

CatalogGuard manual

Every feature and behaviour of CatalogGuard, as built.

Version 2026-11-14

A GuardHQ app for Shopify

Support: support@catalogguardhq.com · https://catalogguardhq.com/support

Privacy: https://catalogguardhq.com/privacy · Terms: https://catalogguardhq.com/terms

Contents

1. What CatalogGuard does
2. Snapshots
3. A snapshot page
4. Timeline
5. Coverage
6. Restore
7. A run page
8. Runs
9. Settings
10. Plan
11. What a snapshot holds
12. Restores
13. Runs, reports and undo
14. Plans
15. What CatalogGuard never does
16. If something goes wrong
17. Support

CatalogGuard manual

CatalogGuardHQ, called CatalogGuard below.

Written 2026-10-18 from the app as it is built. Every sentence in quotation marks is the text the app shows on screen or prints in a report.

1. What CatalogGuard does

CatalogGuard takes a snapshot of your product catalogue every night at 2:00 in your store's time zone, and whenever you ask for one. A snapshot holds your products, variants, prices, inventory quantities, images by reference, product and variant metafields, and collections, exactly as the Coverage page lists them.

When something goes wrong, you choose a snapshot, the scope (the whole catalogue, one collection, or a list of products), the fields, and Safe or Force. CatalogGuard shows you a preview of every value that would change before it writes anything. You apply the restore from the preview. Before the first write it takes a snapshot of the current values, so every restore can be undone. Every write is read back from your store and checked. No product, variant or image of yours is deleted: a product, variant, stocked location or metafield that exists now and not in the snapshot is listed and left; the two exceptions Settings names (a product taken out of a hand-picked collection the snapshot did not list it in, the product itself staying, and a product the restore itself created seconds earlier and could not complete) are stated in the report.

CatalogGuard asks Shopify for these permissions only: read and write products, read and write inventory, read locations. It never reads customer or order data.

The app has six pages, in the order they appear in its navigation: Snapshots, Coverage, Restore, Runs, Settings, Plan. Opening the app takes you to Snapshots.

By plan

Every plan gets the same snapshots, the same restore, preview, verification and undo. Four limits differ:

- Free: a restore touches at most 100 products; snapshots kept 7 days; 1 on-demand snapshot a day; at most 20 deleted products or variants recreated in one restore.
- Starter ($9/month): 2,000 products; 30 days; 5 on-demand snapshots a day; 200 recreations.
- Growth ($19/month): 10,000 products; 90 days; unlimited on-demand snapshots; 1,000 recreations.
- Scale ($49/month): 100,000 products; 365 days; unlimited on-demand snapshots; 10,000 recreations.

2. Snapshots

The Snapshots page is where you see your snapshots and take one now.

Take a snapshot now

The section "Take a snapshot now" explains the schedule:

> "A snapshot of every product, variant, collection, metafield, image reference and inventory quantity is taken every night at 2:00 in <your time zone>. You can also take one now: your <Plan> plan allows <allowance> (<n> taken today). Snapshots are kept <days> days. What a snapshot captures and what a restore brings back: see Coverage."

The allowance reads "1 on-demand snapshot a day" on Free, "5 on-demand snapshots a day" on Starter, and "unlimited on-demand snapshots" on Growth and Scale; the "(n taken today)" part is shown only on plans with a daily allowance. On-demand snapshots are counted per calendar day in your store's time zone.

The button "Take a snapshot now" is disabled while a snapshot is running (the text " A snapshot is running." appears next to it) and when today's allowance is used up, in which case the reason appears next to it, for example on Free:

> "Your Free plan allows 1 on-demand snapshot a day; you have taken 1 today. Tonight's snapshot still runs. Starter allows 5, Growth and Scale unlimited. See the Plan page."

On Starter the last sentence reads "Growth and Scale unlimited. See the Plan page."

When you press the button, a banner "Snapshot started" appears with "It appears below when it finishes, usually within a few minutes. Refresh to see it." If a snapshot is already running, or a snapshot job for your store is already queued, a warning banner says "A snapshot is already running. It will appear in the list when it finishes, usually within a few minutes." An on-demand snapshot counts against the day only once its job has been queued; a refused request costs nothing.

The list

The section heading is "Snapshots (<count>, <total size>)", counting every snapshot the store keeps, with the size in KB or MB; the table below shows the 100 newest, and "Show older snapshots" under it shows the next 100, and so on back to the oldest snapshot your plan keeps. Under the heading, one

sentence says when the next nightly snapshot is due:

- "The first nightly snapshot is due now and starts within a few minutes." (no nightly snapshot yet; added 2026-10-22, CG-39)
- "The next nightly snapshot is taken today at 2:00 in <zone>." (before 2:00)
- "The next nightly snapshot is taken tomorrow at 2:00 in <zone>." (tonight's has been taken)
- "Tonight's nightly snapshot is due now and starts within a few minutes (2:00 in <zone> has passed)."

With no snapshots yet, the page says "No snapshots yet. The first one is taken within a few minutes of installing, then every night at 2:00 in your store's time zone; or take one now." and shows the section "What a restore does" (the same text as on the Settings page, quoted under Settings below). The first snapshot after installing is a nightly one: the worker takes it on its next check, within a minute or two, whatever the hour (changed 2026-10-22, CG-39; before, an install between midnight and 2:00 waited for 2:00).

Otherwise the page shows a table of the 100 newest snapshots with the columns "Taken", "Kind", "Status", "Products", "Variants", "Collections", "Size". When your store keeps more than 100, the table ends with "Show older snapshots", which shows the next 100, until the oldest one kept; the heading's count stays the same. Every kept snapshot has its own page, with "Download this snapshot" and "Restore from this snapshot" (added 2026-11-13, CC-CG-21; before, only the 100 newest could be opened from the list). Snapshots taken before a restore are never deleted by retention, so the link can appear on any plan once more than 100 are kept.

- "Taken" is the time, shown in your store's time zone. A stored snapshot's time is a link to its page.
- "Kind" is "nightly", "on demand", or "before a restore" (the snapshot of current values taken before a restore or an undo wrote anything). A "before a restore" snapshot that holds nothing reads "before a restore (recreate only)": its restore only recreated deleted products, so nothing existed to save (added 2026-11-14, CC-CG-23; before, such a row read "before a restore" with 0 / 0 / 0, as if something had gone wrong).
- "Status" is "stored", "in progress", "failed", or "failed: <reason>".
- "Products", "Variants" and "Collections" are the counts in that snapshot. For a "before a restore" snapshot they count only what that restore touched, so a restore that changed only a collection reads 0 products, 0 variants, 1 collection (the Collections column was added 2026-11-14, CC-CG-22; before, such a row read as if empty).
- "Size" is the size of the encrypted file.

If your store has more products than your plan covers, every page that knows your plan shows the warning banner "Your store has more products than your plan covers" with a sentence such as:

> "Your Free plan covers 100 products; your store has 617. Snapshots are still taken in full; a restore touches at most 100 until you upgrade. Starter covers 2,000, Growth 10,000, Scale 100,000."

followed by the link "See plans". On Scale the sentence ends "a restore touches at most 100,000. No larger plan is offered."

By plan

- Free: 1 on-demand snapshot a day; snapshots kept 7 days; the over-limit banner appears above 100 products.
- Starter: 5 on-demand snapshots a day; snapshots kept 30 days; the over-limit banner appears above 2,000 products.
- Growth: unlimited on-demand snapshots; snapshots kept 90 days; the over-limit banner appears above 10,000 products.
- Scale: unlimited on-demand snapshots; snapshots kept 365 days; the over-limit banner appears above 100,000 products.

3. A snapshot page

The line under the heading ends with "Download this snapshot": it saves one plain JSON file, `catalogguard-snapshot-{time}.json`, holding the snapshot exactly as CatalogGuard stores it (every product, variant, price, inventory quantity per location, image reference, metafield and collection the Coverage page lists), so you hold a copy outside the app. On every plan, for "before a restore" snapshots too. (Added 2026-10-21.)

