Don Stefani

Senior developer — custom Shopify applications and the AWS integrations behind them.

CAPABILITY BRIEF
AVAILABLE FOR NEW CLIENT WORK
UPDATED AUGUST 2026

Boise, Idaho · Mountain Time · Remote / don@donstefani.com / donstefani.com /

linkedin.com/in/donstefani

30+	**500,000+**	**5**	**Daily**
years of engineering experience	products in the largest live catalog synced	shipped systems — 2 embedded Shopify Admin apps, 3 AWS pipelines	production cadence of the catalog pipelines in use today

FOCUS

I build custom Shopify applications and the AWS integrations behind them — for catalog, order, and operational workflows that off-the-shelf apps and theme customization can't safely handle. Merchants and agencies, from a hundred products to hundreds of thousands.

Custom Shopify applications

Embedded Admin apps for operational work the Shopify admin doesn't cover — bulk catalog edits, metafields at scale, workflows shaped around how a team actually works. Remix, Polaris, Admin GraphQL API.

AWS integrations & pipelines

The serverless backends behind the storefront: catalog sync from external systems, queue-based processing, verification after every write. Lambda, DynamoDB, SQS, S3.

Application review

For an application you inherited, or had built fast with AI tooling: I reconstruct what it actually does, test its critical paths, and hand back a risk-ranked list of what to fix first.

GOOD FITS

- A Shopify workflow no off-the-shelf app handles safely.
- An integration that keeps failing, or one nobody wants to touch.
- An agency with client work that runs past its backend or AWS depth.
- Catalog or operational data that has to stay correct at scale.
- An application built quickly — by AI tooling or a contractor — that isn't yet trusted in production.

SHIPPED SYSTEMS

AWS DATA PIPELINE

Bulk product create & update

A serverless system that syncs product catalogs from external sources into Shopify — creating, updating, and verifying products as a continuous pipeline rather than a one-time import.

> IN PRODUCTION Runs daily in production. First put to work on a live catalog of over 500,000 products.

Built with Lambda (import → create/update → verification) coordinated by SQS · DynamoDB product store and SKU index · S3 · Secrets Manager · dead-letter queue · CloudWatch alarms on queue depth, errors, and DLQ activity · Serverless Framework v4 · Node.js, Python · Shopify Admin API

AWS DATA PIPELINE

Shopify taxonomy & category mapper

Classifies any store's catalog against Shopify's Standard Product Taxonomy — deterministic rules first, AI only where the rules fall short, with human review before anything is written back. Store-agnostic: a new client is a new config.

> IN PRODUCTION Run once across a live catalog of over 500,000 products. New products are now classified in the daily import.

Built with Python engine package with a thin CLI wrapper · two-stage classifier (rules + Claude AI) · high-confidence / flagged-for-review split, no silent writes · dry-run and stage-by-stage modes · per-run audit report including AI cost estimate · Shopify Admin API write-back

AWS DATA PIPELINE

Taxonomy import enricher

A standalone service inside a daily ERP-to-Shopify pipeline. It classifies each incoming product against a reviewed rule set and stamps the batch before the importer sees it — so categorization happens automatically, and the importer is unchanged. If classification can't complete, the batch passes through untouched rather than holding up a time-sensitive run.

> IN PRODUCTION Runs in production with every daily import.

Built with event-driven Lambda on object creation · Python · Serverless Framework · version-pinned reuse of the classification engine with recorded provenance · tests run against the real rule set and feed output into the importer's own acceptance check · structured JSON logging with alarms on errors, DLQ, and the pass-through path · reversible cutover with single-command rollback

EMBEDDED SHOPIFY ADMIN APPLICATION

Collection Price Updater

Reprices every product in a collection at once, with an exact before/after preview using the same price math that writes, and one-click rollback from saved snapshots.

Built with Remix + Polaris · SST v4, Lambda behind CloudFront (scale-to-zero) · DynamoDB history · S3 before/after snapshots · Shopify Admin GraphQL API · TypeScript strict · chunked updates for small sets, Shopify async bulk operations at 200+ products

EMBEDDED SHOPIFY ADMIN APPLICATION

Metafield Manager

Applies a consistent set of product metafields across a catalog — from a single product to thousands — through saved field-set views, a five-step bulk wizard, and a full audit log of who changed what and when.

Built with Remix + Vite on Polaris v12 · hand-rolled fetch-based Shopify GraphQL Admin client, no SDK dependency · DynamoDB for field sets and audit log · Lambda behind CloudFront via SST · Shopify Admin API 2026-04

STACK

Commerce	Shopify App Development · Shopify Plus · Admin GraphQL API · Polaris · Catalog Management · PIM · E-commerce
AWS & backend	Lambda · DynamoDB · SQS · S3 · CloudWatch · CloudFront · Serverless Framework · SST · Data pipelines / ETL · Distributed systems
Languages	TypeScript · JavaScript · Node.js · Python · GraphQL · SQL · Remix · React
Practice	Software architecture · Systems integration · Test automation · Code review · Requirements analysis · Technical leadership

HOW I WORK

Delivery is gated, and the gates leave a record. The numbering below is the order the work actually moves through.

01 **Requirements become testable acceptance criteria**
"Prices update correctly" becomes a specific, checkable statement about what happens to which products under which conditions — before anything is built.

02 **Important work is reviewed in a fresh context**
Reviewed again from a clean slate, without the assumptions that built it.

03 **Tests and live verification are recorded**
The system is exercised against real conditions, and those runs are recorded — so "it was verified" points at a result rather than a claim.

04 **Handoff includes a quality record you can read**
What was built, what was tested, what was verified, and what is known to be outstanding — in plain language, and it travels with the system.

WORKING TOGETHER

Remote, from Boise, Idaho (Mountain Time). The most useful first message describes the workflow to improve and what it would be worth to get right — rough is fine.

I reply, usually within a business day. We spend 25–30 minutes on the situation: what the system does today, what goes wrong, and what it costs when it does. If it's a fit, a written scope follows — the outcome, what's included, what isn't, the schedule, and the price. If it isn't, I'll say so and point elsewhere where I can.

The two embedded Admin applications are live and in use. I demo them on a call, against the kind of catalog you actually have, rather than leaving a general-purpose tool open on the web.

don@donstefani.com · donstefani.com