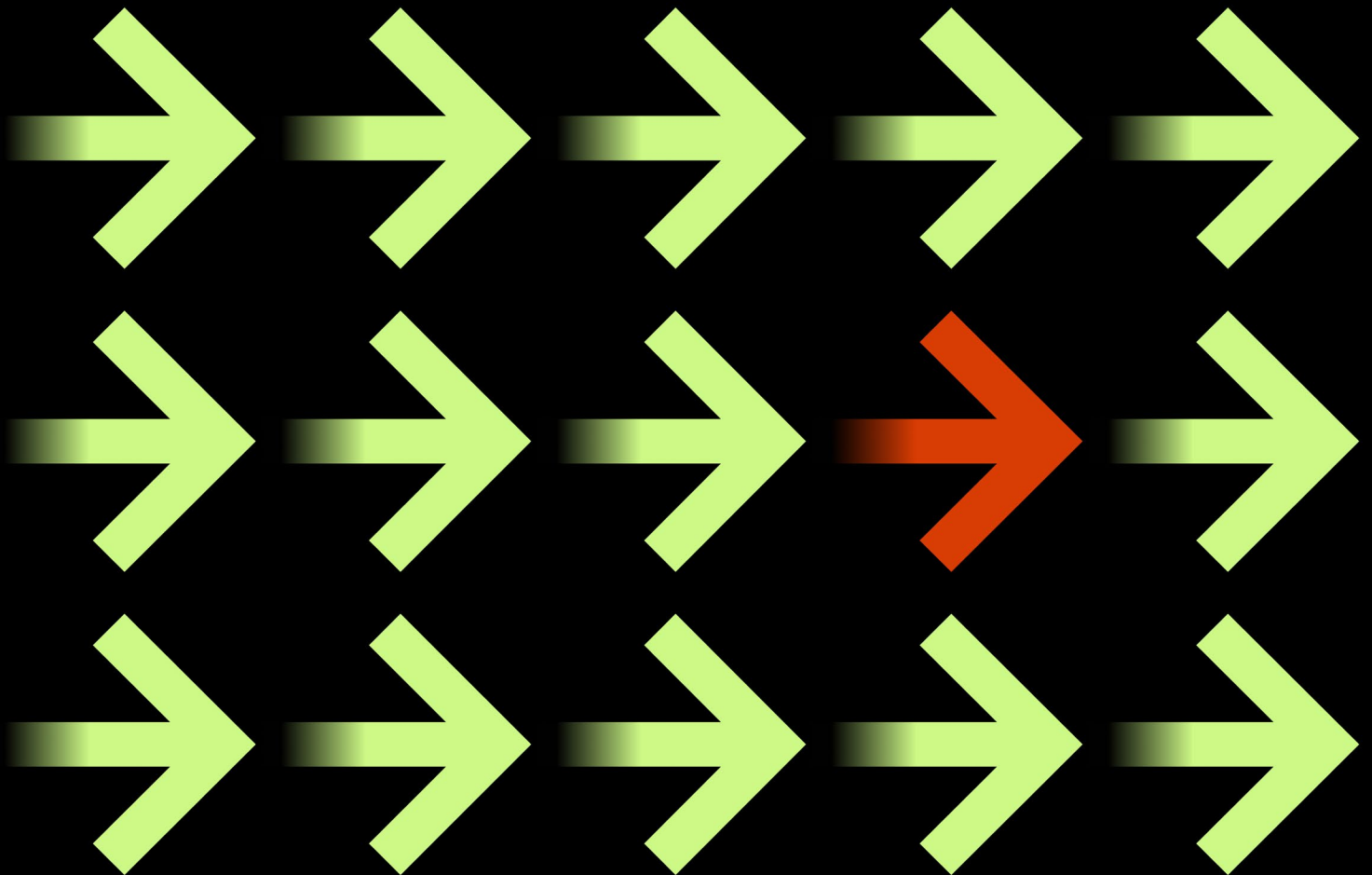


# Adobe Commerce to Shopify

A proven migration framework for  
technical decision makers moving  
from Adobe Commerce to Shopify.



# Contents

|    |                                                                    |    |
|----|--------------------------------------------------------------------|----|
| 01 | Executive summary                                                  | 3  |
| 02 | Adobe Commerce to Shopify:<br>Time to value                        | 6  |
| 03 | Core architecture differences:<br>Adobe Commerce Cloud vs. Shopify | 8  |
| 04 | Shopify's answers to complex<br>ecommerce challenges               | 15 |
|    | B2B and multi-market                                               | 18 |
|    | Enterprise growth                                                  | 21 |
| 05 | How to migrate from<br>Adobe Commerce to Shopify                   | 15 |
|    | Phase 1: Discovery                                                 | 26 |
|    | Phase 2: Planning                                                  | 28 |
|    | Phase 3: Execution                                                 | 30 |
|    | Phase 4: Post-migration optimization                               | 33 |
| 06 | Plan your next migration steps                                     | 36 |

## 01

---

# Executive summary

The mounting complexity of Adobe Commerce (formerly Magento) is forcing technical leaders into a critical decision: invest heavily in platform overhaul or migrate to a more agile solution. For many scaling brands, this growing infrastructure burden, upgrade fatigue, and integration overhead have made migration to Shopify not just attractive, but necessary.

This guide is designed for technical decision-makers actively planning the transition. Rather than debating whether to migrate, it focuses on executing a successful migration, honing in on the real challenges, critical trade-offs, and step-by-step decisions that determine whether your migration delivers on its promise or becomes a costly setback.

Specifically, we'll focus on these areas that heavily inform the success of your migration:

**01**

Platform performance and architecture trade-offs

**02**

Shopify's native tools for enterprise, B2B, and multi-market operations

**03**

A step-by-step migration execution framework

## Methodology

The recommendations in this guide are based on direct experience from engineers, architects, and migration specialists who have moved complex brands from Adobe Commerce to Shopify. Input comes from:

- **Domaine**, a global Shopify engineering partner trusted by enterprise brands for complex migrations and high-performance builds
- **Technical leaders**—VPs of engineering, tech leads, and solutions architects—who have completed hundreds of replatforming projects
- Shopify’s own **product and engineering teams**, with insight into platform capabilities and migration pitfalls

## Key findings

Here’s what you need to know before you start:

### Infrastructure and operational overhead

Shopify reduces the operational burden of managing the hosting, scaling, and patching that come with Adobe Commerce. You trade some low-level control for managed infrastructure, but most teams find the reduction in manual work outweighs the trade-off. This guide will show you how to navigate those differences.

### Development workflow and speed

Migrating to Shopify means adopting new patterns for development, deployment, and QA. You trade off some flexibility in infrastructure but gain faster release cycles and less time spent maintaining environments. The migration process itself demands careful mapping of customizations and workflows.