Click a stored snapshot's time to open it. The heading is "Snapshot of <time>". Under it: "← Snapshots · <n> products, <m> collections · Restore from this snapshot". The link "Restore from this snapshot" opens the Restore page with this snapshot already selected. On a "before a restore" snapshot the link is replaced by "taken before a restore, for its undo; only what that restore touched is here": such a snapshot holds only the products and collections its restore touched, and is used by Undo rather than offered on the Restore page. When it is empty, it reads instead "taken before a restore that only recreated deleted products or variants; nothing existed to save, so this snapshot is empty, and that restore's undo keeps what it recreated".

The page has two sections.

Look up a product in this snapshot

> "See what this snapshot stored for one product. Enter its id (the number from the product's URL), its gid, its handle or its title. The match choice applies only to a title; an id, gid or handle is found as typed."

Type into the box and choose one of "For a title: match it exactly" or "For a title: contains this (any case)", then press "Look up".

- A number is taken as a product id. A value starting with "gid://" is taken as a gid. Anything else is matched against handles (any case) and titles.
- A title is matched exactly, or as a case-insensitive part when you chose "contains".
- At most 10 products are shown. If more match, the page says "<total> products match; the first 10 are shown. Narrow the search to see the others."
- If nothing matches: "This snapshot holds no product that matches "<what you typed>"."

Each product found shows its title, then "Handle <handle> · id <gid> · status <active, draft or archived> · vendor <vendor or none> · type <type or none>", then a table of its variants with the columns "SKU", "Variant", "Price", "Compare-at", "Inventory". "Inventory" is the total of the available quantity across all

locations in that snapshot. The line under the title ends with a "Timeline" link to the product's Timeline (section "Timeline").

The lookup reads the stored snapshot only, never your live store.

What changed since the previous snapshot

CatalogGuard compares this snapshot with the previous stored nightly or on-demand snapshot. (A "before a restore" snapshot is never the one compared against.)

- For the oldest snapshot kept: "This is the earliest snapshot kept, so there is nothing to compare it with."
- Otherwise: "Compared with the snapshot of <time>: <summary> Who changed a product, where Shopify keeps a record, is on the product's Timeline (the link beside each product below)." (Wording changed 2026-10-21 with the Timeline.)
- On a "before a restore" snapshot: "Compared with the snapshot of <time>, on the products and collections this snapshot holds: <summary> …". Only those products and collections are compared, so nothing else is listed as removed.

The summary is "Nothing changed between these snapshots." or a sentence made of these parts, separated by semicolons: "<n> changes across <m> products", "<x> products and <y> variants removed", "<x> products and <y> variants added", "<k> collections changed".

Below the summary, the products that changed are listed by handle, in the order they appear, each with its changes, one per line:

- a product field: "title "HAT" → "Hat"", "status draft → active", "tags "new, hats" → "hats"", "2 images → 3 images" (when the number of images changed); when the number is the same, one line per image that differs, "image 2 alt "Back" → "Back view"" (a line break in an alt reads as a space), "image 1 (<address>) → (<address>)" when a different image stands there, and "images reordered" once when only their order changed. The restore report's Changes lines read the same way (changed 2026-11-14, CC-CG-24; before, an alt change read "1 image (<address>) → 1 image (<address>)", the same on both sides);
- a variant field, prefixed by the variant's SKU: "hat-1: price 99.00 → 10.00", "hat-1: inventory at Warehouse 0 → 5", "hat-1: metafield custom.size (none) → "M"";
- "product removed", "product added", "<sku>: variant removed", "<sku>: variant added".

Collections that changed follow, each headed "Collection <handle>", with lines such as "title "summer" → "Summer"" or "products and their order 12 products → 11 products".

Only products that changed are listed, 200 a page. With more than one page the section shows "Products 1–200 of <total> that changed (page 1 of <pages>)." with "Previous" and "Next" links.

Values are printed the way reports print them: text in quotation marks and cut to 200 characters, money bare, yes/no for switches, a description as plain text without its HTML, an empty value as "(none)". Location ids are replaced by location names when CatalogGuard can read your locations.

By plan

The same on every plan. The lookup and the comparison work the same on every plan; what differs is only how long a snapshot is kept before it is deleted (7, 30, 90 or 365 days).

4. Timeline

The Timeline shows one product across the kept snapshots: what each snapshot holds for it, what changed since the snapshot before, and, where Shopify keeps a record, who changed it. It is reached from a snapshot page: each product found by "Look up a product in this snapshot" and each product listed under "What changed since the previous snapshot" carries a "Timeline" link. Its address is `/app/timeline/{product id}` (the number from the product's URL in your admin).

The heading is "Timeline of {title}", then "← Snapshots · handle {handle} · id {id}". When no kept snapshot in the range holds the product: "No kept snapshot in this range holds a product with id {id}."

"Snapshots {a}–{b} of {n}, oldest first; each row says what changed for this product since the snapshot before it." with "Newer" and "Older" links when there are more; 31 snapshots are shown at a time, the newest range first. Only nightly and on-demand snapshots are rows; a "before a restore" snapshot holds only what its restore touched, so it is never a row.

Each row is "Snapshot of {time}" (a link to that snapshot's page) followed by one of: "the earliest snapshot shown", "not in this snapshot", "added", "changed", "unchanged", "removed". A changed row lists the changes in the same words as "What changed since the previous snapshot" (section "A snapshot page"): "title "HAT" → "Hat"", "hat-1: price 99.00 → 10.00", and so on; "added" and "removed" rows carry "product added" / "product removed".

Where Shopify keeps a record of who changed the product, the record is shown under the first snapshot that holds the change: "Shopify's record: {Shopify's sentence} By {the app Shopify names | a person in the admin | unknown}, {time}." Records newer than every snapshot shown appear under the newest row as "Since this snapshot, Shopify's record: …". A changed or removed row with no record carries "No record from Shopify for this change (its record does not cover every kind of edit, variant prices among them), so who made it is unknown." Above the rows the page says either "Where Shopify keeps a record of who changed the product (an app, or a person in the admin), it is shown under the snapshot that first holds the change." or, when Shopify gave none, "Shopify gave no record of who changed this product, so the cause is unknown." The record is read from Shopify when the page opens and is never stored.

The Timeline reads snapshots and Shopify's record; it never writes anything. (Added 2026-10-21.)

By plan

The same on every plan; the plan's retention decides how many snapshots exist to show.

5. Coverage

The Coverage page is generated from the same table the engine uses to take snapshots and plan restores, so it cannot promise more than the engine does. Its introduction:

> "What a snapshot captures and what a restore can put back, field by field, so nothing is learned at restore time. Images are stored by reference, not as files. Snapshots are compared and restored exactly as listed here."

Three sections, "Products", "Variants" and "Collections", each a table with the columns "Field", "Captured", "Restored", "Note". "Captured" is "yes" on every row. The tables, as the page shows them:

Products

Field	Captured	Restored	Note
id	yes	no	Shopify assigns ids. A recreated product gets a new id; links that used the old one (order history, old URLs by id) do not follow.
handle	yes	yes	The URL handle. Restored as it was; Shopify refuses a handle another product now uses, which is reported.
title	yes	yes	Restored exactly.
description	yes	yes	The full HTML description, restored exactly.
vendor	yes	yes	Restored exactly.
product type	yes	yes	Restored exactly.
tags	yes	yes	Restored as a set; Shopify keeps tags in its own order.
status	yes	yes	Active, draft or archived. Restored exactly.
theme template	yes	yes	Which theme template the product page uses. Restored exactly; the template itself is part of the theme, which is outside this app.
options	yes	yes	The option names (Size, Colour, ...) in order with their values. A recreated product gets them back before its variants; renaming an option on an existing product is restored too. An option's values are the ones its variants use: a value no variant uses (Shopify keeps one) is not kept or compared, and never removed.
search listing	yes	yes	The page title and meta description, restored exactly.

Field	Captured	Restored	Note
images	yes	yes	By reference: the image's address, alt text and position, not the file. Images are compared by position and alt text, since Shopify re-hosts a re-added image under a new address and id; a changed file at the same position is not detected. A restore re-adds a missing image from its stored address; if Shopify has since removed the file that image cannot come back and is reported.
metafields	yes	yes	Every product metafield the app can read, with its type, restored exactly. A metafield added since the snapshot is listed and never removed. Metafields owned by another app that hides them are not visible to us and not captured.
variants	yes	yes	Every variant, with the fields below. A deleted variant is recreated with a new id.

