klarna/Affirm as options because Stripe Checkout only shows what you've enabled. Bad-feeling moment. **Always enable in Stripe FIRST, then flip the env flag.**

2. Cloudflare Turnstile (anti-bot CAPTCHA)

What it is. Cloudflare's invisible CAPTCHA. Adds a challenge to your sign-in page that humans pass automatically and bots fail. Free.

When you'd want it. As soon as you go live. Without this, bots can hammer your sign-in page trying credential-stuffing attacks, fill your contact form with spam, etc.

Setup.

1. Sign up at <https://dash.cloudflare.com> → Turnstile → **+ Add Site**
2. **Domain:** your storefront domain. **Widget mode:** Managed
3. Copy the **Site Key** (starts with `0x4...`) and the **Secret Key**

4. In Vercel env vars:

- `NEXT_PUBLIC_TURNSTILE_SITE_KEY=0x4...`
- `TURNSTILE_SECRET_KEY=...`

5. **Important:** the site key MUST be prefixed with `NEXT_PUBLIC_` — without that prefix, Next.js doesn't expose it to the browser bundle and the widget never renders

6. Redeploy with the build cache UNCHECKED (so the new env value gets baked in)

What breaks. Nothing breaks if you forget — the kit silently skips the CAPTCHA. But spammers and bots are unprotected.

3. Crisp live chat

What it is. A chat widget in the bottom-right corner of every page so buyers can message you in real time. Crisp's free tier covers 2 operators.

When you'd want it. When you want to be reachable for pre-sale questions ("does this work with X?", "do you offer Y?"). Conversion-positive for higher-priced products.

Setup.

1. Sign up at <https://crisp.chat> → create a website
2. Copy the **Website ID** (looks like a UUID)
3. Vercel env: `NEXT_PUBLIC_CRISP_WEBSITE_ID=<uuid>`
4. Redeploy

What breaks. Nothing. The widget just doesn't appear if the env var is unset.

4. Sentry error monitoring

What it is. When the kit throws an unexpected error (database failure, third-party API timeout, etc.), Sentry captures the stack trace + context and sends it to a dashboard you can review. Free tier covers ~5K events/month.

When you'd want it. Always, in production. Without this, you find out about errors when buyers email to complain.

Setup.

1. Sign up at <https://sentry.io> → create a Next.js project
2. Copy the DSN URL (looks like `https://abc123@o12345.ingest.sentry.io/67890`)
3. Vercel env vars (set BOTH — one is server-side, one is client-side):

- `SENTRY_DSN=<your DSN>`
- `NEXT_PUBLIC_SENTRY_DSN=<same DSN>`

4. Redeploy

What breaks. Nothing if missing — errors still print to Vercel logs, you just don't get a Sentry dashboard.

Recommended Sentry alert rules (set in Sentry dashboard, not env):

- Notify when error frequency > 10/hour
- Notify on any error tagged `level:error` or higher
- Notify when a new error type appears for the first time

5. Google Analytics 4

What it is. Standard web analytics — pageviews, sessions, source/medium attribution.

When you'd want it. If you run Google Ads, post on Reddit, write blog posts that link to your store, or otherwise care about traffic source attribution.

Setup.

1. <https://analytics.google.com> → create a GA4 property → grab the Measurement ID (`G-XXXXXXXXXX`)
2. Vercel env: `NEXT_PUBLIC_GA_MEASUREMENT_ID=G-XXXXXXXXXX`
3. Redeploy

What breaks. Nothing if missing. The kit just doesn't load the GA script.

Privacy note: if you're EU/UK or have buyers there, GA4 requires user consent before firing. The kit's cookie consent banner (under Site Editor → Cookie banner) handles this — it gates GA loading on the buyer accepting analytics cookies.

6. Meta Pixel (Facebook / Instagram retargeting)

What it is. Tracking pixel that lets you retarget visitors with Meta ads.

When you'd want it. Only if you plan to run Meta ads for your store. Otherwise skip — there's no benefit to having it loaded.

Setup.

1. Meta Business → Events Manager → create a Pixel → grab the Pixel ID (numeric)
2. Vercel env: `NEXT_PUBLIC_META_PIXEL_ID=1234567890`

3. Redeploy

What breaks. Nothing if missing.