### Data structure and migration complexity

Adobe Commerce’s entity-attribute-value (EAV) data model is flexible but complex. Shopify’s product schema is simpler, but requires flattening and mapping of attributes. Data transformation is a core part of any migration, and this guide outlines practical steps to get it right.

## Integrations and customizations

Most Adobe Commerce integrations have Shopify equivalents, but not all. Some custom modules will need to be rebuilt as apps or via Shopify Functions. Expect to audit your tech stack and plan for some reengineering.

## Total cost of ownership and resource allocation

Shopify's managed approach reduces infrastructure and maintenance costs. However, migration involves upfront investment in planning, data transformation, and team enablement. The long-term payoff is less time spent on platform upkeep.

## Next steps

As mentioned previously, this guide also provides a step-by-step migration framework, including:

### 01

How to audit your current Adobe Commerce environment and identify what needs to change

### 02

How to plan and sequence your migration to minimize risk and downtime

### 03

How to map data and customizations to Shopify's architecture

### 04

How to validate, test, and optimize your new Shopify store

### 05

What metrics to track post-migration to measure success

Think of this guide as a technical playbook. It's built to help you move with confidence, avoid common pitfalls, and make informed decisions at every stage of your migration.

02

# Adobe Commerce to Shopify: Time to value

For technical leaders planning an Adobe Commerce exit, timeline pressure is real: how quickly can you migrate and start seeing ROI on Shopify?

The answer is highly compelling. Among enterprise brands, even extremely customized Adobe Commerce estates with large catalogs, complex B2B logic, and deep ERP or OMS integrations can usually be migrated to Shopify in **nine months or less**. Mid-market to large brands usually complete the transition in three to six months.

| COMPLEXITY TIER     | TYPICAL ADOBE COMMERCE PROFILE                                                                                                                                                                      | ESTIMATED SHOPIFY MIGRATION WINDOW* |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------|
| Tier 1 — Standard   | <ul style="list-style-type: none"> <li>• Fewer than 10,000 SKUs</li> <li>• Five or fewer integrations</li> <li>• Theme-level customizations only</li> </ul>                                         | 12–20 weeks                         |
| Tier 2 — Advanced   | <ul style="list-style-type: none"> <li>• 10,000–50,000 SKUs</li> <li>• 6 to 15 integrations</li> <li>• Custom modules</li> <li>• Light B2B features</li> </ul>                                      | 20–30 weeks                         |
| Tier 3 — Enterprise | <ul style="list-style-type: none"> <li>• More than 50,000 SKUs</li> <li>• More than 15 integrations</li> <li>• Heavy B2B and multi-store</li> <li>• Multiple custom modules and services</li> </ul> | 30–36 weeks (~9 months)             |

\*Assumes a dedicated client team, a solution partner experienced with both platforms, and no blackout periods such as Black Friday and Cyber Monday. Actual timelines may vary.

## Now, a look at how Shopify simplifies project compression:

### Managed infrastructure

No server sizing, container orchestration, or patch sequencing.

### Flat data model

Moving from EAV to Shopify's product and variant schema eliminates most schema refactoring.

### Prebuilt connectors

Tax and payment integrations are often plug and play or require only light configuration.

### Theme and app modularity

The front end, custom apps, and integrations run in parallel because they are isolated from Shopify core updates.

### Automated testing and deployment

Using CI/CD pipelines and tools like the Shopify GitHub App and Shopify CLI, teams can push updates to development stores or unpublished themes, accelerating QA and user-acceptance testing cycles.

Because Shopify removes infrastructure gating and offers modular integration patterns, **time to value is measured in months, not years.** Even the most complex Adobe Commerce Cloud implementations can be achieved and often result in net-new capabilities within a nine-month window.

# 03

## Core architecture differences Adobe Commerce Cloud vs. Shopify

Although Adobe Commerce Cloud and Shopify both reduce some hosting overhead compared to purely self-hosted solutions, their technical foundations and DevOps models vary greatly. In this table, we compare the two platforms from an engineering perspective.

| ASPECT        | ADOBE COMMERCE CLOUD                                                                                 | SHOPIFY                                                                                           |
|---------------|------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|
| Hosting       | PaaS: Adobe-managed cloud, but still environment-level configs                                       | SaaS: Fully hosted, no server management                                                          |
| Data model    | Entity-attribute-value (EAV) for products, can be large and complex                                  | Flat product structure, variants, and metafields                                                  |
| DevOps        | Code merges via Git to integration, staging, production; some manual patching and environment tuning | Deploy themes, custom apps, or headless front ends; Shopify handles scaling and OS-level patching |
| Customization | Potentially deeper code-level control; updates may conflict with custom modules                      | Apps, Shopify Functions, or headless solutions; isolates custom code from core platform updates   |
| Performance   | Cloud autoscaling is possible but may require additional environment-level tuning                    | Automatic scaling with 99.9% uptime;* CDN built-in                                                |

#### COMMON MIGRATION CONSIDERATIONS: TEAM SKILL SET

##### CONSIDERATION:

How do teams adapt to Shopify technical languages vs. languages on Adobe Commerce?

##### SOLUTION:

Shopify mostly uses Liquid/Javascript/CSS for theming, plus React + Remix frameworks (e.g., Hydrogen) for headless setups. While that's a familiar framework for many dev teams, you'll want to ensure your developers undergo training on Shopify's theme architecture, webhooks, and API structure.

## Hosting and deployment model

**Adobe Commerce Cloud:** Operates as a PaaS offering. While you no longer manage bare metal or your own AWS/Azure accounts, you still maintain environment configs, manage deployments via Git, and handle certain aspects of patching.

**Shopify:** Uses a fully hosted, multi-tenant SaaS infrastructure. Shopify handles the underlying servers, scaling, CDN distribution, and security patching. Your engineering efforts revolve around front-end and custom app development rather than environment maintenance.

## DevOps workflows

### COMMON MIGRATION CONSIDERATIONS: HIGH-TRAFFIC EVENTS

#### CONSIDERATION:

“Adobe Commerce Cloud includes autoscaling features, but we have used separate staging and production environments. How does Shopify compare?”

