https://cloudblue.com

Documentation Modules Expanse

Getting Started With Expanse

This article has been generated from the online version of the documentation and might be out of date. Please, make sure to always refer to the online version of the documentation for the up-to-date information.

Auto-generated at August 28, 2026

Expanse lets you build a working fulfillment integration for your product without writing code. You describe your API, an AI agent drafts the integration logic, and you refine it in plain language until it's ready to go live — typically in hours rather than the weeks a custom-coded integration takes.

This guide walks through setting up your first automation, from picking a product to promoting it to production.

Before you start

Make sure you have the following ready:

- **A product already configured in Connect.** Expanse automates fulfillment for a product that exists in your Connect catalog — it doesn't create the product itself. Your parameters and items must be set up first.
- **An OpenAPI specification for your API.** Expanse reads this document (either OpenAPI Spec URL or OpenAPI Spec File in .json or .yaml / .yml) to understand your API's endpoints, request/response shapes, and authentication scheme, and uses it to draft the integration. The spec should include clear endpoints for the operations you want to automate (for example, creating a subscription, cancelling one, or applying a change).
- **API credentials for your system.** You'll connect these through the Vault module during setup so Expanse can call your API on your behalf. Expanse doesn't store credentials itself.

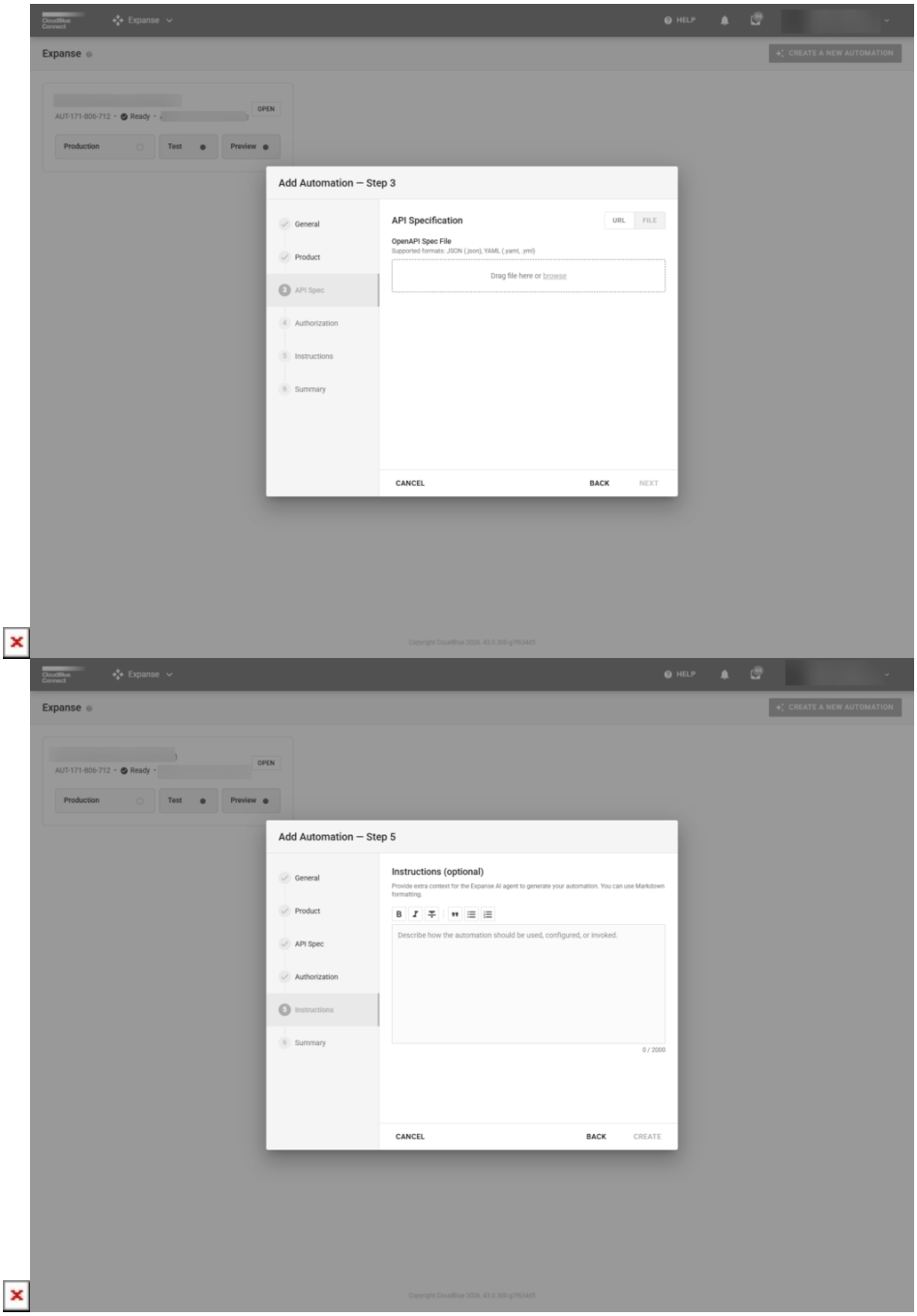
Note: If your API doesn't cleanly map one endpoint to one Connect fulfillment action, generation may not produce a usable result and may require user-input refinement.