Variants

Field	Captured	Restored	Note
id	yes	no	Shopify assigns ids. A recreated variant gets a new id; order lines that pointed at the old one do not follow.
SKU	yes	yes	Restored exactly.
barcode	yes	yes	Restored exactly.
variant title	yes	no	Shopify builds this from the option values; restore the options and the title follows.
option values	yes	yes	The variant's values for the product's options (size, colour, ...), restored exactly.
position	yes	yes	The variant's place in the product's list, restored by reordering.
image	yes	yes	Which of the product's images the variant shows. After a product is recreated its images have new ids, so the variant's image is matched by position.
price	yes	yes	Restored exactly, in the store's currency.
compare-at price	yes	yes	Restored exactly, including none.
weight	yes	yes	Restored with its unit.
weight unit	yes	yes	Restored with the weight.
requires shipping	yes	yes	Restored exactly.
when sold out	yes	yes	Whether selling continues when out of stock. Restored exactly.

Field	Captured	Restored	Note
charges tax	yes	yes	Restored exactly.
inventory item id	yes	no	Shopify assigns it; used to read and set quantities, never restored itself.
tracks quantity	yes	yes	Whether Shopify tracks this variant's stock. Restored exactly.
cost per item	yes	yes	Restored exactly, including none.
HS code	yes	yes	The customs tariff code, restored exactly.
country of origin	yes	yes	Restored exactly.
inventory	yes	yes	The available quantity at each location. Restored one location at a time, only if the quantity still equals what it was when the restore was planned; a sale in between fails that one quantity rather than overwriting it. A location stocked since the snapshot is listed and never set to nothing.
variant metafields	yes	yes	Every variant metafield the app can read, with its type, restored exactly. A metafield added since the snapshot is listed and never removed.

Collections

Field	Captured	Restored	Note
id	yes	no	Shopify assigns ids. Collections are not recreated by this app.
handle	yes	yes	Restored exactly.
title	yes	yes	Restored exactly.
description	yes	yes	The full HTML description, restored exactly.
image	yes	yes	By reference: the address and alt text, not the file. Compared by alt text; a re-added image is re-hosted by Shopify.
search listing	yes	yes	The page title and meta description, restored exactly.
theme template	yes	yes	Which theme template the collection page uses. Restored exactly.
sort order	yes	yes	Manual, best selling, alphabetical, price or date. Restored exactly.

Field	Captured	Restored	Note
products and their order	yes	yes	For a hand-picked collection, which products it holds and, when sorted manually, in what order. An automated collection's membership follows its rules and is not captured; a collection that changed between hand-picked and automated since the snapshot is left as it is, since Shopify cannot switch it back.

Not captured in this release

What	Why
Product category	Shopify's standard product category is not captured in this release; it is not changed by a restore.
Video and 3D media	Only images are captured, by reference. Videos and 3D models are left as they are.
Automated collection rules	The rules of an automated collection are not captured; its membership follows its rules and is not restored.
Publications and sales channels	Where a product or collection is published (online store, point of sale, other channels) is not captured in this release; it needs its own permission. A recreated product is published nowhere until you publish it.
B2B catalogs and price lists	Prices set per catalog or market are not captured; the base price is.
Selling plans	Subscription and pre-order plans are not captured.
Product reviews	Reviews belong to the reviews app that holds them and are not captured.
Tax codes	Per-variant tax codes are not captured in this release.
Bundle components	Which variants make up a bundle is not captured.
Unit price measurement	Reference units for unit pricing are not captured.
Customs details beyond the HS code and country of origin	Province of origin and per-country tariff codes are not captured in this release.
Collection metafields	Metafields on collections are not captured in this release; product and variant metafields are.
Gift card settings	Whether a product is a gift card, and its template, are not captured; a gift card product cannot be recreated by this app.

Outside this app

What	Why
Navigation menus	Outside this app's scope: the app does not request the navigation permissions, so menus are neither captured nor restored.

What	Why
Themes	Theme files are outside this app; Shopify keeps its own theme versions.
Orders	Orders are records of what happened and are never changed by this app; no order data is read.
Customers	No customer data is read or stored by this app.
Discounts	Discount codes and automatic discounts are outside this app.
Files	Files in the Shopify files area, including image files themselves, are outside this app; images are captured by reference.

By plan

The same on every plan.

6. Restore

The Restore page is a form in five numbered steps. Until you have a stored nightly or on-demand snapshot it says only "There is no snapshot to restore from yet. Take one on the Snapshots page, or wait for tonight's."

1. The snapshot to go back to

A list of your stored nightly and on-demand snapshots, newest first, up to 60, each shown as "<time> (nightly, <n> products)" or "<time> (on demand, <n> products)". The newest is selected unless you came from a snapshot page, in which case that snapshot is selected; if it is older than the 60 newest it is added to the list (at the end) so it can still be chosen. A "before a restore" snapshot is not offered here; it is used by Undo.

2. The scope

- "The whole catalogue" (selected by default).
- "A collection", with a box for the collection's gid ("gid://shopify/Collection/..."). Anything else (a bare number, a handle, another kind of gid) is refused before a run is made, with "Choose the collection by its gid (gid://shopify/Collection/...): "{value}" is not one." (Added 2026-10-27, REQ-CG-110; before, a bare number was queued and failed with Shopify's own "Invalid global id" text.)
- "A list of products", with a box and the note "One product per line: its id (the number from the product's URL), its gid, or its handle. Handles are looked up in the snapshot you chose, not in your store now." Lines may also be separated by commas. A number becomes a product id; a value starting with "gid://" is used as is; anything else is a handle, matched in any case. Repeats are dropped.
- "A list of SKUs", with a box and the note "One SKU per line or comma-separated. A SKU names one variant; a SKU ending in * names every variant whose SKU begins with those letters (TPAAC12 one variant, TPAAC* the brand, TPA* the family). Only the named variants' fields are restored, and each product's own fields once." An empty list is refused with "List at least one SKU, one per line."; a star anywhere but at the end, or with anything but letters, digits, "-", "_" and "." before it, with "A SKU pattern needs letters, digits, -, _ or . before its *: "{sku}"." (Added 2026-10-21.)

3. The fields

"What a snapshot captures and what a restore brings back: see Coverage."

The box "Every restorable field" is ticked by default. Under it are three groups you can open: "Product fields (all 12 included)", "Variant fields (all 18 included)", "Collection fields (all 8 included)". Untick "Every restorable field" and the headings change to "<n> of 12 selected" and so on as you tick single fields. The

fields carry the labels of the Coverage page's rows marked "Restored: yes". While "Every restorable field" is ticked, every field is restored whatever single boxes you ticked. There is no "variants" box (removed 2026-10-27, REQ-CG-123): a variant's fields are restored through the Variant fields boxes, and a deleted variant is recreated only with "Every restorable field". Collection fields ticked one by one with "A list of products" or "A list of SKUs" are refused with "Collection fields are not restored in a list of products or a list of SKUs; choose the whole catalogue or a collection, or untick them." (Added 2026-10-27, REQ-CG-124; before, they were silently ignored.)

4. Safe or force

- "Safe: a field you changed again after the latest nightly or on-demand snapshot is left as it is now and named." (selected by default)
- "Force: every difference goes back, including anything changed since."
- The box "Do not recreate deleted products and variants", followed by "A deleted product or variant comes back whole, so it is recreated only when "Every restorable field" is ticked above; with single fields ticked it is listed and left." Unticked by default: a product or variant the snapshot holds and the store no longer has is recreated when every restorable field is chosen.

5. Preview

> "Nothing is written yet. The preview lists every change, what would be recreated, and what would be left alone; you apply it from there. Every restore is itself snapshotted and can be undone."

Press "Preview the restore". CatalogGuard creates a run with the intent "preview", queues it, and takes you to its run page. A restore is always previewed first; there is no way to apply without a preview.