#### SOLUTION:

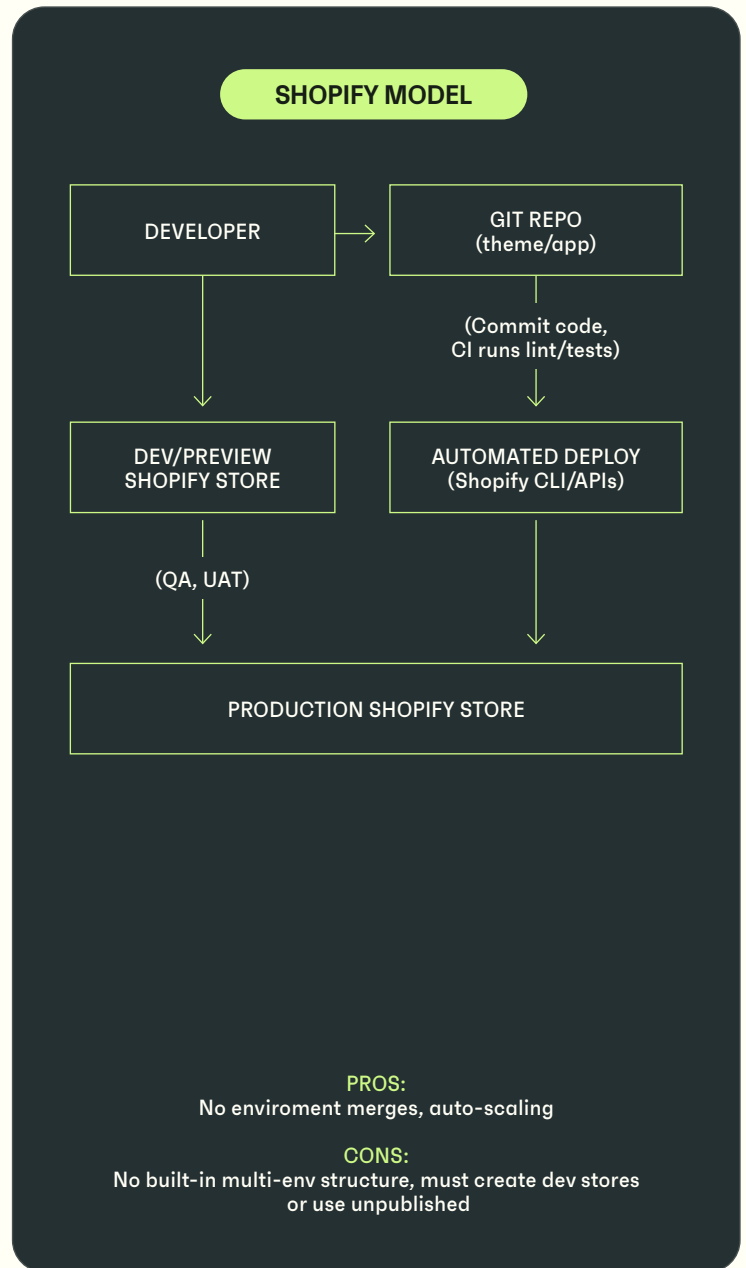
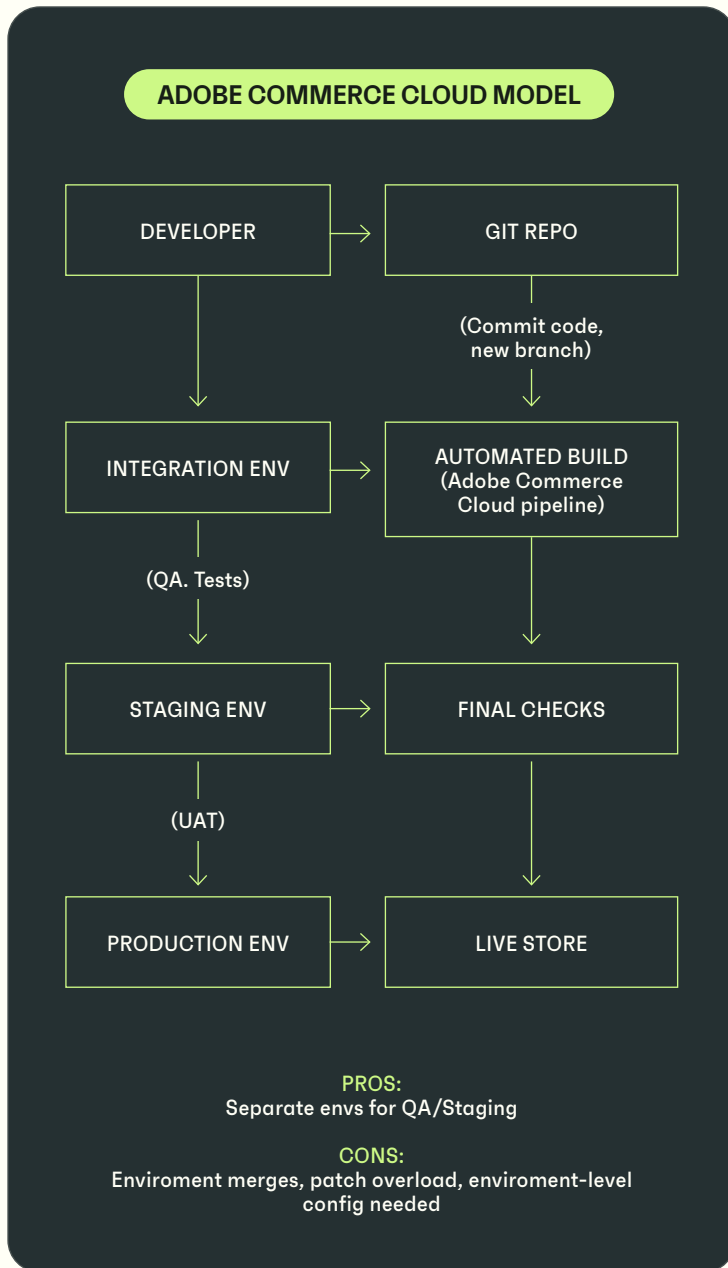
Shopify auto-scales behind the scenes. You do not manage the underlying container or VM environment. This reduces the need for environment-level testing of server capacity and streamlines build/merge cycles.

**Adobe Commerce Cloud:** You push code to a Git repository that triggers builds and deployments. You manage multiple environments (integration, staging, production) that are each tied to a specific branch. While some tasks are automated, you will still configure variables, environment relationships, and service-scaling options.

**Shopify:** You do not manage servers or container orchestration. You deploy theme changes, custom apps, or headless storefronts via the Shopify CLI and version control. Continuous integration processes can still be used for your apps and front-end code, but infrastructure-level tasks are handled by Shopify.

## CI/CD and environment differences

You can think of Adobe Commerce Cloud as having a built-in environment pipeline you are obligated to follow, while Shopify pushes you to create your own staging or development setup (via dev stores or unpublished theme branches). This approach is more flexible for smaller or agile teams, but it can feel less structured to teams used to a mandatory “Integration -> Staging -> Production” pipeline.



