

Operator Guide

Twenty-four procedures for the people who run the connector day to day — sales, inventory and accounting staff. Each one says who does it, when, what it depends on, the exact screen, the steps, and how you know it worked.

MODULE baseup_shopify	VERSION 19.0.2.0.0	ODOO 19.0
PROCEDURES 24 in 6 phases	AUDIENCE Operators	ACCESS NEEDED Sales Manager

Contents

PHASE A · DAILY CHECKS

- 01** Confirm the store is connected
- 02** Read the log after an overnight sync
- 03** Trace a delivery to its Shopify order

PHASE B · PRODUCTS ONTO SHOPIFY

- 04** Push one product to Shopify
- 05** Push a batch of products
- 06** Pull one product back from Shopify
- 07** Two-way sync the product list
- 08** Queue a full catalogue resync

PHASE C · STOCK LEVELS

- 09** Decide which system owns stock
- 10** Pull stock levels from Shopify
- 11** Push stock levels to Shopify
- 12** Add a warehouse and map its Shopify location

PHASE D · ORDERS COMING IN

- 13** Pull orders on demand
- 14** Re-pull orders you think were missed
- 15** Release an order stuck in Draft
- 16** Check a draft order completed in Shopify

PHASE E · SHIPPING & MONEY

- 17** Fulfil a Shopify order
- 18** Add or correct a tracking number

- 19 Post a refund credit note
- 20 Confirm a cancellation reached Odoo

PHASE F · HOUSEKEEPING

- 21 Send an Odoo quotation to Shopify
- 22 Clean up errored products
- 23 Re-register webhooks after a URL change
- 24 Walk a new user through the tours

Phase A

Daily checks

Three habits that catch almost every problem early. None takes more than a minute.

01 Confirm the store is connected

WHO	WHEN	NEEDS
Sales Manager	Each morning	A connected account

Shopify > Configuration > Shopify Accounts

1. Open the account list. The **Status** badge should read **Connected** in green.
2. Check the **Last Sync** column. On the default schedule it should never be more than about 30 minutes old.
3. Open the account. The four timestamps under **Last Sync Timestamps** — overall, product, order and inventory — show which parts are actually running.
4. If the status is **Error**, click **Test Connection**. A failure here means the credentials need attention; see procedure 23 or escalate.

YOU'LL KNOW IT WORKED

Status is **Connected**, and the **Products** and **Orders** counters at the top of the form show the numbers you expect for that store.

02 Read the log after an overnight sync

WHO	WHEN	NEEDS
Sales Manager	Each morning	Procedure 01

Shopify > Operations > Logs

1. The list is newest first. Scan the **Type** column for **Error** and **Warning** entries.
2. Group or filter by **Operation** to see which activity failed — for example `cron_sync_all`, `import_order`, `import_refund` or `webhook`.
3. Open an entry to read the full message, plus the Odoo record and Shopify ID it relates to.
4. Act on the two you will actually see in normal running: an order **imported but left in Draft** (procedure 15) and a **refund with no posted invoice** (procedure 19).

YOU'LL KNOW IT WORKED

Only **Info** and **Success** entries since the last run, or every error has a matching action taken.

WATCH FOR

A run of identical errors minutes apart usually means the scheduled sync is retrying the same account. Read the newest one — the earlier ones are the same fault.

03 Trace a delivery to its Shopify order

WHO	WHEN	NEEDS
Warehouse staff	While picking	An imported order

Inventory ▸ Deliveries ▸ *open a transfer*

1. On the delivery form, find the **Shopify** group. It appears only on transfers linked to a Shopify order.
2. **Shopify Order** is the store's own order number, for example **#1149** — quote this to the customer, not the Odoo reference.
3. **Shopify Fulfillment Status** tells you whether Shopify already considers it shipped.
4. To go the other way, open the sale order and use its **Shopify** tab, which lists every mapping with its GID and sync time.

YOU'LL KNOW IT WORKED

The delivery shows a Shopify order number, and the sale order carries a green **Shopify** ribbon.

Phase B

Products onto Shopify

Automatic pushing is **off** by default, so getting products onto Shopify is normally a deliberate act. Pick the procedure that matches how many products you are moving.