When the form is refused

A critical banner "The restore could not be set up" lists the problems:

- "Choose the snapshot to restore from."
- "Choose a scope: the whole catalogue, a collection, a list of products, or a list of SKUs."
- "List at least one product, one per line."
- "Choose the collection."
- "Choose at least one field to restore, or every field."
- ""<field>" cannot be restored; see the coverage page."
- "Choose safe or force."
- "The snapshot you chose holds no product with the handle "<handle>". Check the spelling, or use the product's id." (several handles are listed together, separated by commas)

- "A restore or snapshot for this store is already queued or in progress. Wait for it to finish, then try again."

A snapshot id that is not one of your store's stored snapshots is answered with "Not found".

By plan

The form is the same on every plan. What the plan changes is enforced when the run is planned:

- Free: a restore may touch at most 100 products and recreate at most 20 deleted products or variants.
- Starter: at most 2,000 products and 200 recreations in one restore.
- Growth: at most 10,000 products and 1,000 recreations in one restore.
- Scale: at most 100,000 products and 10,000 recreations in one restore.

7. A run page

A preview whose report says "Nothing to restore" offers no "Apply this restore" button (since 2026-10-21).

Every preview, apply and undo is a run with its own page. The heading is "Restore preview: <status>", "Restore: <status>" or "Undo: <status>", where the status is one of "queued", "running", "previewed", "stopped by a guardrail", "applied", "partial", "failed".

Under the heading: "← Runs · <time the run was created> · from the snapshot of <time> · safe mode · Coverage (what a restore brings back)". An undo shows no mode; "from the snapshot of" names the chosen snapshot for a restore and the pre-restore snapshot for an undo.

Banners

- While queued or running: "Running" with "This run is in progress; this page updates by itself every few seconds." The page reloads its data every four seconds until the run ends.
- "This run could not be completed" with "The report below says why. Nothing more was changed." when the run failed.
- "Stopped by a guardrail; these reasons cannot be overridden" with "Nothing was written. Narrow the scope, or the store, and preview again." when a hard reason stopped it.
- "Stopped by a guardrail" with "Nothing was written. Read the reasons below; apply anyway only if the change is what you want." when only soft reasons stopped it.
- "Not everything could be made" with "Some changes were made and some were not, or could not be verified. The report says which. Undo puts back what was made." when the status is partial.
- "This run was undone" with the link "See the undo." once an undo of this run has run.
- "This preview has been applied" (or "This preview is being applied" while the apply is queued or running) with the link "See the restore." on a previewed or held run once an apply pressed on it has gone through; the page then offers no Apply. An apply that was itself stopped by a guardrail or failed leaves the preview applicable.
- A critical banner with "This restore has a reason that cannot be overridden; change the scope and preview again." if you try to apply a run with a hard reason, "This preview has already been applied. Preview again to apply it another time." if an apply pressed on it already went through, or "A restore or snapshot for this store is already queued or in progress. Wait for it to finish, then try again." if the queue refused the job.

Report

The section "Report" shows the run's report as plain text, or "(no report yet)" while the run is queued or running. Every sentence a report can contain is listed under "Runs, reports and undo" below. A report stored by an older version of the app that begins "Held:" is shown beginning "Stopped by a guardrail:".

Actions

The section "Actions" appears when there is something to do.

- "Apply this restore", followed by " The store is read again before anything is written; if it moved since this preview, the restore holds and you preview again." The button is offered on a restore run whose status is previewed or stopped by a guardrail with no hard reason. Pressing it creates a new run with the same snapshot, scope, fields, mode and recreation choice and the intent "apply", and takes you to that run's page; the preview run stays in Runs as a record. The apply plans again and holds if its plan differs from this run's (see "Preview, then apply"), so what is written is what you saw. Once that apply has gone through, the preview's page says so and offers no second Apply; preview again to apply the same restore another time.
- The box "Override the guardrails for this restore" appears above the button when the run has soft reasons and no hard one: on a preview whose report lists them under "Guardrails flagged:", and on a run stopped by them alone. Tick it and apply to go ahead despite them; the apply then ends applied, with "Guardrails were overridden for this restore" in its report, rather than stopping first.
- "Undo this restore" (or "Undo this undo" on an undo run), followed by " Puts back what this run wrote where the store still holds it; what you changed since stays; recreated products and variants are kept." The button is offered on an applied or partial run that wrote at least one change (a run whose every change was refused or failed gets none; a change written but not verified counts as written), whose pre-restore snapshot is still stored (retention never deletes one), and that has not been undone already.

By plan

The page, its banners, Apply, Override and Undo are the same on every plan. The two hard reasons that name a plan are:

- Free: more than 100 products touched, or more than 20 recreations, in one restore.
- Starter: more than 2,000 products touched, or more than 200 recreations.
- Growth: more than 10,000 products touched, or more than 1,000 recreations.
- Scale: more than 100,000 products touched, or more than 10,000 recreations.

8. Runs

The Runs page lists your 50 most recent runs, newest first, in a table with the columns "When", "Kind", "Mode", "Status", "Changes". "When" is a link to the run page. "Kind" is "restore preview", "restore" or "undo". "Mode" is "safe" or "force", blank for an undo. "Status" is the run's status, with " (undone)" appended once the run has been undone. "Changes" is the number of changes the run planned (for a preview, or a run stopped before writing) or made (for an apply or undo: the changes Shopify accepted, so a partial run that made 1 of 2 shows 1).

With no runs: "No restores yet. Preview one from the Restore page to see it here."

By plan

The same on every plan.

9. Settings

The section "Your store" shows:

- "Time zone: <zone>. The nightly snapshot is taken at 2:00 in this zone, and the on-demand allowance counts per day in it."
- "Products: <count> at the last read."
- "Refresh reads your store's time zone and product count again from Shopify; it does not take a snapshot or change anything in your store." with the button "Refresh from Shopify". After pressing it: "Refreshed from your store".

CatalogGuard reads the time zone and product count from Shopify when it first sees your store, again when you open a page six hours or more after the last such read (however often you opened the app in between), again when you reinstall, and it updates the product count every time a snapshot is stored.

The section "E-mail" says: "When a restore or an undo you applied is stopped or lands only partly, or a nightly or on-demand snapshot could not be taken, CatalogGuard can e-mail your store's address with the reason and a link." It has one box, **E-mail me when a run stops or a snapshot fails (off until you tick it)**, unticked for every store until you tick it, and a **Save** button; saving shows "Saved". Ticked, the mail goes to the e-mail address of your store as Shopify holds it, from "CatalogGuard <no-reply@mail.getguardhq.com>", one per stopped restore or undo ("CatalogGuard stopped a restore" / "CatalogGuard stopped an undo", with the report's headline, counts and reasons, never the change lines, and the link to the run's page) and one per failed nightly or on-demand snapshot ("CatalogGuard could not take the nightly snapshot", with the time, the reason and a link to Snapshots). A preview, a run that applied in full, and a "before a restore" snapshot send nothing. Replies reach support@catalogguardhq.com. (Added 2026-10-21.)

The section "Accident alarm" says: "After each nightly or on-demand snapshot, CatalogGuard counts the products that changed, were added or were removed since the full snapshot before it. Over the line you set, it can e-mail your store's address once, with the summary and a link to what changed." It has the box **E-mail me when more products than this changed between two snapshots (off until you tick it)**, unticked for every store until you tick it, the number **Line, in products** (a whole number of at least 1; 50 until you change it) and a **Save** button; saving shows "Saved". Ticked, one mail per snapshot whose count passes the line: "CatalogGuard: {n} products changed since the previous snapshot", with the two snapshots' times, the comparison's summary, your line, and a link to that snapshot's page. A first snapshot, with no full snapshot before it, sends nothing. (Added 2026-10-21.) The mail's first sentence also names how many of those products had a variant's price or compare-at price move: "{n} products changed, were added or were removed; prices moved on {p} of them." With one it reads "a price moved on 1 of them.", with none "no prices moved." The line is still compared with every product counted, prices or not. (Added 2026-10-27, REQ-CG-19f.) For one product the sentence reads "1 product changed, was added or was removed". (Added 2026-11-01, CC-CG-12.)