## Comparing CI/CD and environment models

| CATEGORY                          | ADOBE                                                                                                                                               | SHOPIFY                                                                                                                                                                                                                                  |
|-----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Environment structure</b>      | Multiple environments (integration, staging, production)                                                                                            | No native “staging vs. production” at the platform level. Typically a single live store, plus dev stores or theme branches                                                                                                               |
| <b>Version control and merges</b> | Git-based merges from integration to staging to production                                                                                          | Themes can be version-controlled externally. Multiple unpublished theme versions or separate dev stores for testing                                                                                                                      |
| <b>Build process</b>              | Adobe’s pipeline compiles code, runs Composer, generates static assets.                                                                             | For theme code, a local or hosted CI server can compile CSS and JS, and push to Shopify using CLI/API. For custom apps, you deploy to your own hosting.                                                                                  |
| <b>Deployment steps</b>           | <ol style="list-style-type: none"> <li>1. Push to Git</li> <li>2. Adobe builds</li> <li>3. Staging tests</li> <li>4. Merge to production</li> </ol> | <ol style="list-style-type: none"> <li>1. Push theme or app code to Git</li> <li>2. CI runs tests</li> <li>3. Automatic or manual push to dev store or unpublished theme</li> <li>4. Publish theme or update production store</li> </ol> |
| <b>DevOps overhead</b>            | Must manage environment configs, patch merges, and scaling settings.                                                                                | Shopify auto-scales and handles patches. DevOps overhead is mostly around theme/app pipeline and version control.                                                                                                                        |
| <b>Rollback strategy</b>          | Revert merges/commits in Git. Potentially revert environment states.                                                                                | Revert to an older theme version or code commit. Dev store can be used to restore or test changes.                                                                                                                                       |
| <b>Common tools</b>               | Native Adobe build pipeline, Jenkins, CircleCI                                                                                                      | Shopify CLI, GitHub Actions, Sidekick, Cursor                                                                                                                                                                                            |
| <b>Key differences</b>            | Environment merges can become a bottleneck, specialized environment management is needed.                                                           | Fewer environment merges, rely on dev/staging stores or unpublished themes. Patching and scaling are fully automated by Shopify.                                                                                                         |

## Application layer and extensions

**Adobe Commerce Cloud:** Offers a rich ecosystem of modules. However, updates can break dependencies or conflict with custom modules. Maintaining these customizations can require specialized PHP expertise. Also, while cloud-based, you still rely on the core Adobe Commerce framework, which can be resource-heavy.

**Shopify:** Allows access to the Shopify App Store with more than 8,000 integrations. Many technology providers found in the Adobe Commerce ecosystem are also Shopify Plus Technology Partners. These partners offer dedicated apps or enhanced support specifically optimized for Shopify, allowing brands to retain familiar tools while improving reliability and performance post-migration.

For enterprise scenarios, you can build custom apps or even a headless architecture using Storefront and Admin APIs. Shopify Functions extends the platform further, allowing custom discount logic, shipping rules, or checkout enhancements without modifying core code.

### COMMON MIGRATION CONSIDERATIONS: CUSTOM MODULES AND COMPLEX INTEGRATIONS

#### CONSIDERATION:

“Our team wrote proprietary modules in PHP that handle advanced business logic.”

#### SOLUTION:

Evaluate whether you can combine Shopify Functions with custom apps. Many popular modules in Adobe Commerce Cloud have app equivalents on Shopify. Where none exist, you can build a tailor-made solution using your team’s preferred framework.

## Data structure

**Adobe Commerce Cloud:** Retains the Adobe Commerce data model, which can be complex due to the large number of tables and the entity-attribute-value (EAV) approach for products. Cloud hosting mitigates some performance issues, yet data architecture can still be a bottleneck if not optimized.

**Shopify:** Employs a simpler product schema (products, variants, and metafields) that streamlines inventory, pricing, and attribute management. This eliminates the overhead of EAV, making data migration and performance tuning more straightforward once you map your attributes correctly.

## Data differences

**Adobe Commerce Cloud:** In Adobe Commerce Cloud, products (entities) have attributes stored in separate tables depending on data type (varchar, int, decimal, text, etc.). The platform assigns each attribute an ID, which references the product in an attribute-specific table. Below is a simplified view of how data might be structured:

### 1. catalog\_product\_entity

- entity\_id: 123
- sku: HOOD123
- type\_id: simple

### 2. catalog\_product\_entity\_varchar

- attribute\_id=45 might refer to the "Name" attribute.
- attribute\_id=47 might refer to "Brand."
- attribute\_id=52 might refer to "Color."

| value_id | attribute_id | store_id | entity_id | value                |
|----------|--------------|----------|-----------|----------------------|
| 501      | 45           | 0        | 123       | "Hoodie With Pocket" |
| 502      | 47           | 0        | 123       | "Example Brand"      |
| 503      | 52           | 0        | 123       | "Black"              |

### 3. catalog\_product\_entity\_text

- attribute\_id=58 might refer to a "Long Description" or "Material Details."

| value_id | attribute_id | store_id | entity_id | value                                   |
|----------|--------------|----------|-----------|-----------------------------------------|
| 601      | 58           | 0        | 123       | "80% Cotton. 20% Polyester. Wash cold." |

### 4. catalog\_product\_entity\_decimal

- attribute\_id=61 might refer "Price."

| value_id | attribute_id | store_id | entity_id | value |
|----------|--------------|----------|-----------|-------|
| 701      | 61           | 0        | 123       | 39.99 |

In an EAV model, each product attribute is stored in a specialized table. This allows for high flexibility but can be more complex to query and maintain. As you add custom attributes (e.g., brand, fabric, style), you increase the number of rows across these tables.

**Shopify:** In Shopify, each product typically has a single record containing core fields (title, description, price) plus optional fields (variants, options, metafields) for more specialized data. You do not have separate tables for each attribute type. Instead, you have a unified structure, and if you need custom fields, you add them as **metafields**.

## JSON-Like View of the Same Product in Shopify

```
{
  "title": "Hoodie With Pocket",
  "body_html": "<p>80% Cotton. 20% Polyester. Wash cold.</p>",
  "vendor": "Example Brand",
  "product_type": "Apparel",
  "variants": [
    {
      "sku": "H00D123",
      "price": "39.99",
      "option1": "Black - Small"
    },
    {
      "sku": "H00D123-M",
      "price": "39.99",
      "option1": "Black - Medium"
    }
  ],
  "options": [
    {
      "name": "Size",
      "values": ["Small", "Medium", "Large", "XL"]
    },
    {
      "name": "Color",
      "values": ["Black"]
    }
  ],
  "metafields": [
    {
      "namespace": "custom",
      "key": "material_details",
      "value": "80% Cotton. 20% Polyester",
      "type": "single_line_text_field"
    }
  ],
  "tags": "Winter, Hooded, Pocket"
}
```

### title

Combines what was the “Name” attribute in Adobe Commerce Cloud ([attribute\\_id=45](#))

### body\_html

Stores your product description or material details that previously lived in the [catalog\\_product\\_entity\\_text](#) table

### vendor

Equivalent to “Brand” (you had [attribute\\_id=47](#) in Adobe Commerce Cloud)

### variants & options

Flatten your color and size attributes so that each variant has a unique price, SKU, or inventory level.