04 Push one product to Shopify

WHO	WHEN	NEEDS
Sales Manager	Per new product	A connected account

Sales ▸ Products ▸ *open a product* ▸ Push to Shopify

1. Fill in the fields Shopify receives before pushing: **Name**, **Sales Description**, **Product Category**, **Sales Price** and the product image.
2. Click the **Push to Shopify** button in the button box at the top right of the form.
3. In the dialog, choose the **Shopify Store**. It is pre-selected when only one store is connected; with several you must choose deliberately.
4. Click **Push** and wait for the notification.

YOU'LL KNOW IT WORKED

A green notification confirms one product pushed, and a **Shopify** group appears near the product title showing the store and the Shopify product ID.

WATCH FOR

The push sends the product, its variant prices, its stock and its image in that order. A product with no variants still pushes fine, but a product whose price sits only on an attribute extra needs that attribute saved first.

05 Push a batch of products

WHO	WHEN	NEEDS
Sales Manager	Range launches	Procedure 04 once

Sales ▸ Products ▸ *select rows* ▸ ⚙ Actions ▸ Push to Shopify

1. Switch to list view and tick the products you want to send.
2. Open the **Actions** (gear) menu and choose **Push to Shopify**.
3. Choose the target store. The products you selected are listed read-only so you can confirm the set before committing.
4. Click **Push**. Each product is sent independently, so one failure does not stop the rest.

YOU'LL KNOW IT WORKED

The notification reports how many pushed. If any failed it turns orange and stays on screen — check the log (procedure 02) for the specific products.

WATCH FOR

Push large batches outside business hours. Each product is several API calls, and image uploads are the slow part.

06 Pull one product back from Shopify

WHO

Sales Manager

WHEN

After editing in Shopify

NEEDS

A linked product

Sales ▸ Products ▸ *open a product* ▸ Pull from Shopify

1. Confirm the product is linked — the **Shopify** group near the title shows the store and product ID.
2. Click **Pull from Shopify** in the button box.
3. The product's fields, variant structure and images are refreshed from Shopify.

YOU'LL KNOW IT WORKED

A notification confirms the number of mappings pulled, and the change you made in Shopify is now visible in Odoo.

WATCH FOR

This does nothing on a product with no Shopify mapping — you get a "No mapping found" warning rather than a new import. Push it first (procedure 04).

07 Two-way sync the product list

WHO

Sales Manager

WHEN

Weekly, or after bulk edits

NEEDS

Linked products

Sales ▸ Products ▸ Sync Shopify Products

1. Filter the list to the products you want to reconcile. The button acts on what is in front of you, so filter first.
2. Click **Sync Shopify Products** in the list header.
3. Each linked product is pushed to Shopify and then pulled back, for every connected store.
4. Read the breakdown in the notification: pushed, pulled, not linked, failed.

YOU'LL KNOW IT WORKED

Pushed and pulled counts match the number of linked products in your filter, with *Not linked (skipped)* accounting for the rest.

WATCH FOR

Unlinked products are skipped, never created. Use the **Shopify Linked** filter to see which products this button will actually touch.

08 Queue a full catalogue resync

WHO	WHEN	NEEDS
Sales Manager	Rarely — see below	A connected account

Shopify ▸ Configuration ▸ Shopify Accounts ▸ Full Product Resync

1. Use this when you suspect Odoo has drifted from Shopify across many products — not for a single product, which procedure 06 handles.
2. Click **Full Product Resync** and accept the confirmation.
3. The button returns immediately. The work is queued and starts on the next auto-sync cycle, within about 30 minutes.
4. To speed up a large catalogue, first enable **Skip product images in bulk sync** in Settings, then sync images later.

YOU'LL KNOW IT WORKED

A "Full Resync Scheduled" notification appears, and over the following cycles the **Last Product Sync** timestamp and product counter keep advancing.

WATCH FOR

A big catalogue takes several cron cycles to finish, and that is normal — the connector remembers its position and resumes each time rather than starting over. Do not keep clicking the button.

Stock levels

READ THIS BEFORE TOUCHING INVENTORY SYNC

Inventory sync **overwrites**; it never merges. Whichever side you nominate as the source replaces the other side's numbers. Decide the direction before you run anything, and test on a staging database first.

09 Decide which system owns stock

WHO	WHEN	NEEDS
Sales Manager	Once, at go-live	Procedure 12

Settings > Shopify Connector > Sync Defaults

1. Answer one question: is Odoo or Shopify the truth for on-hand quantities? For most businesses running warehouses in Odoo, it is Odoo.
2. Set **Cron inventory direction** accordingly — **Pull from Shopify** means Shopify wins, **Push to Shopify** means Odoo wins.
3. Decide whether the scheduled sync should handle inventory at all with **Run inventory sync in scheduled cron**. It is off by default because inventory is the slowest phase.
4. Leave it off and run inventory manually (procedures 10 and 11) until you are confident the direction is right.