The section "What a restore does":

> "A restore writes the chosen fields of the chosen products back to the values a snapshot holds. Every write is read back from your store; if the store does not show what was written, the run says so instead of claiming success. No product, variant or image of yours is deleted: a product, variant, stocked location or metafield that exists now and not in the snapshot is listed and left. Two exceptions, both stated in the report: restoring a hand-picked collection takes out of it a product the snapshot did not list there (the product itself stays), and a product or variant the restore created itself seconds earlier and could not complete is removed again. Before anything is written, a snapshot of the current values is taken, so every restore can be undone, however long ago it ran."
>
> "No customer or order data is ever read. Your snapshots are stored encrypted on our server and kept for your plan's retention period; the snapshot taken before a restore is kept as long as its run. Everything is deleted when Shopify sends the redaction notice, 48 hours after you uninstall; reinstalling before then keeps it all."
>
> (Both paragraphs reworded 2026-10-22: the first named no exception and the second said "uninstall notice", while the code deletes on the redaction notice.)

The section "Support": "Questions, or a restore that surprised you: support@catalogguardhq.com. Replies within one business day. Include the link to the run page."

By plan

The same on every plan.

10. Plan

The section "You are on <Plan>" says: "Covers <products> products (your store has <count>). Snapshots kept <days> days. On-demand snapshots: <1 a day, 5 a day or unlimited>. Up to <n> deleted products or variants recreated in one restore." Under it, the link "Change plan" opens Shopify's pricing page for the app in your admin, and the button "Refresh plan" reads your plan again from Shopify. CatalogGuard remembers your plan for an hour; when Shopify sends you back to this page after a plan change it reads the plan fresh at once.

The section "Your plan, and what the next plan adds" lists "Your <Plan> plan includes", for example on Free:

- Covers 100 products
- 7 days of snapshots
- 1 on-demand snapshot a day
- Recreates up to 20 deleted products in one restore

followed by "Starter, $9/month, adds" and only the limits that differ, each with the current value in brackets:

- Covers 2,000 products (from 100)
- 30 days of snapshots (from 7)
- 5 on-demand snapshots a day (from 1)
- Recreates up to 200 deleted products in one restore (from 20)

On Starter the on-demand line reads "Unlimited on-demand snapshots (from 5 a day)". On Growth the next panel omits on-demand snapshots, since Scale has the same. On Scale the panel reads "You are on the largest plan."

The section "All plans" shows this table; your plan's name carries " (current)":

Plan	Price	Products	Snapshots kept	On-demand snapshots
Free	Free	100	7 days	1 a day
Starter	$9/month	2,000	30 days	5 a day
Growth	$19/month	10,000	90 days	Unlimited
Scale	$49/month	100,000	365 days	Unlimited

Under the table: "Nightly snapshots, the coverage page, the timeline, restore with preview and verification, and undo are on every plan, including Free. A store with more products than its plan covers still gets full snapshots; a restore may touch at most the plan's count until you upgrade."

By plan

- Free: Free; covers 100 products; 7 days of snapshots; 1 on-demand snapshot a day; recreates up to 20 deleted products or variants in one restore.
- Starter: $9/month; covers 2,000 products; 30 days; 5 on-demand snapshots a day; recreates up to 200.
- Growth: $19/month; covers 10,000 products; 90 days; unlimited on-demand snapshots; recreates up to 1,000.
- Scale: $49/month; covers 100,000 products; 365 days; unlimited on-demand snapshots; recreates up to 10,000.

11. What a snapshot holds

When snapshots are taken

- Nightly: due from 2:00 in your store's time zone, once per local day. A worker checks every minute; the first check after 2:00 starts it. If a night is missed (the worker was down, or a snapshot was still running at 2:00), it is taken later the same day. The nightly snapshot never counts against the on-demand allowance. It is not taken while another snapshot of your store is running, and not after you uninstall.
- On demand: from the Snapshots page, within the plan's daily allowance.
- Before a restore or an undo: automatically, holding only the products and collections that run touches. It is listed as "before a restore".

One snapshot of a store runs at a time.

How a snapshot is taken

CatalogGuard runs four bulk reads against your store, one after another: products (with their images, product metafields, variants and the available quantity at each location), variant metafields and each variant's image, hand-picked collections with their products in order, and the list of all collections. It waits for each read with increasing pauses (2 seconds, 4, 8, up to a minute between checks, at most 60 checks). The files are read line by line and assembled into the catalogue. The snapshot is refused if it carries any field the Coverage table does not list, so the table and the data cannot drift apart. The file is then compressed and encrypted (AES-256-GCM) and stored on CatalogGuard's server; the counts and the size are recorded; and your store's product count is updated.

If a read fails, nothing is stored and the snapshot is marked failed with one of these reasons:

- "Shopify refused to start the <products / variant metafields / collections / collection index> query: <Shopify's message>."
- "Shopify could not finish the <query> query (<error code>). Nothing was stored; the next snapshot will try again."
- "Shopify had not finished the <query> query after 60 checks. Nothing was stored; the next snapshot will try again."
- "Shopify's result file could not be downloaded (<status>)."
- "The worker restarted before this snapshot finished. Nothing was stored; the next snapshot will try again."
- "A snapshot job for this store was already queued."

A snapshot still "in progress" after three hours is given up and marked failed; its reads at Shopify are cancelled.

What is captured

Everything in the Coverage page's three tables, by product, variant and collection; nothing else. In particular:

- Images are captured by reference: the address, alt text and position. The file itself stays with Shopify.
- Inventory is the available quantity at each location.
- Metafields are every product and variant metafield the app can read, with their type. Metafields another app keeps private are not visible to CatalogGuard.
- Only hand-picked collections have their products captured; for an automated collection the collection's own fields are captured and its membership is not.
- Everything in "Not captured in this release" and "Outside this app" is not in a snapshot.

Retention

Snapshots are kept for your plan's retention period: 7 days on Free, 30 on Starter, 90 on Growth, 365 on Scale. After every snapshot job, whether it succeeded or failed, CatalogGuard deletes stored snapshots older than that, except the most recent stored nightly or on-demand snapshot, which is always kept so you always have something to restore from. A failed snapshot is removed after a day, as is one stuck in progress for a day. "Before a restore" snapshots are not deleted by retention: each is kept as long as its run, so a restore can be undone whatever its age.

By plan

What a snapshot captures is the same on every plan, and a snapshot is always taken in full. What differs:

- Free: kept 7 days; 1 on-demand snapshot a day.
- Starter: kept 30 days; 5 on-demand snapshots a day.
- Growth: kept 90 days; unlimited on-demand snapshots.
- Scale: kept 365 days; unlimited on-demand snapshots.

12. Restores

Scopes

- The whole catalogue: every product and every collection. For the current state CatalogGuard takes a fresh full read of your store (the same four bulk reads as a snapshot, not stored). This read is refused while a snapshot of your store is running: the run fails with "A snapshot is running; try again when it finishes."
- A collection: the collection itself and, for a hand-picked collection, the products it holds now or held in the snapshot. A product that merely left the collection is read and seen as present, so it is re-added to the collection rather than recreated. For an automated collection only the collection's own fields are in scope.
- A list of products: those products. An id found in neither the snapshot nor the store is dropped.
- A list of SKUs: every product that holds a named variant, in the snapshot or in the store now (so a product re-added under a new id is found by its SKU and restored, not duplicated). Of such a product, the named variants' fields and the product's own fields are restored; a variant not named is not compared, restored, recreated or listed, even when it differs. A deleted product with a named variant is recreated whole. Collections are never in a SKU scope. (Added 2026-10-21.)

For a collection or a list of products, a product an earlier restore recreated is also read under the new id that run recorded, so it is recognised and not recreated again (see "Recreating deleted products and variants"). A list of products may name such a product by either id: the one the snapshot holds or the one it has now. Only the whole-catalogue scope can recognise a product you re-added by hand under a new id, since it reads the whole store.

If the scope matches nothing on either side, the restore is stopped with the hard reason "Nothing in that snapshot or in the store matches the chosen scope."