### metafields

For more specialized attributes. For example, if you need additional metadata about this product (fabric composition, wash instructions), you might store that in a custom metafield.

### tags

Used for filtering or grouping. In Adobe Commerce Cloud, you might have multiple attributes or layered navigation filters that you now condense to a simpler system of tags.

So, why flatten EAV Attributes, and what are the benefits?

### 1. Performance and simplicity:

- In Adobe Commerce Cloud, you may run multiple joins across EAV tables to retrieve all product data.
- Shopify stores data in a more standardized format, simplifying queries and improving performance.

### 2. Data mapping:

- Each EAV attribute in Adobe Commerce Cloud must be mapped to either a standard Shopify field (title, description, variant, etc.) or a metafield.
- This step often involves cleaning up redundant or unused attributes that can clutter your new store.

### 3. Variants vs. separate products:

- In Adobe Commerce Cloud, you might store size and color as separate attributes in EAV.
- In Shopify, these typically become product **options** (Size, Color), generating variants automatically.

### 4. Legacy attributes:

- Adobe Commerce Cloud merchants often create many custom attributes over time. Some might be irrelevant to your new Shopify store. The migration phase is an opportunity to prune or consolidate them.

In sum, Adobe Commerce Cloud's EAV model theoretically gives you unlimited flexibility, but it can become cumbersome with large catalogs or numerous custom attributes. During a migration to Shopify, you flatten these attributes into Shopify's simpler product fields (title, variants, options, metafields) and optionally use tags or collections for further categorization. This flattening process is a crucial step in ensuring your new Shopify store is both logically organized and optimized for performance.

## Practical steps to flatten EAV data

#### Inventory all attributes

Export a list of all EAV attributes and decide which ones are critical, which ones can map to variants/options, and which ones are best stored as metafields or tags.

#### Consolidate overlapping attributes

If you have multiple attributes for "Color" or "Material," standardize them.

#### Create a data-mapping document

For each attribute, specify where it goes in Shopify (e.g., title, vendor, metafield, option, variant).

#### Transform and load

Use an ETL (extract, transform, and load) process or custom scripts to pull data from Adobe Commerce Cloud, transform it, and insert it into Shopify, taking advantage of the GraphQL Admin API.

## Example data mapping table

| ADOBE COMMERCE CLOUD ATTRIBUTE               | EAV TABLE                           | PURPOSE                        | SHOPIFY EQUIVALENT                   | NOTES                                             |
|----------------------------------------------|-------------------------------------|--------------------------------|--------------------------------------|---------------------------------------------------|
| <b>Name</b><br>(attribute_id=45)             | catalog_product_entity_varchar      | Main product title             | <b>title</b>                         | Combine brand with title if needed                |
| <b>Brand</b><br>(attribute_id=47)            | catalog_product_entity_varchar      | Product brand                  | <b>vendor</b>                        | Typically mapped to the “vendor” field            |
| <b>Color</b><br>(attribute_id=52)            | catalog_product_entity_varchar      | Variation dimension            | <b>option/variant (Color)</b>        | Might combine with Size in a single variant set   |
| <b>Material details</b><br>(attribute_id=58) | catalog_product_entity_text         | Extended product info          | <b>metafield (material_details)</b>  | Freed from EAV, stored in a single text metafield |
| <b>Price</b><br>(attribute_id=61)            | catalog_product_entity_decimal      | Base price                     | <b>variants[].price</b>              | Each variant can have its own price in Shopify    |
| <b>Seasonal category (custom)</b>            | Possibly a custom EAV attribute set | Used for marketing or grouping | <b>tags or collection assignment</b> | Convert to Shopify tags or auto-collections       |

By systematically flattening your attributes in this manner, you transform a sometimes unwieldy EAV structure into a more streamlined Shopify product setup. This not only simplifies everyday store management but can also improve performance, since you no longer need multiple table joins to retrieve basic product information.

# 04

## Shopify's answers to complex ecommerce challenges

When evaluating Shopify for complex requirements, technical leaders often ask tough questions about enterprise capabilities. Let's walk through some of the most common concerns we hear from Adobe Commerce teams, as well as how Shopify addresses them.

### B2B and multi-market

*"Can Shopify handle advanced B2B features like bulk ordering, custom pricing, or net terms?"*

Shopify offers a dedicated B2B on Shopify feature set for enterprise businesses on the Plus plan. This includes:

**Company profiles and user roles:** Associate multiple buyers with a single company and control permissions for purchasing or payment.

**Custom price lists:** Assign distinct price tiers to specific companies or groups.

**Net payment terms:** Offer payment schedules such as Net 30 or Net 60 directly through the Shopify admin.

**Bulk ordering options:** Configure bulk purchase workflows with custom apps or Shopify Functions.

If you are currently using Adobe Commerce to handle complex quoting or unique workflows, you can often replicate these processes with Shopify's B2B features or extend them via a custom app or headless approach.

## *“Can Shopify provide a true B2B buyer portal?”*

Thanks to B2B on Shopify, you can create an experience tailored to business buyers while still leveraging the same admin panel:

### **Company and buyer accounts**

Each B2B customer can log in to view negotiated pricing, order history, and other account-specific details.

### **Checkout customizations**

Shopify Functions or third-party apps let you shape payment and shipping flows, factoring in tax exemptions, volume discounts, or specialized shipping rules for B2B buyers.

### **Hybrid B2B and DTC**

A single Shopify store can serve both audiences. You can segment them by login, price lists, or even by giving B2B buyers a unique storefront URL that is still managed under one admin.

This approach streamlines operations while providing discrete user experiences for your wholesale or corporate customers.

## *“Do you need multiple Shopify stores to handle multiple B2B markets?”*

Depending on your strategy, you can manage multiple markets (and currencies) from a single store using Shopify Markets, or you may prefer separate stores for operational or regulatory reasons. For example:

**Single-store approach:** Serve all B2B customers in one place using localized storefronts and price lists. This keeps everything in one admin but still addresses region-specific content, currencies, and languages.

**Multiple-store approach:** If you run truly distinct product lines, separate brand experiences, or different compliance requirements, having multiple Shopify stores can make sense. You can still manage them under one organization and share certain apps or integrations.

In either scenario, you have fewer DevOps considerations compared to orchestrating multiple Adobe Commerce environments that each require environment variables, code merges, or specialized modules.

## *“Can B2B on Shopify match Adobe Commerce’s level of back-end customization for complex purchase flows?”*

B2B on Shopify is designed for enterprise workflows, and you can extend it further as needed:

### **Shopify Functions**

Overhaul discount logic, payment flows, and shipping rules without forking core code

### **Custom apps**

Build an unpublished or published app for advanced quoting, RFP processes, or multi-level approval workflows.

