

PARTNER CASE STUDY | RAPIDCANVAS + DATABRICKS

From Delta Tables To Governed Agents And Jobs

How RapidCanvas ships Databricks ingestion as versioned Asset Bundles, keeps Unity Catalog in charge, and reads Delta tables through Databricks SQL for agents and applications.

2
use cases

0
in production

September 2026
edition

Real AI transformation, accelerated.



AT A GLANCE

RapidCanvas + Databricks

Where a customer runs its lakehouse on Databricks, RapidCanvas works with it rather than beside it. Our Hybrid Approach™, an agentic platform paired with expert-in-the-loop delivery, packages ingestion as Databricks Asset Bundles versioned with the rest of the solution, keeps Unity Catalog as the governance boundary, and reads Delta tables through Databricks SQL to feed agents and applications.

The two stories here show what that looks like today. A regulatory ingestion pipeline runs as a scheduled Databricks job, deployed from the solution repository, cleaning and cross-referencing records inside Unity Catalog. Separately, a sales-agent pattern shows RapidCanvas agents calling the Databricks SQL Statement Execution API and Model Serving endpoints for customer context and lead scoring; it runs against a local stand-in, ready to point at a live workspace once credentials arrive.

2

Databricks use cases documented

3

Medallion layers: bronze, silver, gold Delta tables

4

Databricks features used: bundles, catalog, SQL, Model Serving

PARTNERSHIP

RapidCanvas works directly inside customers' Databricks workspaces today; there is no Databricks partner listing or formal partnership in place.

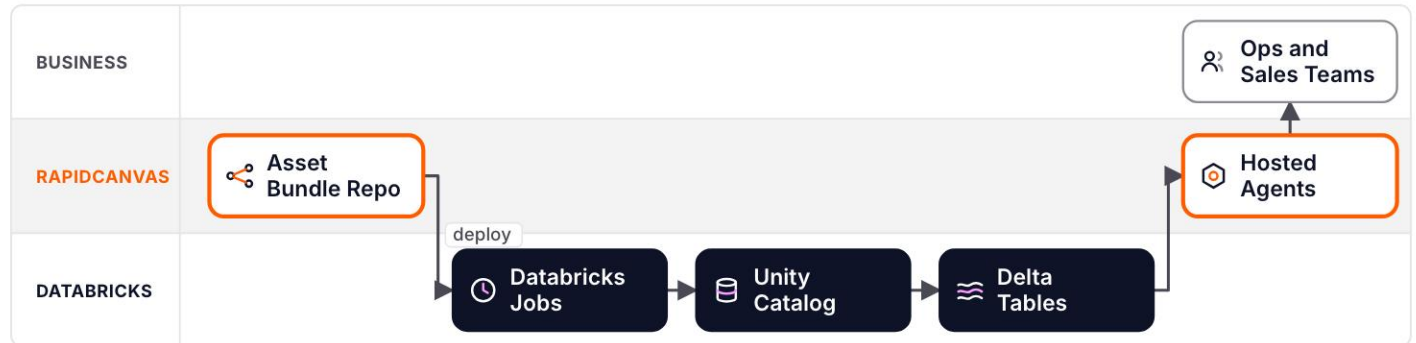
When RapidCanvas Is a Fit

- 1 The lakehouse runs on Databricks and the customer wants agents and applications, not another notebook.
- 2 Ingestion needs to ship as versioned, reviewed code, not a hand-built job in the workspace.
- 3 Unity Catalog governance must stay intact while RapidCanvas adds the delivery and application layer.
- 4 Models already run on Databricks Model Serving and should power agents and Data Apps.

REFERENCE ARCHITECTURE

How RapidCanvas Builds on Databricks

RapidCanvas ships ingestion as a Databricks Asset Bundle, keeps Unity Catalog as the governance boundary, and reads Delta tables through Databricks SQL for agents and applications.



Code, Not Manual Jobs

Ingestion notebooks and jobs are defined in a Databricks Asset Bundle and deployed from the solution repository with the Databricks CLI, reviewed like any other code.

Unity Catalog Stays Governance

Catalogs, schemas and volumes remain the landing zone and the permission boundary; RapidCanvas queries respect the grants already in place.

Agents Read, Not Replace

RapidCanvas services call Databricks SQL and Model Serving endpoints to bring gold Delta tables and models into agents and Data Apps.

Use Cases in This Document

NO.	USE CASE	INDUSTRY	STATUS
01	A Regulatory Registry Pipeline Shipped As Code	Telecommunications, Latin America	In delivery
02	An Agent Pattern Ready For Live Databricks Data	Telecommunications, Latin America	Reference pattern