Fields

Only the fields you chose are compared and restored, and only fields the Coverage page marks restorable can be chosen. Values are compared the way Shopify means them: "18", "18.0" and "18.00" are the same price; tags are a set; images are matched by position and alt text (a collection's single image by its alt text alone); the image a variant shows is matched by that image's position, so a product whose images were re-added or recreated (new ids, same places) needs no change.

Safe mode

Safe mode compares three states: the snapshot you chose, the latest stored nightly or on-demand snapshot, and your store now. A field is put back only if your store still holds the value the latest nightly or on-demand snapshot saw. A product recreated since the snapshot you chose is looked up in the latest nightly or on-demand snapshot by its new id, so its fields are compared like any other product's. If you changed it again after the latest nightly or on-demand snapshot, it is left as it is and named in the report:

> "- <handle or SKU>: <field> is <value now>, changed since the latest snapshot; the snapshot you chose has <snapshot value>. Choose force to set it anyway."

Likewise a product or variant that was deleted after the latest nightly or on-demand snapshot is not recreated:

> "- <handle or SKU>: product was deleted after the latest nightly or on-demand snapshot; not recreated. Choose force to recreate it."

The report says what it compared against: "Compared against the latest snapshot (<time>): anything changed since it is left as it is now." If the snapshot you chose is itself the latest, there is nothing to tell a later edit from the change being undone, and the report says "No later snapshot to compare against, so every difference goes back."

Force mode

Every difference between the snapshot and your store goes back, in the chosen scope and fields. The report says "Force: every difference goes back, including anything changed since the snapshot."

What exists now and not in the snapshot

A product, variant, stocked location or metafield that your store has now and the snapshot does not is never removed. It is listed under "Exists now, not in that snapshot (left alone):" as

- "- <handle>: product"
- "- <sku>: variant"
- "- <sku>: inventory at <location> (stocked since the snapshot)"
- "- <handle or sku>: metafield <namespace.key> (added since the snapshot)"

Recreating deleted products and variants

A recreation is several writes in a row (the product, its variants, their stock at each location, their metafields). When a later one fails, the product Shopify had already created is removed again and the report ends with "The product Shopify had created for it was removed again, so nothing of it remains." (for a variant under an existing product: "The variant Shopify had created for it was removed again …"). If that removal fails too, the report says what to delete by hand: "A product was created but could not be completed or removed: delete "{handle}" (id {id}) in your admin before trying again." Stock lands exactly as the snapshot had it per location, including the location Shopify activates a new variant at on its own. For a list of products or a collection, a product that exists now under a new id with the snapshot's handle (re-added by hand, or left by an earlier failed restore) is recognised and restored against the new id, not recreated again. In a list of products or a list of SKUs, a recreated product is also put back into every hand-picked collection the snapshot held it in, and the report ends with "Recreated products and their hand-picked collections:" naming each (in a whole-catalogue or collection restore the collections are in scope and restored as before). (Added 2026-10-21.)

A product or variant the snapshot holds and your store no longer has is recreated, unless you ticked "Do not recreate deleted products and variants" (then it is listed as "- <handle or SKU>: product is missing from the store; recreation was switched off for this restore."), you chose single fields rather than "Every restorable field" (a deleted product or variant comes back whole or not at all, so it is listed as "- <handle or SKU>: product is missing from the store; deleted products and variants are recreated only when every restorable field is chosen."), or Safe mode saw it deleted after the latest nightly or on-demand snapshot.

- A recreated product gets a new id from Shopify; so do its variants. The preview says "- <handle>: product, with a new id"; the applied report says "- <handle>: product, now <new gid> (was <old gid>)".
- It is created with its handle, title, description, vendor, type, tags, status, theme template, search listing, options, product metafields and images (re-added from their stored addresses), then its variants in order with their fields, their image matched by position, their inventory activated at each location at the stored quantity, and their variant metafields.
- It is published nowhere until you publish it; publications are not captured.
- A collection scope or whole-catalogue restore that puts the product back into a hand-picked collection names it by its new id; the report then says "<n> changes referred to recreated products and were updated to their new ids."
- Collections are never recreated: "- <handle>: collection no longer exists; collections are not recreated."
- Your plan limits how many deleted products and variants one restore may recreate: 20 on Free, 200 on Starter, 1,000 on Growth, 10,000 on Scale. Above that the restore is stopped with a hard reason, for example "Your Free plan recreates up to 20 deleted products or variants in one restore; this restore would recreate 25. Starter recreates 200, Growth 1,000, Scale 10,000. See the Plan page."

Recreation is not repeated. If the store already has the product again under another id (you re-added it by hand, or an earlier restore recreated it), CatalogGuard recognises it by the id an earlier run recorded or by the same handle; a variant by the recorded id, the same SKU, or the same option values. Its fields are restored against the new id instead, a hand-picked collection's membership names it by the new id, and the report lists it under "Already recreated, so not recreated again; fields restored against the new id:" as "- <handle or SKU>: product, now <new gid> (was <old gid>)". Previewing and applying the same restore twice therefore recreates each product once.

A deleted product that the restore does not recreate (recreation switched off, or Safe mode saw it deleted after the latest nightly or on-demand snapshot) is left out of any collection membership the restore writes; the rest of the collection is restored, and the product is named under "Left as it is now".

Guardrails

A restore is held before anything is written when:

- Hard reasons, which cannot be overridden:
  - "Nothing in that snapshot or in the store matches the chosen scope."
  - the recreation limit sentence quoted above;
  - your plan's product limit: "Your Free plan covers 100 products in one restore; this restore would touch 120, so it is stopped. Narrow the scope or upgrade. Starter covers 2,000, Growth 10,000, Scale 100,000. See the Plan page." (On Scale the sentence ends "Narrow the scope. No larger plan is offered. See the Plan page.")
  - the store moved between the preview and the apply, so the plan differs from the preview or a live value differs from the plan (see below).
- A soft reason, which you may override from the run page: "<changed> of <store> products in the store (<percent>%) would change in one restore. The limit is 50%." This rule applies once your store has at least 10 products.

A product counts as touched, for the plan's product limit and the 50% rule alike, when the restore sets one of its fields or a variant's, recreates it or one of its variants, or adds it to, removes it from or moves it within a hand-picked collection. The headline "<n> changes would be made across <m> products" counts the same way, and names the collections a restore changes as well (below).

A held run's report opens with "Stopped by a guardrail: the reasons below cannot be overridden." or "Stopped by a guardrail: read the reasons below. The restore can be applied with an override." and lists the reasons, each once, under one "Guardrails flagged:" heading, the plan's limits included. When you override, the applied report says "Guardrails were overridden for this restore. They had flagged:" followed by the reasons.

Preview, then apply

A preview plans the restore and stops. It writes nothing and takes no snapshot. A preview that a hard reason would stop (an empty scope, a plan limit) ends "stopped by a guardrail" with the same report an apply would get; a preview with soft reasons only ends "previewed", lists them under "Guardrails flagged:" and offers the override box.

An apply is a new run. It reads your store again and plans again, and compares that plan with the preview you pressed Apply on. If the store moved in between so that a change is now needed that the preview did not show, a change would now set other values, or a change the preview showed is no longer needed, the apply is held with the hard reason:

> "The restore the store needs now differs from the preview you pressed Apply on (<handle>: <field> <value now> → <snapshot value> was not in the preview; <handle>: <field> <value now> → <snapshot value> (the preview showed <value> → <snapshot value>); <handle>: <field> … is no longer needed; and <k> more). Preview again."

Inventory is left out of that comparison (a sale in between is caught at write time, quantity by quantity), so nothing is ever written that a preview did not show. Then it reads the live values of everything the plan touches. If any value differs from what the plan expected (inventory excepted; it has its own check at write time), the restore is held with the hard reason:

> "<n> products and <m> collections changed in the store since this restore was planned (<handle>: <field> <planned value> is now <value>; …; and <k> more). Preview again."

A variant that would be recreated under a product that is gone, or that exists again with the same SKU or option values, is reported the same way: "<handle>: the product that <sku> would be recreated under is gone" or "<sku>: a variant with this SKU exists again under <handle> (<gid>)".

If nothing moved, that live state is stored as the pre-restore snapshot and its row is saved before the first write, so even an interrupted run can be undone. Then the writes go out: creates first, then sets, in batches of 50. If the plan has no changes at all, the run ends as applied with "Nothing to restore: the chosen fields already match that snapshot." and no snapshot is taken.