### **Headless commerce**

If you have large-scale or highly specialized front-end requirements, decouple the storefront. Your B2B portal can run on a framework like React, Next.js, or Remix, with Shopify as the commerce engine.

### **Integration with ERP/CRM**

Sync complex pricing, product data, or contract terms from your ERP to Shopify using GraphQL APIs plus webhooks. This ensures your B2B environment stays updated in real time.

While Shopify is not open source, it provides a stable foundation. Many brands find that building custom logic or bridging external systems in a microservices architecture like Shopify is simpler to maintain than deeply modifying Adobe Commerce or writing custom PHP modules.

## *“Does Shopify support multi-language or multi-currency operations like Adobe Commerce does?”*

Shopify offers flexible solutions for global commerce through [Shopify Markets](#), which centralizes your global selling strategy by letting you configure currencies, domains, and languages in a single interface:

### **Enable local**

**currencies:** Allow international buyers to see and pay in their local currency.

### **Provide multi-language storefronts:**

Configure translation tools (native or via apps) for localized content.

### **Handle duties and**

**taxes:** Collect duties and VAT upfront, improving the checkout experience for international buyers.

### **Manage multiple**

**markets:** Set up custom pricing, inventory rules, or shipping options for each market while handling everything from one admin.

### **Work across multiple business**

**entities:** Connect multiple business entities with their own bank accounts to a single store for payouts in domestic currency.

Beyond Markets, you can also set up multiple storefronts:

#### Create regional variations

Set up distinct stores for different geographical markets or brand identities.

#### Maintain separate domains

Control multiple storefronts while managing all operations from a centralized admin.

#### Configure B2B specifics

Implement region-specific pricing, catalogs, shipping, and taxes through B2B on Shopify.

#### Choose architecture approach

Opt for traditional multi-store setup or use headless commerce to manage multiple regions from a single codebase.

This flexibility allows you to choose the best structure for your business—whether that’s a single store with multiple localizations or separate stores for different markets—all while maintaining centralized control over your global operations.

## Enterprise growth

### “Can Shopify’s API limits support enterprise growth?”

Yes, Shopify has API rate limits, but the API architecture is built for scale.

- API thresholds depend on the Shopify plan.
- The Storefront API and GraphQL Admin API are built to handle high-volume operations.
- Webhooks minimize API calls by enabling event-driven syncs, reducing unnecessary requests.

Most enterprise brands can integrate seamlessly with the right data strategy.

| API                        | RATE-LIMITING METHOD  | STANDARD LIMIT      | ADVANCED SHOPIFY LIMIT | ADVANCED SHOPIFY LIMIT | ADVANCED SHOPIFY LIMIT |
|----------------------------|-----------------------|---------------------|------------------------|------------------------|------------------------|
| GraphQL Admin API          | Calculated query cost | 100 points/second   | 200 points/second      | 1000 points/second     | 2000 points/second     |
| Storefront API             | None                  | None                | None                   | None                   | None                   |
| Payment Apps API (GraphQL) | Calculated query cost | 27300 points/second | 27300 points/second    | 27300 points/second    | 109200 points/second   |
| Customer Account API       | Calculated query cost | 100 points/second   | 200 points/second      | 200 points/second      | 400 points/second      |

## *“Does Shopify have more downtime due to its SaaS nature?”*

The reality is that Shopify has a highly reliable infrastructure with 99.9% historical uptime and automatic scaling to handle spikes in traffic.

Plus, there are no manual updates. Unlike Adobe Commerce, Shopify handles updates seamlessly without disruption to the store:

### **Global CDN**

- Provides fast load times and reliability across regions.
- Shopify’s cloud infrastructure reduces risk compared to self hosted platforms like Adobe Commerce.
- Shopify provides constant system status tracking at [shopifystatus.com](https://shopifystatus.com).

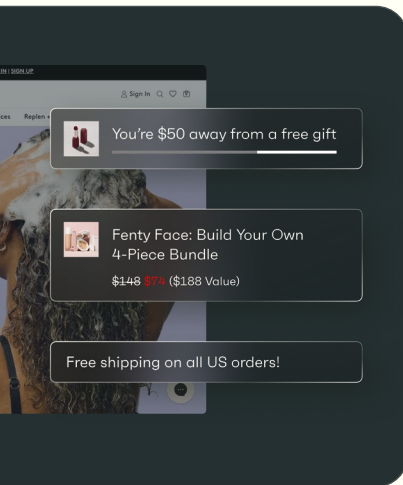
## *“Does Shopify have complex product promotion capabilities?”*

Shopify’s Plus plan offers flexible discounting and promotional tools, with space to build out hyper-specific promotions based on your needs.

For the highest level of complexity, you can work with a combination of built-in features and third-party apps. Everything from discount codes to automatic and custom discounts is possible with Shopify Functions and external apps—all without requiring custom code.

- Shopify’s native discount engine supports tiered pricing, bulk discounts, and automatic promotions.
- Shopify Functions enable fully customized discount logic and checkout flows.
- Seamless integrations allow for multilayered promotions across DTC, B2B, and omnichannel, ensuring a holistic customer experience across channels.

Shopify provides the flexibility to execute advanced, high-volume promotional strategies with ease.



## *“How does Shopify handle large-scale businesses?”*

Shopify powers high-volume enterprises globally. Household names like Allbirds and [Benchmade](#) trust Shopify to drive their operations.

Influencers with a massive global reach like Kylie Jenner who uses Shopify for her “Kylie Skin” brand know that Shopify can handle the launch of their products—and scale up with their rapid growth.

While both Shopify and Adobe Commerce can support enterprise-level operations, Shopify offers superior hands-off scalability. The platform handles traffic spikes with ease, and without the manual server management and tuning often required by Adobe Commerce. This reduced DevOps overhead makes Shopify an attractive option for businesses prioritizing operational efficiency. There’s no limit on the size, scope, or complexity of business that can grow using Shopify—and the platform’s toolset allows for integrated operations across retail, DTC, and B2B operations.

On top of all that, Shopify is simple and streamlined to use for lightning-fast changes that don’t bottleneck your business.



### **CASE STUDY**

## Repositioning Benchmade on Shopify

Benchmade partnered with Domaine to migrate their storefront and reposition themselves as a DTC brand. Originally on Adobe Commerce (Magento), Benchmade found site maintenance consumed excessive time and resources. To prioritize growth, ideation, and testing, they decided to replatform—and Shopify offered everything they needed..

The new Shopify website had to resonate with a broader range of customers, while highlighting Benchmade’s expertise in high-quality, custom manufacturing and industry leadership. The replatform focused heavily on product customization and unique 3D renderings to set the brand apart.

The result is an ecommerce experience that showcases just how capable Shopify is when it comes to meeting complex brand needs.

## *“How much control over infrastructure does Shopify give companies?”*

Shopify offers a huge amount of flexibility through app integrations, APIs, and Shopify Functions.