7. Shippo (live shipping rates + label printing)

What it is. Real-time shipping rate quotes from USPS, UPS, FedEx, etc. at checkout. After purchase, admin can buy a label in one click and have it emailed for printing.

When you'd want it. If you sell physical products. Skip if you're 100% digital.

Setup.

1. Sign up at <https://goshippo.com> (free tier: 30 paid labels/month, unlimited rate quoting)
2. <https://apps.goshippo.com/settings/api> → grab API key (test or live; use test for development)
3. Vercel env vars:

- `SHIPPO_API_KEY=shippo_test...` (or `shippo_live...`)
- `SHIPPING_FROM_NAME="Your Store Name"`
- `SHIPPING_FROM_LINE1="123 Main St"`
- `SHIPPING_FROM_CITY="San Francisco"`
- `SHIPPING_FROM_STATE="CA"`
- `SHIPPING_FROM_ZIP="94110"`
- `SHIPPING_FROM_PHONE=""` (optional, helps with international labels)

4. (Strongly recommended) Webhook setup: Shippo Dashboard → Settings → Webhooks → Add endpoint → URL `https://YOUR-DOMAIN/api/webhooks/shippo` → events: `track_updated` + `transaction_created` → copy the signing secret to `SHIPPO_WEBHOOK_SECRET` in Vercel

5. Redeploy

6. For each physical product: edit in admin → set `is_physical = true`, fill in dimensions (length/width/height/weight)

What breaks. Without this, physical products fall back to a flat-rate or manual-label flow (admin uploads a label image manually). Without the webhook secret, the tracking webhook accepts unsigned requests — low risk for a small shop but defense-in-depth recommends turning it on.

8. Anthropic Claude AI (product description generator)

What it is. A "Generate description" button on the admin product form. Click it, the kit asks Claude to write a product description from the product name + tags + price. You edit and save.

When you'd want it. When you have 20+ products to write copy for and want a starting draft instead of a blank page.

Setup.

1. Sign up at <https://console.anthropic.com> → API Keys → create one
2. Vercel env: `ANTHROPIC_API_KEY=sk-ant-...`
3. Redeploy

What breaks. Nothing if missing. The button just doesn't appear in the admin product form.

9. Mandatory admin MFA enforcement

What it is. Hard-blocks `/admin/*` routes unless the admin has completed a TOTP or passkey verification in their session. Prevents admin compromise even if your password leaks.

When you'd want it. ALWAYS, in production. Your admin compromise is the catastrophic failure mode for the store.

Setup.

1. **First**, enroll yourself in either TOTP (Authentication → Multi-Factor → toggle TOTP in Supabase, then `/account/security` → Add authenticator app) OR passkey (`/account/security` → Add passkey)
2. **Then**, log out and log back in. Verify the new factor on login. Confirm `/admin` still loads.
3. **Only then**, set `ENFORCE_ADMIN_MFA=true` on Vercel.
4. Redeploy.

What breaks if you flip the flag without enrolling first. You lock yourself out of `/admin`. Recovery: delete the env var on Vercel (or set to `false`), redeploy, then enroll first.

10. CRON_SECRET (background jobs)

What it is. The kit ships 4 daily background jobs (low-stock check, abandoned-cart recovery, release stale inventory reservations, cleanup old Stripe coupons). They run via Vercel Cron, but Vercel calls a public URL for each one — so they need a shared secret to verify the request is from Vercel and not a random attacker.

When you'd want it. As soon as you have any active products. The release-stale-reservations job is particularly important if you use product variants — without it, expired Stripe sessions hold inventory forever.

Setup.

1. Generate a secret: `openssl rand -hex 32` (or any 32-character random string from a password manager)
2. Vercel env: `CRON_SECRET=<that string>`
3. Redeploy

What breaks. Without this, the cron routes refuse with 403. The app keeps working but the jobs silently skip. After a few weeks you may notice inventory reservations piling up; flip this on to fix.