YOU'LL KNOW IT WORKED

Stock levels in Shopify and Odoo agree after a run, and they stay in agreement without anyone correcting them by hand.

10 Pull stock levels from Shopify

WHO	WHEN	NEEDS
Sales Manager	Shopify is the source	Procedures 09, 12

Shopify > Operations > Sync Now

1. Choose the **Shopify Account**.
2. Set **Direction** to **Pull from Shopify**.
3. Set **What to Sync** to **Inventory Only**.
4. Click **Start Sync**. Shopify's quantities are written over Odoo's on-hand figures for every mapped variant.

YOU'LL KNOW IT WORKED

A "Sync Complete" notification appears, and spot-checking a product's on-hand quantity in Odoo matches Shopify.

WATCH FOR

Nothing happens if no location mappings exist — there is nowhere to read quantities for. Run procedure 12 first.

11 Push stock levels to Shopify

WHO	WHEN	NEEDS
Sales Manager	Odoo is the source	Procedures 09, 12

Shopify ▸ Operations ▸ Sync Now

1. Choose the account, set **Direction** to **Push to Shopify** and **What to Sync** to **Inventory Only**.
2. Click **Start Sync**. Odoo's available quantity is written to Shopify for each mapped (warehouse, location) pair.
3. For a large catalogue, tick **Batch Mode** and run the wizard repeatedly until nothing changes — each run processes one batch.

YOU'LL KNOW IT WORKED

Shopify's inventory for a spot-checked variant matches Odoo. Quantities that already agreed are skipped rather than rewritten, so a second run should be quick and quiet.

WATCH FOR

You rarely need this on a running system: validating a transfer or editing a quant already pushes the affected products automatically. Use it for the initial load or after bulk stock corrections.

12 Add a warehouse and map its Shopify location

WHO	WHEN	NEEDS
Sales Manager	Per new location	A connected account

Shopify ▸ Configuration ▸ Shopify Accounts ▸ Sync Locations

1. If the new location was created in **Shopify**, click **Sync Locations**. Matching Odoo warehouses are linked by name; anything unmatched gets a new warehouse created for it.
2. If the new warehouse was created in **Odoo**, click **Push Warehouses to Shopify** to create the corresponding Shopify location from the warehouse address.

3. Check the **Location Mappings** table on the account form. Every warehouse you sync stock for needs a row with a Shopify location ID.
4. Each warehouse may appear once, and each Shopify location once — Odoo refuses duplicates.

YOU'LL KNOW IT WORKED

A "Locations Synced" notification reports what was created or matched, and the mappings table lists every location with both a name and an ID.

WATCH FOR

Location mapping is the prerequisite for all inventory sync. If stock sync appears to do nothing, check this table first.

Orders coming in

Orders normally arrive by themselves within seconds, pushed by Shopify. These procedures are for when you need to reach for them, or when one needs a hand.

13 Pull orders on demand

WHO	WHEN	NEEDS
Sales Manager	An order has not appeared	A connected account

Shopify ▸ Operations ▸ Sync Now

1. Choose the account, leave **Direction** on **Pull from Shopify** and set **What to Sync** to **Orders Only**.
2. Click **Start Sync**.
3. Check **Shopify ▸ Operations ▸ Orders** for the order, then follow its **Sale Order** link.

YOU'LL KNOW IT WORKED

The order appears in the Shopify Orders list with its financial and fulfillment status badges, linked to an Odoo sale order.

WATCH FOR

Orders cannot be pushed to Shopify — they originate there. Choosing **Push** with **Orders Only** is refused; the wizard switches the scope back to **All** for you.

14 Re-pull orders you think were missed

WHO	WHEN	NEEDS
Sales Manager	After an outage	Procedure 13 tried first

Settings ▸ Shopify Connector ▸ Order sync lookback window

1. A normal sync only asks Shopify for orders changed since the last successful run, minus a small overlap. If Odoo was down for a while, widen that overlap.
2. Set **Order sync lookback window** to comfortably cover the gap — for a day-long outage, **48** hours.
3. Save, then run procedure 13, or click **Sync All** on the account.
4. Put the value back to **1** afterwards. A permanently wide window makes every routine sync re-examine orders it has already handled.