Adobe Commerce, on the other hand, requires brands to build much of their infrastructure from scratch. While this allows for more intensive customization in some areas, it comes with a massive bump in complexity. It also slows down the process of making changes and requires more specialized knowledge at every step.

Adobe’s open-source model allows for broad integration possibilities, but the trade-off is a heavier ongoing commitment from your DevOps team. It’s an overall slower, more complex process to launch and make changes.

And although Shopify’s checkout is less customizable, that’s by design. These guardrails allow brands to safely make changes without risking checkout stability. After years of optimization, Shopify’s checkout delivers industry-leading conversion rates—proof that “if it ain’t broke, don’t fix it.”

Even so, Shopify offers a range of extensions and custom validation rules to tailor the checkout experience.

Replatforming isn’t just about data migration—it’s about building a stronger, more flexible foundation for your brand. Shopify creates opportunities to remove workarounds, unlock scalability, and give your team a true competitive edge.

# 05

## How to migrate from Adobe Commerce to Shopify

### Overview of the migration process

#### PHASE 1: DISCOVERY

High-level site and requirement analysis

Considerations:

- Catalog size
- Legacy customizations
- Third-party integrations

#### PHASE 2: PLANNING

Detailed migration roadmap and timeline

Considerations:

- Budget and resource planning
- Custom feature replication
- Peak sales periods

#### PHASE 3: EXECUTION

Data migrations, store configuration, and integration testing

Considerations:

- Minimizing downtime
- Testing payments and shipping flows
- Finalizing redirects

#### PHASE 4: POST-MIGRATION OPTIMIZATION

Ongoing performance and conversion optimization

Considerations:

- Analytics and KPI tracking
- Ongoing dev support

## PHASE 1: DISCOVERY

Before migrating, you need a **complete audit of your existing platform** to identify potential challenges and opportunities.

Measure twice, cut once.

First, you need to understand the current state of your digital ecosystem. Start by breaking down which elements are baked into your Adobe Commerce account and which elements you've custom-built.

**Where does your site use custom code? Which functions are supported by third-party apps?**

Make sure you document this clearly.

You'll also need to identify any data or content that might need to be restructured to fit into the new site. That could include product data, blogs, and CMS pages.

### COMMON MIGRATION CONSIDERATIONS: INTEGRATIONS

#### CONSIDERATION:

How do brands migrate third-party technology from Adobe Commerce?

#### SOLUTION:

First, you'll list out your existing tech stack. Many of those tools will already directly integrate with Shopify, and others will have clear equivalents. For more specific needs, webhooks and custom code can fill the gap. From there, Shopify offers an enormous range of tools built from the ground up for ecommerce, with a focus on future optimization and development.

## Key considerations:

- **Catalog size and complexity:** Adobe Commerce uses an EAV model for products, which often requires flattening before importing into Shopify's simpler product schema.
- **Legacy customizations:** Adobe Commerce commonly uses custom modules. Determine which modules map to Shopify features, which will require third-party apps, and which might require a custom app or headless approach.
- **Third-party integrations:** List existing integrations (ERP, OMS, CRM, marketing automation) and document whether they have direct Shopify equivalents that need to be rebuilt or reconnected.

### ACTION STEPS:

- ☐ Map out existing custom features and workflows.
- ☐ Identify data structures, including attribute sets that may need flattening or renaming for Shopify.
- ☐ Determine which integrations need reconfiguration or replacement
- ☐ Identify must-have vs. legacy functionalities to determine what carries over.
- ☐ Establish data-migration needs—products, customers, orders, and content.

## PHASE 2: PLANNING

This phase is where you put pen to paper (or keyboard to screen, if you will). This step is for limiting issues by setting clear migration goals, allocating resources, and defining a feasible timeline. The goal of this phase is to outline your technical blueprint for migrating data and rebuilding custom features in Shopify.

### COMMON MIGRATION CONSIDERATIONS: COMPLIANCE AND SECURITY

#### CONSIDERATION:

“Adobe Commerce Cloud helps meet certain compliance standards, but we still maintain partial responsibility for OS-level patches.”

#### SOLUTION:

Shopify is Level 1 PCI DSS-compliant and handles operating system patches automatically. You must still confirm that your chosen apps and custom integrations adhere to data privacy and security best practices.

## Key considerations:

- **Budget and resource allocation:** Mapping internal and external development support (Remember to check in with other departments to ensure company-wide participation in launch efforts.)
- **Custom feature replication:** Rebuilding or optimizing custom workflows on Shopify.
- **High-risk sales periods:** Scheduling migration to high-traffic windows (like BFCM and seasonal sales).

## ACTION STEPS:

- ☐ **Feature mapping:** Map Adobe Commerce features (native and custom) to Shopify equivalents or workflows. Check if your custom modules can be handled by existing Shopify apps, or if you will need to build a custom-coded app. Adobe Commerce can have advanced promotional features. Compare these to Shopify Functions or third-party discount apps.
- ☐ **Timeline:** Set a company-wide migration timeline with clear milestones to stay on track.
- ☐ **Define a testing framework:** Ensure avoidance of disruption to the customer experience.
- ☐ **Data mapping:** Build a product attribute matrix. Decide which Adobe Commerce attributes translate into Shopify metafields, product variants, or separate product records.
- ☐ **DevOps strategy:** Shift your mindset from environment management in Adobe Commerce to a combination of Shopify's environment (for themes and basic setup) plus your own CI/CD system (for custom apps or front-end code in a headless approach).
- ☐ **Security and compliance:** Adobe Commerce Cloud provides some automated security updates, but you may still manage certain OS-level patches. Shopify takes over these patches entirely. Evaluate your brand's compliance requirements (PCI, GDPR, etc.) to ensure Shopify meets all standards.

## PHASE 3: EXECUTION

This is where data is transferred, integrations are built, and launch is prepped for your Shopify store.

You can migrate data using Shopify's Admin API or an extract, transform, and load (ETL) computing process, which may work better for larger catalogs. Domaine can help decide which direction is best for your unique brand.

As you work through this process, consistent testing will be vital. Alongside your internal processes, you'll need to test critical integrations: make sure your payment gateways, shipping solutions, CRM, and ERP connections all work seamlessly.

### COMMON MIGRATION CONSIDERATIONS: PROJECT MANAGEMENT

#### CONSIDERATION:

What are the resource implications of running an existing website while also providing migration support?

#### SOLUTION:

Part of replatforming is operating with a split focus for a period. There will be an additional DevOps challenge in both maintaining your existing site while building and honing your Shopify store. That process will look different from brand to brand, but there are a few fundamentals to keep in mind:

- First, make sure someone has ownership over the migration. That person will schedule meetings, check in with different teams, set up testing periods, and generally manage the process as it moves along.
- Second, set aside time for training. While working with Shopify generally reduces the load on dev teams compared to Adobe Commerce, make sure that everyone is comfortable and familiar with the differences between platforms.