Step 1: Start a new automation

From the Expanse module in the Connect Vendor portal, select the product you want to automate and start a new automation. This opens the **General** tab, where you'll:

1. Confirm the product the automation is bound to.
2. Upload or link your OpenAPI specification.
3. Optionally add a few sentences of context for the agent — for example, business rules that aren't obvious from the API alone.

You don't need to write any code or configure infrastructure at this step.



Step 2: Let the agent generate a draft

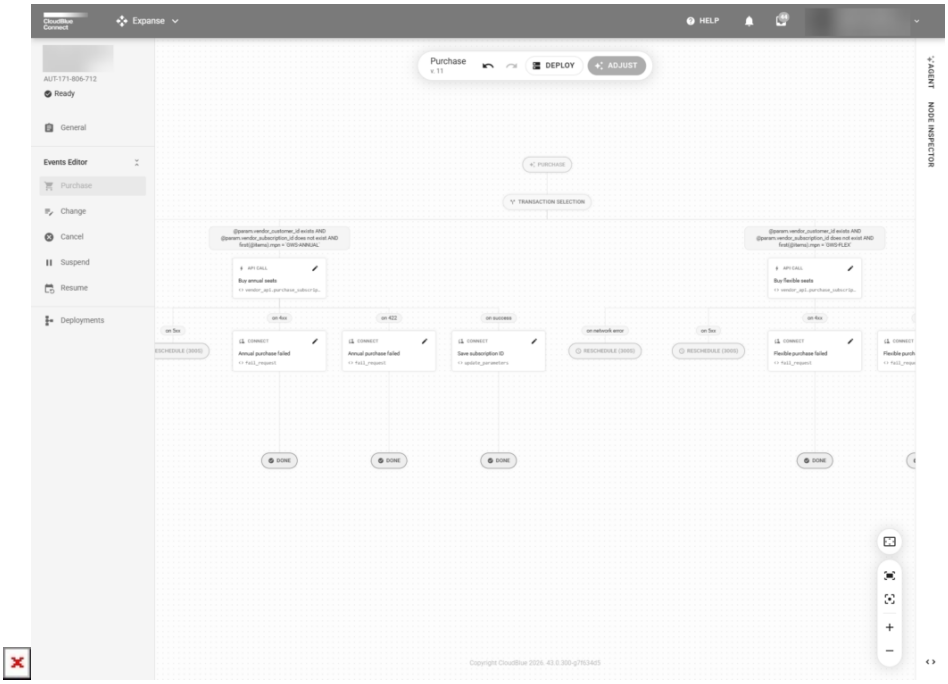
Once your spec is in place, the agent reads your product configuration and your API and drafts a fulfillment flow for each lifecycle event your product needs to handle — for example, purchase, change, and cancellation. This typically takes a minute or more; you can leave the page and come back once it’s ready.

Step 3: Review the automation in the Workspace

Open the **Workspace** tab to see the result. Each event is shown as a node graph — the sequence of API calls, decisions, and parameter saves the agent has drafted. You don’t need to read code to check the agent’s work: the graph shows you what will