How writes are made

- Product fields are set on the product. Option names are restored by position. Images are matched by position: an image at a wanted position has its alt text corrected in place, a position with no image is re-added from the stored address, images beyond the wanted ones stay, and the order is put back. Nothing is deleted.

- Variant fields are set together per product; positions are put back by reordering. Shopify takes a variant's weight and its unit as one measurement, so restoring only the weight keeps the unit the variant has now, and restoring only the unit keeps the weight.
- Inventory is set per location by compare-and-set: the quantity is written only if it still equals what it was when the restore was planned. A quantity that moved (a sale in between) is refused for that one location and reported as "could not be set: stock at <location> moved since the snapshot (now <x>, was <y> when this restore was planned)"; the rest of the restore carries on. A location the variant is not stocked at is activated at the wanted quantity.
- Metafields are set with their type.
- A collection's fields are set; its membership is brought back to the snapshot's list by adding the missing products and removing the extra ones from the collection (the products themselves are untouched), and, when the collection is sorted manually, its order is put back. If Shopify is still applying a collection change after a short wait, the batch fails with "Shopify is still applying a collection change; try again in a minute."
- Any refusal by Shopify (for example a handle another product now uses) stops that batch. The report names the handle or SKU and Shopify's message. Nothing is retried.

Verify

After the writes, CatalogGuard reads back every change it made and compares the store with what it wrote. A set the store does not show is counted as not made and listed under "Not verified, so not counted as made:". A recreated product or variant that cannot be read at all is counted the same way. One that exists but differs in a field stays counted as recreated and is listed under "Recreated with differences:" as "- <handle>: product recreated, but the store shows <field> <value> where <value> was written", and the headline says "<k> of them read back with differences (below)" rather than "applied and verified". If the read-back itself fails, the report says "The read-back failed (<reason>), so no change could be verified; every change written is counted as not made until you check the store. This restore can still be undone."

A run whose every change was made and verified ends "applied"; any other outcome ends "partial".

By plan

Scopes, fields, Safe and Force, guardrails and verification work the same on every plan. The two limits a plan sets:

- Free: a restore may touch at most 100 products and recreate at most 20 deleted products or variants.
- Starter: at most 2,000 products and 200 recreations.
- Growth: at most 10,000 products and 1,000 recreations.
- Scale: at most 100,000 products and 10,000 recreations.

13. Runs, reports and undo

Run statuses

"queued" (waiting for the worker), "running", "previewed" (a preview finished), "stopped by a guardrail", "applied", "partial" (some changes made, some not, or not verified), "failed" (the run could not start or could not continue; the report says why).

Jobs expire after three hours. A run still queued or running past that, or whose job ended without the run being closed, is marked failed with the report "The run could not be completed. The worker restarted before this run finished.\nWhat was written before that is in the store; a pre-restore snapshot, if one was taken, is listed under Snapshots. Preview again." An apply keeps the list of what it has written on the run after every batch, so a run cut off after writing is marked "partial" instead, with "The run stopped part-way. The worker restarted before this run finished. \n{n} changes were written before that; this run can be undone. The rest was not written: preview again.", and its Undo puts those changes back. (Added 2026-10-25; before, such a run was marked failed and could not be undone.)

Every sentence a restore report can print

Headline:

- "Stopped by a guardrail: the reasons below cannot be overridden."
- "Stopped by a guardrail: read the reasons below. The restore can be applied with an override."
- "Nothing to restore: <n> differences were left alone (below)." (or "1 difference was left alone (below).")
- "Nothing to restore: the chosen fields already match that snapshot."
- "<n> changes would be made across <m> products." optionally followed by "<x> deleted products and <y> deleted variants would be recreated, with new ids." (singular: "1 deleted variant would be recreated, with a new id.")
- When collections change too: "<n> changes would be made across <m> products and <k> collections."; when only collections change: "1 change would be made to 1 collection." or "<n> changes would be made to <k> collections." A restore that changes something never says "across 0 products" (changed 2026-11-13, gap 757; before, a change to a collection alone read "1 change would be made across 0 products.").
- "<n> changes applied and verified across <m> products." optionally followed by "<x> deleted products and <y> deleted variants recreated, with new ids." With collections: "applied and verified across <m> products and <k> collections"; collections alone: "applied and verified on <k> collections"

- "<n> changes applied across <m> products. <x> deleted products recreated, with new ids. <k> of them read back with differences (below); the rest verified." (singular: "1 of them reads back with differences") when a recreated product or variant exists but differs in a field.
- "The restore stopped part-way: <a> applied, <b> failed, <c> not attempted." then "Reason: <Shopify's message>"
- "<b> of <n> changes could not be made:"
- "The read-back failed (<reason>), so no change could be verified; every change written is counted as not made until you check the store. This restore can still be undone."
- "- <label>: <field> could not be set: <reason>" and "- <label>: product could not be recreated: <reason>"
- "- <label>: <field> written <value>, store shows <value>" (or "store shows nothing readable")
- "- <label>: product recreated, but the store shows nothing readable"

Guardrails:

- "Guardrails flagged:" or "Guardrails were overridden for this restore. They had flagged:" then "- <reason>" per reason.

Mode:

- "Force: every difference goes back, including anything changed since the snapshot."
- "Compared against the latest snapshot (<time>): anything changed since it is left as it is now."
- "No later snapshot to compare against, so every difference goes back."

Lists, each headed and then one line per item:

- "Changes:" (preview) and "Changes made:" (apply): "- <handle or SKU>: <field> <from> → <to>", for example "- hat: title "HAT" → "Hat"", "- hat-1: price 99.00 → 10.00", "- hat-1: inventory at Warehouse 0 → 5", "- hat: metafield custom.size (none) → "M"", "- summer: collection title "summer" → "Summer"". A variant with no SKU is named "variant <gid>".
- "Would be recreated:": "- <handle or SKU>: product, with a new id" / "variant, with a new id".
- "Recreated:": "- <label>: product, now <new gid> (was <old gid>)" or "- <label>: product, recreated".
- "Recreated with differences:" (see Verify).
- "<n> changes referred to recreated products and were updated to their new ids."
- "Not verified, so not counted as made:"
- "Failed:"
- "Not attempted:"

- "Left as it is now:" with the Safe-mode, recreation-off, fields-chosen, collection-gone and collection-kind lines quoted under Restores, the last being "- <handle>: this collection changed between hand-picked and automated since the snapshot; Shopify cannot switch it back, so its products are left as they are."
- "Already recreated, so not recreated again; fields restored against the new id:"
- "Exists now, not in that snapshot (left alone):"
- "A snapshot of the current values was taken at <time> before anything was written; this restore can be undone."

Times inside a report are shown in your store's time zone, as on the pages, with the zone named (for example 10/17/2026, 10:00:00 PM (America/New_York)).

A run that could not start or continue has a two-line report:

- "The restore could not start. The snapshot to restore from is not available.\nNothing was changed."
- "The restore could not start. A snapshot is running; try again when it finishes.\nNothing was changed."
- "The restore could not start. Shopify could not be read: <reason>\nNothing was changed."
- "The restore could not be completed. <reason>\nNothing more was changed."

Undo

Undo is offered on the run page of an applied or partial restore (or undo) that wrote at least one change and whose pre-restore snapshot still exists. It is a run of its own, listed in Runs as "undo".

An undo reads the live value of every change the run wrote, including a change the read-back could not verify. Where your store still holds exactly the value the run wrote, the value from the pre-restore snapshot is put back. Where you changed it since, or it cannot be read, it is left and named. What the run recreated is kept, and so is a location the run stocked or a metafield it set where there was none before: an undo never deletes anything, and the report names each such value so you can remove it by hand if you want it gone. Before writing, the undo takes its own snapshot of the values it is about to change, so an undo can itself be undone from its run page with "Undo this undo". An undo with nothing to put back (you changed every value since, or its restore only recreated) takes no snapshot and offers no "Undo this undo": there is nothing to undo again. Every value put back is read back and verified like a restore.