YOU'LL KNOW IT WORKED

The missing orders appear in the Shopify Orders list. Orders already imported are recognised and skipped rather than duplicated.

WATCH FOR

If an order still will not appear, the problem is not the window — check the log (procedure 02) for an import error naming that order.

15 Release an order stuck in Draft

WHO

Sales Manager

WHEN

Flagged by the log

NEEDS

Procedure 02

Sales ▸ Orders ▸ *open the quotation*

1. A paid Shopify order is normally confirmed automatically. When it stays in **Draft**, the connector wrote a warning to the log explaining why — nearly always a missing route or procurement rule on a product.
2. Read that log entry (procedure 02) to get the exact reason and the product involved.
3. Fix the underlying configuration — usually the product's **Routes** on the Inventory tab, or its product type.
4. Return to the order and click **Confirm**.

YOU'LL KNOW IT WORKED

The order moves to **Sales Order** and a delivery transfer is created, ready to pick.

WATCH FOR

Confirming without fixing the cause means the next order for that product sticks too. Fix the product, not just the order.

16 Check a draft order completed in Shopify

WHO

Sales Manager

WHEN

After invoicing a draft order

NEEDS

An imported draft order

Shopify ▸ Operations ▸ Orders

1. Shopify draft orders arrive as **draft** Odoo quotations, badged as draft orders in the list.
2. When the draft order is completed in Shopify it becomes a real order with a new number. The connector re-points the existing record rather than creating a second one.

3. Confirm this happened: the row's order number should change from the draft form (D1234) to the real order number (#1149), and the financial and fulfillment badges should appear.
4. If the number has not changed, run procedure 13 to pull orders again.

YOU'LL KNOW IT WORKED

One record, not two: the same Odoo sale order now shows the real Shopify order number.

WATCH FOR

Two Odoo orders for one Shopify sale means the promotion did not happen. Cancel the duplicate rather than deleting it, and check the log.

Shipping & money

17 Fulfil a Shopify order

WHO	WHEN	NEEDS
Warehouse staff	Per shipment	A confirmed order

Inventory > Deliveries > *open the transfer* > Validate

1. Pick the goods and set the done quantities as you would for any delivery.
2. Enter the **Tracking Reference** and choose the **Carrier** *before* validating, so both travel to Shopify with the fulfillment.
3. Click **Validate**. The connector creates the Shopify fulfillment automatically — there is no separate button to press.
4. Confirm the **Shopify Fulfillment Status** on the transfer now reads **FULFILLED**.

YOU'LL KNOW IT WORKED

The Shopify order shows as fulfilled in the Shopify admin, with the tracking number attached, and the customer receives Shopify's shipping notification.

WATCH FOR

Validating a transfer also pushes the affected stock levels to Shopify. That is expected, and it happens after the move is committed so the quantity sent is the final one.

18 Add or correct a tracking number

WHO	WHEN	NEEDS
Warehouse staff	After shipping	Procedure 17

Inventory > Deliveries > *open the done transfer*

1. Open the already-validated delivery.
2. Type the new value into **Tracking Reference** and save.
3. The connector updates the tracking on the existing Shopify fulfillment. The customer is *not* re-notified by default.
4. If tracking was added in Shopify instead, it flows the other way on the next order sync and lands in the same field.

YOU'LL KNOW IT WORKED

The Shopify order's fulfillment shows the corrected tracking number, and the log records the update.

WATCH FOR

This needs a fulfillment to already exist. Adding tracking to a transfer that was never validated changes nothing in Shopify — validate it first.

19 Post a refund credit note

WHO	WHEN	NEEDS
Accountant	Per Shopify refund	A posted invoice

Accounting ▸ Customers ▸ Credit Notes

1. When a refund is issued in Shopify, the connector creates a **draft** credit note against the order's posted invoice. Nothing is posted without you.
2. Find it by searching the **Reference** field for `shopify-refund-` . Each carries the Shopify refund ID, so it is traceable both ways.
3. Check the lines. A refund naming specific items is matched line by line from the invoice; a shipping or goodwill refund arrives as a single amount line.
4. Adjust if needed, then **Confirm** to post it.

YOU'LL KNOW IT WORKED

The credit note is posted and reduces the customer's balance by the amount refunded in Shopify.