### Key considerations:

- **Minimizing downtime:** Migrate sections of the business in strategic phases (e.g., product catalog first, then checkout, then full storefront), allowing for controlled testing and minimal risk to the customer experience.
- **Testing payment and shipping flows:** Confirming transactional and operational accuracy to ensure orders flow through smoothly.
- **Finalizing redirects:** Mapping old URLs to new Shopify counterparts, preventing broken links and preserving SEO rankings.

## ACTION STEPS:

- ☐ **Migrate products, customers, orders, and historical data:**
  - ☐ Choose an approach that suits your catalog size. Smaller catalogs may rely on CSV exports. Larger ones should use an ETL pipeline or custom scripts.
  - ☐ Migrate products, customers, and orders in distinct phases. Validate each step in a staging or development store.
  - ☐ Watch out for differences in how content is structured (for example, Adobe Commerce might store CMS pages differently than Shopify Pages).
- ☐ **Quality assurance:** Conduct extensive QA on checkout, search, and integrations.
- ☐ **SEO:** Set up SEO redirects and 301 mappings to maintain rankings.
- ☐ **Integration setup:**
  - ☐ Adobe Commerce often uses cron-based tasks, whereas Shopify uses webhooks for real-time events. Migrate relevant logic to handle event-driven triggers.
  - ☐ Update ERP or OMS connections to pull order data from Shopify's GraphQL API.
  - ☐ Configure shipping and tax providers. Shopify offers integrated solutions, but you may need to replicate advanced shipping rules using apps or Shopify Functions.
- ☐ **Testing and validation:** Test in a non-production Shopify store before syncing real customer data or linking your primary domain. Inspect all integrations for correct data flow. If you find gaps, address them in your custom or third-party apps. Perform at least one test run of the data migration to catch any unexpected mapping issues.
  - ☐ **Functional tests:** Confirm that your cart, checkout, discount logic, and email notifications function as expected.
  - ☐ **Performance tests:** Compare site loading speeds between your new Shopify store and your Adobe Commerce site. Shopify's global CDN typically reduces TTFB, but you can confirm this using performance logs or an APM tool.
  - ☐ **SEO considerations:** Implement 301 redirects to maintain existing search rankings and watch Google Search Console for any crawl errors.

## PHASE 4: POST-MIGRATION OPTIMIZATION

The real exciting work begins once you've made the change. Once you've migrated onto Shopify, you can use the real-time performance metrics to iterate on the shopping experience.

Remember: Your bounce rate [nearly triples](#) if your page takes longer than three seconds to load.

Learn more about how Shopify can help boost your site speed [here](#)

### Tap into AI-powered tools and powerful analytics

Alongside improvements to site speed, you can build on your existing user experience, test, and tweak your funnel for better conversions. Shopify gives your team access to innovative features designed to streamline ongoing operations and decision-making:

- **Shopify Sidekick:** Use Shopify's built-in AI assistant to automate routine tasks, generate content (such as professional quality images, product descriptions, or marketing emails), and receive step-by-step help for store management. Sidekick can help teams quickly resolve issues, optimize storefronts, analyze performance, and implement best practices directly within the Shopify admin.
- **ShopifyQL reporting:** ShopifyQL is a query language designed specifically for commerce analytics. You can build custom reports and dashboards, track granular KPIs, and surface actionable insights from your store's data. This tool makes it easier to monitor site performance, customer trends, and post-migration business outcomes without relying on third-party BI tools.

Integrating these features into your workflow can help your team respond faster, uncover growth opportunities, and ensure your migration delivers measurable, ongoing value. The metrics that you track here will be the KPIs for the overall success of the migration approach.

## Post-migration success metrics

After migration, there will be some areas where the difference will be obvious. You'll probably feel that parts of your business run more smoothly, and that parts of your team are less stressed and have more space to focus on what's important.

Tracking the difference means setting clear KPIs. While those will vary from brand to brand depending on the goals that drove the replatforming, consider tracking:

- **Load times:** You'll want to compare these to your site before making the move to Shopify, and continue to track and optimize as you move forward.
- **Bounce rates:** Your bounce rate is a related metric, and also worth tracking.
- **Conversion rates:** There will be a whole range of factors influencing these (ad copy, landing page functionality, checkout flow, etc.), and it will take additional A/B testing to keep honing in on the optimum funnel.
- **Time to market for new features:** Compare how quickly your team can deploy changes on Shopify versus the multi-environment structure and branching model in Adobe Commerce.
- **Ongoing maintenance:** Track how many dev hours go toward routine patches, environment troubleshooting, or hotfixes. Shopify reduces these overheads substantially.
- **Performance improvement:** Use historical speed and conversion data from Adobe Commerce for baseline comparisons.
- **Scalability and reliability:** Monitor uptime and incident frequency. If you were managing environment-level escalations on Adobe Commerce, you may find far fewer interruptions on Shopify.

Shopify powers operational efficiencies such as seamless connectivity between B2B, DTC, and POS systems, unlocking a more agile and scalable commerce strategy.

## Key considerations:

- **Analytics and KPI tracking:** Measuring conversion rates, site speed, and uptime (at minimum)
- **Ongoing development support:** Ensuring your team is continuously trained on the Shopify platform (including new and evolving updates and integrations)
- **Customer experience improvements:** Iterating the platform based on user behavior and conversion rates

### ACTION STEPS:

- ☐ Implement real-time analytics tracking to measure post-migration success.
- ☐ Conduct post-launch audits for SEO, performance, and integrations.
- ☐ Set up ongoing reviews to optimize CX and functionality—ensure CX, marketing, and operational teams are involved in this review cycle.
- ☐ Plan for incremental improvements, such as upgrading your theme, adding new third-party apps, or branching into B2B or wholesale capabilities.

# 06

## Plan your next migration steps with Domaine and Shopify

Migrating your brand to Shopify means unifying your team around a new vision. It's a step forward in the fundamentals of your business—building a stronger base from which every other element of your brand can grow.

That means building out flexibility in your processes as well as your platform. The infinite flexibility of open-source software doesn't mean much if your business grinds to a halt every time you need to make changes.

Rising to the top means working with a partner that can support you through it all: from the little changes to the major pivots. Shopify is built for brands that move quickly and change often.

It's about building a home for your brand that supports your customers wherever they shop, for more conversions and a better return, every time.

If you're ready to learn more about migrating from Adobe Commerce to Shopify, get in touch today.

[Contact Shopify](#)

And if you're ready to get started today with a partner who knows what it takes, let's get to work.

[Contact Domaine](#)

### Sources

[“What Is Page Speed & How to Improve It”](#) Marilyn Wilkson, Semrush Blog, January 20, 2025.