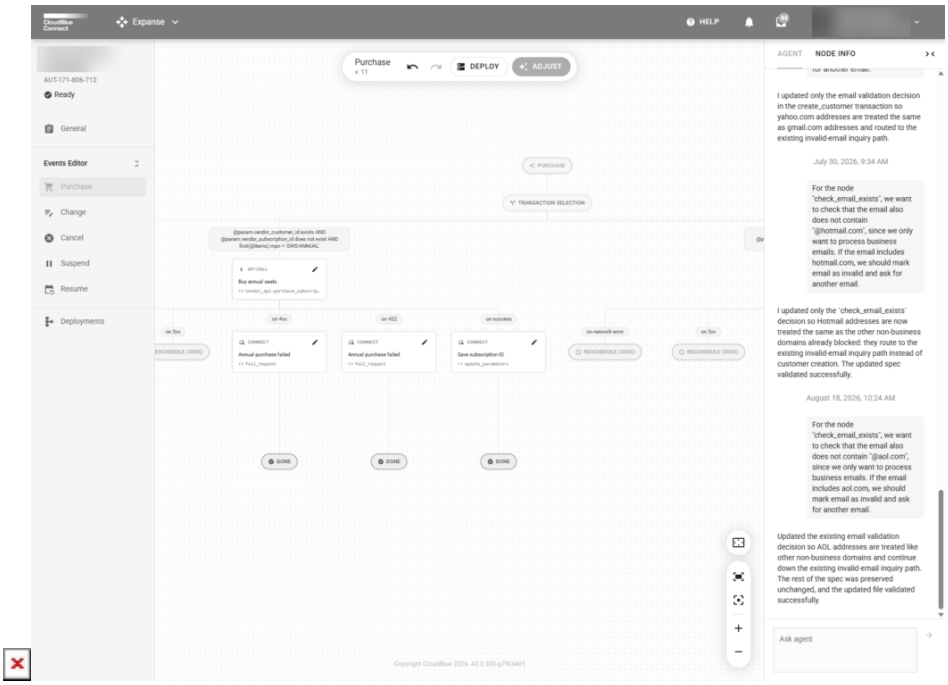
happen, step by step, when a customer places an order.

Switch between event types (purchase, change, cancel, and so on) to review each flow individually.



Step 4: Refine in plain language

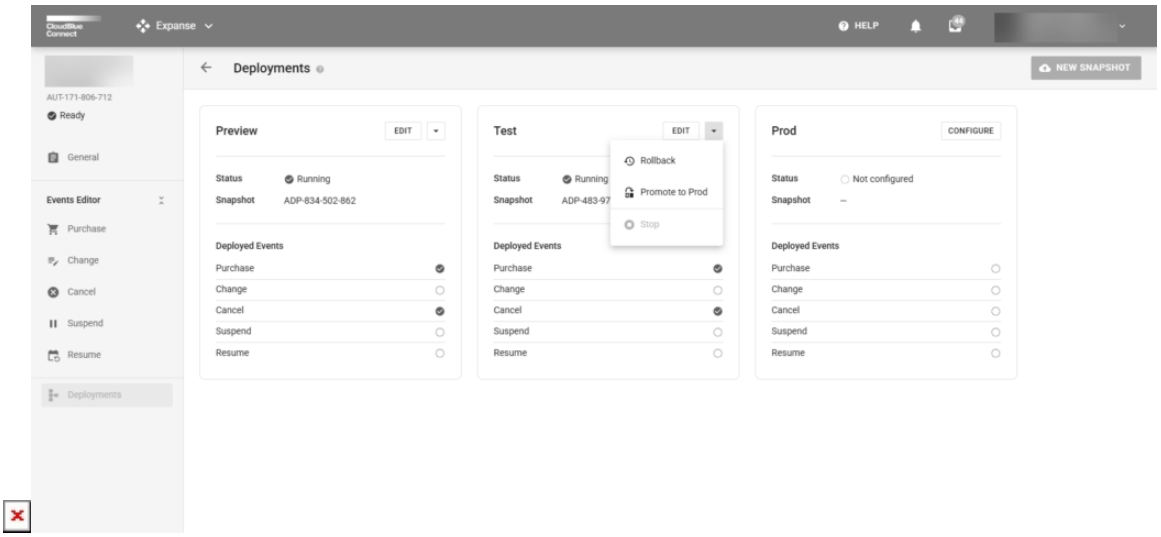
If something isn't right, describe the change you want in natural language — for example, *"don't create a new customer if one already exists with this email, reuse it instead."* The agent updates the affected part of the graph accordingly.



Every refinement is saved as a revision, so you can always see how the automation evolved and roll back to an earlier version if needed.

Step 5: Try it out

Promote your automation to the **Preview** or **Test** environment from the **Deployments** tab to run it against real test requests. Watch how it behaves — which API calls it makes, what it saves, and what the outcome is — and go back to Step 4 to refine further if it doesn't behave the way you expect.



Step 6: Go live

When you're satisfied with how the automation performs in Test, promote it to **Production**. Connect will route live customer orders through the deployed version. If you need to undo a production change, you can roll back to a previous deployment at any time — no rebuild required.

Note: If a connector (EaaS Extension) already handles the same product and event in a given environment, it takes precedence and continues to run — your automation stays published but won't receive traffic until the connector is deactivated. This lets you build and test safely alongside an existing integration.