WATCH FOR

No credit note appears if the order has no *posted* invoice — the refund is written to the log for manual handling instead (procedure 02). Invoice Shopify orders promptly and this will not arise. Re-sent refund notifications never create a second credit note.

20 Confirm a cancellation reached Odoo

WHO	WHEN	NEEDS
Sales Manager	Per cancellation	A linked order

Sales ▸ Orders ▸ *find the order*

1. Cancel the order in Shopify as usual. Odoo follows automatically — you do not cancel it twice.
2. Check the Odoo sale order has moved to **Cancelled**. Orders that were voided without payment and never shipped are cancelled too.
3. If it is still open, run procedure 13. The connector re-examines orders that look unchanged but still need cancelling, which repairs a missed notification.

4. An order Odoo refuses to cancel — because it is already invoiced or delivered — is left alone and recorded as a warning in the log. Handle those as credit notes (procedure 19).

YOU'LL KNOW IT WORKED

The Odoo sale order state is **Cancelled**, and its delivery transfer is cancelled with it.

Housekeeping

21 Send an Odoo quotation to Shopify

WHO	WHEN	NEEDS
Sales Manager	Phone & offline sales	Linked products

Sales > Orders > *new quotation*

1. Build the quotation using products that are already linked to Shopify.
2. **While it is still a draft or sent quotation**, set **Shopify Account** in the Shopify group near the top of the form. This field locks once the order is confirmed.
3. Click **Confirm**. The order is exported to Shopify as a **draft order**.
4. For several at once, select them in the list and use **Actions > Sync Orders to Shopify**. Only orders in Sales Order or Locked state are sent.

YOU'LL KNOW IT WORKED

A draft order appears in the Shopify admin, and the Odoo order's **Shopify** tab lists the new mapping.

WATCH FOR

Set the Shopify Account *before* confirming. Once the order leaves draft the field is read-only, and an order with no account set is never exported — with no error to tell you so.

22 Clean up errored products

WHO	WHEN	NEEDS
Sales Manager	Monthly, or after errors	A connected account

Shopify > Configuration > Shopify Accounts > Clean Up Errored Products

1. Products that failed to sync are flagged in error and retried automatically every hour. Give that a chance to work before intervening.
2. What the retry cannot fix is a product deleted in Shopify — it will never succeed. Click **Clean Up Errored Products** and accept the confirmation.
3. Every errored mapping is checked against Shopify. Those whose Shopify product is gone have their Odoo product archived and the mapping removed; the rest are left for the retry.
4. Read the notification: it reports how many were checked, archived, removed, and left pending.

YOU'LL KNOW IT WORKED

The notification accounts for every errored mapping, and the error count falls on the next sync.

WATCH FOR

This archives Odoo products. Archiving is reversible, but check the notification's count against what you expected before moving on.

23 Re-register webhooks after a URL change

WHO	WHEN	NEEDS
Administrator	After a domain or host move	A public HTTPS URL

Shopify ▸ Configuration ▸ Shopify Accounts ▸ Register Webhooks

1. Registered webhooks contain your Odoo address. Change the address and Shopify keeps calling the old one, so orders stop arriving in real time.
2. Have the system parameter `baseup_shopify.webhook_base_url` updated to the new public HTTPS address first.
3. Click **Register Webhooks**. All existing webhooks on the store are cleared and re-registered against the new address.
4. Test it properly: place a real test order in Shopify and confirm it reaches Odoo within seconds.

YOU'LL KNOW IT WORKED

A notification confirms the webhooks registered, and a test order appears in Odoo without anyone running a sync.

WATCH FOR

A successful registration only proves Shopify accepted the address. Only a test order proves it is reachable. If registration is refused, the address is not public HTTPS — that is one for whoever runs your hosting.

24 Walk a new user through the tours

WHO	WHEN	NEEDS
Sales Manager	Per new operator	Developer mode

Settings ▸ Technical ▸ Tours

1. Four guided walkthroughs drive the real screens, so a new operator learns by doing rather than reading.
2. Start `shopify_onboarding` for connecting a store end to end — credentials, OAuth, webhooks and locations.

3. Then `shopify_first_product_push` (procedure 04),
`shopify_multi_warehouse_setup` (procedure 12) and
`shopify_first_order_review` (procedure 03).
4. Run them on a training database. The onboarding tour walks up to real buttons that connect a live store.

YOU'LL KNOW IT WORKED

The new operator can complete procedures 01 to 04 unaided.