The undo report:

- "<n> values put back and verified."
- "The undo stopped part-way: <a> put back, <b> failed, <c> not attempted, <d> not verified." then "Reason: <message>"
- "<b> of <n> values could not be put back:"

- "The read-back failed (<reason>), so no value could be verified; every value written is counted as not put back until you check the store."
- "- <label>: <field> could not be set: <reason>"
- "Left as you set it:" then "- <label>: <field> is <value>, which you changed after the restore (the restore set <value>)." or "- <label>: <field> could not be read (the restore set <value>)."
- "Recreated by the restore and kept, since an undo never deletes:" then "- <label>: product" or "variant".
- "Added by the restore and kept, since an undo never deletes:" then "- <label>: inventory at <location> <quantity>; there was none before the restore. Remove it by hand if you want it gone." or "- <label>: metafield <namespace.key> "<value>"; there was none before the restore. Remove it by hand if you want it gone."
- "Put back:" then one "- <label>: <field> <from> → <to>" line per value.

An undo that cannot start says, after "The undo could not start.": "The run to undo was not found.", "That restore has already been undone.", "That run wrote nothing, so there is nothing to undo.", or "The snapshot taken before that run is no longer kept, so it cannot be undone." The run page offers no Undo in any of these cases: a run that wrote nothing (every change refused or failed) and a run whose pre-restore snapshot is gone (only a run from before 2026-10-22, when retention still deleted them) show no button. A run can be undone once; afterwards its page shows "This run was undone" and the Runs list marks it "(undone)".

An undo puts a value back to what the pre-restore snapshot held. If the restore set a metafield, or stocked a location, that did not exist before, the pre-restore snapshot holds nothing for it; the undo does not remove it (nothing is ever deleted) but lists it under "Added by the restore and kept", and you remove it by hand if that is what you want.

One run at a time

Your store has one restore job and one undo job queued or active at a time. A second request is refused with "A restore or snapshot for this store is already queued or in progress. Wait for it to finish, then try again." A restore of the whole catalogue cannot run while a snapshot is running, and a snapshot cannot start while a whole-catalogue restore is reading the store.

By plan

The same on every plan. Preview, apply, verify, the run log and undo are on every plan, including Free. An undo has no time limit on any plan: the snapshot taken before a run is kept as long as the run.

14. Plans

The four plans, their prices and limits:

Plan	Price	Products a restore may touch	Snapshots kept	On-demand snapshots a day	Deleted products or variants recreated in one restore
Free	Free	100	7 days	1	20
Starter	$9/month	2,000	30 days	5	200
Growth	$19/month	10,000	90 days	Unlimited	1,000
Scale	$49/month	100,000	365 days	Unlimited	10,000

- The product limit applies to a restore, not to a snapshot. Snapshots are always taken in full. A restore that would touch more products than the plan covers is stopped with a hard reason; narrow the scope or change the plan.
- Nightly snapshots, the Coverage page, the "What changed" comparison, the Timeline, restore with preview and verification, and undo are on every plan, including Free.
- Your plan is read from your Shopify subscription and remembered for an hour. "Refresh plan" on the Plan page reads it again at once. A subscription whose name CatalogGuard does not recognise is served as Free.
- Plans are changed on Shopify's pricing page for the app, reached by "Change plan" on the Plan page.

15. What CatalogGuard never does

- It never deletes a product, a variant, a collection, an image, a metafield or a stocked location in a restore. What exists now and not in the snapshot is listed and left. The one deletion a restore makes is of its own work: when it has just created a product or a variant and cannot complete it, it removes that creation again and the report says so (added 2026-10-21). The one thing a restore takes out is a product's membership of a hand-picked collection, when the snapshot did not list it there; the product itself stays.
- An undo never deletes either: a product or variant the restore recreated, a location it stocked and a metafield it set where there was none are kept and named in the undo's report.
- It never writes without a preview you have seen and an apply you have pressed. A preview writes nothing.
- It never overwrites an inventory quantity that moved since the restore was planned; that one quantity is refused and reported.
- It never retries a refused write.
- It never claims success without reading the value back from your store.
- It never reads customer or order data, and never asks for the permission to. It does not touch themes, navigation, discounts, files, publications or sales channels.
- It never keeps your data after you leave: after uninstall nothing new is taken, and when Shopify sends its redaction notice (48 hours after uninstall) every snapshot, run, session and file of your store is deleted. Mail you sent to support stays in our business mailbox, as the privacy page says. If you reinstall before that notice arrives, nothing is deleted and your snapshots and runs are still there. Requests about customer data are acknowledged; there is no customer data to look up.

By plan

The same on every plan.

16. If something goes wrong

You got an e-mail "CatalogGuard: {n} products changed since the previous snapshot". You ticked the accident alarm in Settings and more products than your line changed, were added or were removed between two snapshots. CatalogGuard changed nothing; open the link to see what changed, product by product, and each product's Timeline for who changed it where Shopify keeps a record. Raise the line or untick the box in Settings if the mails are noise.

You got an e-mail "CatalogGuard stopped a restore" (or "an undo", or "could not take the nightly snapshot"). You ticked "E-mail me when a run stops or a snapshot fails" in Settings. The mail carries the report's headline and reasons, or the snapshot's reason; its link opens the run's page or Snapshots, where the rest of this section applies. Untick the box in Settings if you do not want these.

"Left as it is now" in a report. In safe mode, a field you changed again after the latest nightly or on-demand snapshot is kept, and the report names it with the value you set, the value the snapshot holds, and the value in between. If you want the snapshot's value anyway, run the restore again in force mode, or set it by hand.

"Exists now, not in that snapshot" in a report. A product, variant, stocked location or metafield that was added after the snapshot is listed and left alone. CatalogGuard deletes no product, variant or image of yours; remove it by hand if that is what you want.

"Recreated, with a new id". A deleted product or variant is recreated from the snapshot. Shopify assigns it a new id, so links that used the old id (order history, old URLs by id) do not follow, and it is published nowhere until you publish it. The Coverage page lists what comes back and what does not.

Something you expected in a snapshot is not there. The Coverage page lists, field by field, what a snapshot captures and what a restore can put back. A few things cannot come back. A deleted product or variant is recreated with a new id, and links that used the old id do not follow. An image is stored by its address and alt text, not as a file, so if Shopify has removed the file, that image cannot come back and is reported. Metafields owned by another app that hides them are not visible and not captured. Not captured in this release: product category, video and 3D media, collection metafields, and publications and sales channels. Snapshots are kept for your plan's retention period and then deleted: 7 days on Free, 30 on Starter, 90 on Growth, 365 on Scale.

The restore was stopped by a guardrail. Read the reasons on the run page. A hard reason (an empty scope, more recreations or more products than your plan allows, or a store that moved since the preview) needs a narrower scope or a new preview. A soft reason (more than half the store changing at once) can be overridden on the run page with the box "Override the guardrails for this restore".

The run says partial. Some changes were made and some were not, or could not be verified. The report lists each under "Failed:", "Not attempted:" or "Not verified, so not counted as made:", with the reason where Shopify gave one. You can fix the rest by hand, preview the same restore again (what is already right is not changed again), or undo what was made.

The run says failed. The report's first lines say why: the snapshot was not available, a snapshot was running, Shopify could not be read, or the worker restarted. Nothing more was changed. Preview again.

A snapshot failed. Its status in the list says why. Nothing was stored; the next snapshot will try again. Tonight's nightly snapshot runs whatever happened to an on-demand one.

The Plan page shows the wrong plan or product count. Press "Refresh plan" on the Plan page, or "Refresh from Shopify" on the Settings page.

You need help. Write to support@catalogguardhq.com with the link to the run page (it looks like https://app.catalogguardhq.com/app/runs/<runId>), or the date of the snapshot and the name of the product you are asking about, and say briefly what you expected to happen and what happened instead. Replies within one business day.

By plan

The same on every plan.

17. Support

- Email: support@catalogguardhq.com (replies within one business day; include the link to the run page)
- Support page: https://catalogguardhq.com/support
- Privacy: https://catalogguardhq.com/privacy
- Terms: https://catalogguardhq.com/terms
- The app: https://app.catalogguardhq.com