11. CART_RECOVERY_SECRET (abandoned-cart emails)

What it is. When the abandoned-cart cron emails a buyer to come back, the link in the email is HMAC-signed with this secret. That stops attackers from forging "recovery" links to other people.

When you'd want it. As soon as the abandoned-cart cron is running (i.e. as soon as you set `CRON_SECRET`).

Setup.

1. Generate: `openssl rand -hex 32`
2. Vercel env: `CART_RECOVERY_SECRET=<string>`
3. Redeploy

What breaks. Without this, the kit falls back to `CRON_SECRET` for signing. Works but reuses a secret across two unrelated purposes (good practice is one secret per purpose).

12. DOWNLOAD_LINK_SECRET (signed guest downloads)

What it is. When a guest (non-logged-in) buyer needs to re-download a digital product, the kit can either give them a signed 7-day URL or fall back to a less-strict email-match + time-window check. This env var enables the signed URL path.

When you'd want it. As soon as you have `storage_download` products and accept guest checkout.

Setup.

1. Generate: `openssl rand -hex 32`
2. Vercel env: `DOWNLOAD_LINK_SECRET=<string>`
3. Redeploy

What breaks. Without this, guest downloads use the fallback (still works, just less strict). Logged-in buyer downloads are unaffected — they always go through the auth-protected path.

13. Stripe Tax (automatic US sales tax)

What it is. Stripe calculates and collects US sales tax based on the buyer's billing address.

When you'd want it. Once you exceed economic nexus thresholds in any state. (Most one-person digital-product shops never do — thresholds are typically \$100K+ in revenue or 200+ orders into a state per year.)

Setup.

1. Stripe Dashboard → Settings → Tax → register in the states where you have nexus
2. Vercel env: `STRIPE_AUTOMATIC_TAX=true`
3. Redeploy

What breaks. Don't turn this on until you've actually registered in the states you have nexus in. Otherwise Stripe collects tax you have no legal authority to collect, which is its own problem.

14. A/B theme test

What it is. The kit ships two themes — "Bone & Moss" (Theme A, default) and "Sun & Shadow" (Theme B). Setting this flag enables 50/50 cookie-based bucketing of new visitors so you can A/B test which theme converts better.

When you'd want it. If you want data on which theme to ship as the only theme. Or if you genuinely want to run both forever (visitors get a stable theme via cookie for 180 days).

Setup.

1. Vercel env: `NEXT_PUBLIC_AB_THEME_ENABLED=true`
2. Redeploy

Manual overrides for screenshots / preview:

- `?theme=a` — pin this session to Theme A
- `?theme=b` — pin this session to Theme B
- `?theme=reset` — clear the cookie, get re-bucketed on next request

What breaks. Nothing. Without the flag, all visitors see Theme A (the same as if you flipped the kill switch).

Note: Admin pages and transactional emails always render Theme A regardless of cookie. Only the storefront is bucketed.

A note on env-var hygiene

When you add or change an env var on Vercel:

1. **Save the variable** — Vercel UI sometimes asks if you want to set it for Production / Preview / Development. Pick Production (or Production + Preview if you use preview deploys).
2. **Redeploy** — env changes don't take effect until your next build. Vercel asks "want to redeploy?" — say yes.
3. For **NEXT_PUBLIC_*** vars specifically, Vercel sometimes serves the previous build cache. **Uncheck "Use existing build cache"** on the redeploy dialog so the new value gets baked into the JS bundle.

If you flip a non- `NEXT_PUBLIC_*` env (a server-only secret), no cache concerns — the next request after redeploy uses the new value automatically.

What's NOT in this list (and where to look)

- **Stripe API keys, Supabase keys, Resend key, GitHub token** — those are *required* for the kit to work, see `QUICK_START.md`.
- **Custom email domain** — see `DEEP_DIVE.md` → Email section.
- **Custom domain on Vercel** — Vercel docs at <https://vercel.com/docs/concepts/projects/domains>
- **Theme color customization** — Admin → Settings → Theme (no env var needed)
- **Site copy / legal pages** — Admin → Settings → Site Copy (no env var needed)

For a feature missing from this list, search the codebase: every optional feature has a `process.env.SOMETHING` check guarding it. The full list is documented in `.env.example`.