USE CASE 01 | TELECOMMUNICATIONS, LATIN AMERICA

IN DELIVERY

A Regulatory Registry Pipeline Shipped As Code

Databricks Asset Bundles

Databricks Jobs

Unity Catalog

Delta tables

Scheduled jobs

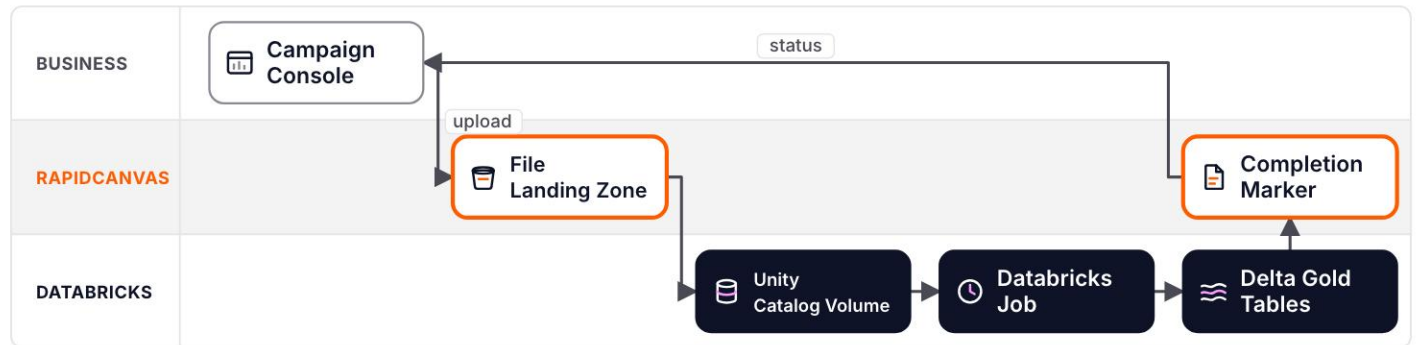
Expert-in-the-Loop

The Challenge

A telecommunications operator must exclude every phone number on a national do-not-call registry from outbound campaigns. The registry is republished irregularly as dozens of spreadsheet files with no deltas and no version marker, and lists must be cross-checked against a large subscriber base. The customer's lakehouse is Databricks, so the pipeline had to live there under Unity Catalog.

Solution Implemented

RapidCanvas built a bronze, silver, gold pipeline deployed as a Databricks Asset Bundle with development and production targets. A file-arrival trigger starts the job once a publication lands in a Unity Catalog volume. Bronze parses each file on serverless compute, skipping anything processed via a hash-based control table. Silver collapses the registry to one row per phone under an explicit rule. Gold crosses the registry with the operator's subscriber tables to produce campaign-ready lists.



How It Works

- 1 An uploaded registry file lands in a Unity Catalog volume, triggering a file-arrival job start.
- 2 Bronze parses each file on serverless compute, skipping anything a hash check flags as processed.
- 3 Silver collapses the registry to one row per phone, logging every rejection with a reason.
- 4 Gold crosses the clean registry with the operator's subscriber tables to build campaign-ready lists.

Results and Benefits

- The pipeline processes more than seven million registry rows into one row per phone.
- Rejected rows land in a separate table with a reason, instead of silently disappearing.
- The prior hand-built job still runs in production; deploying this Asset Bundle there is scheduled next.

HOW IT RUNS ON DATABRICKS

The pipeline runs on Databricks Jobs and serverless compute inside the customer's own workspace, with Unity Catalog governing every table.

USE CASE 02 | TELECOMMUNICATIONS, LATIN AMERICA

● REFERENCE PATTERN

An Agent Pattern Ready For Live Databricks Data

Databricks SQL Statement Execution API

Databricks Model Serving

AI Agents

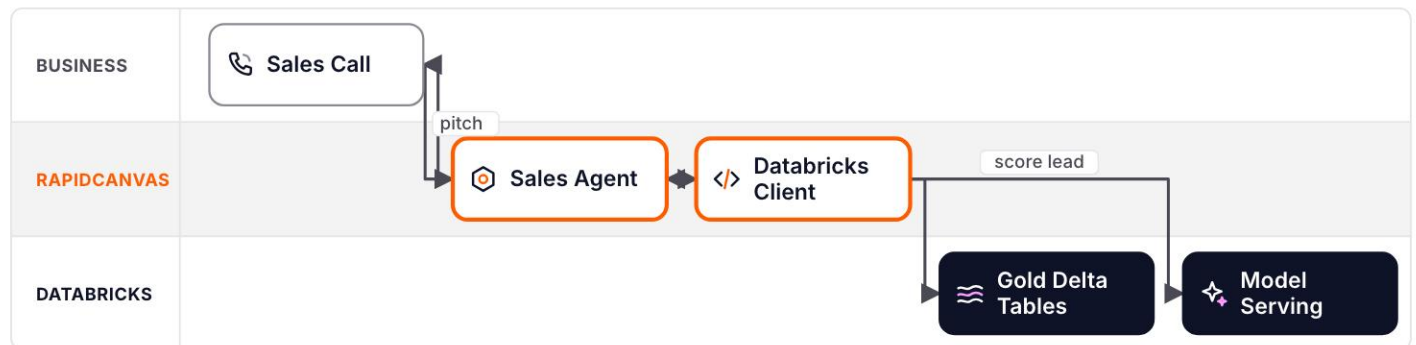
Connectors

The Challenge

Inbound and outbound sales agents need customer purchase context and a lead-propensity score before they speak. In the target architecture that context lives in gold Delta tables and a model served from Databricks. RapidCanvas had to build and demonstrate the agent platform before workspace credentials existed, without hard-wiring vendor specifics into a growing family of agents.

Solution Implemented

RapidCanvas built a shared core library giving every agent one model client, session memory, guardrails and a single Databricks client, so a later credential change touches one place. That client posts a SQL statement to the Statement Execution API for purchase history from a gold table, and posts records to a Model Serving endpoint to score lead propensity. A hosted model composes the pitch from that context. Today both endpoints are served by a local stand-in with precomputed data.



How It Works

- 1 A shared core library gives every planned agent one model client and one Databricks client.
- 2 The client posts a SQL statement to fetch purchase history from a gold Delta table.
- 3 The same client posts records to a Model Serving endpoint to score lead propensity.
- 4 A hosted model composes the pitch from the returned purchase history and propensity score.

Results and Benefits

- One integration surface now serves Databricks reads and scoring calls for every planned agent.
- Moving from demonstration to production is a configuration change, not a rewrite of the agents.
- This pattern is not yet a customer deployment; it runs against a local stand-in today.

HOW IT RUNS ON DATABRICKS

Once pointed at a live workspace, each agent call becomes a Statement Execution API request and a Model Serving invocation, driving Databricks compute while data stays in the customer's workspace.

WORKING TOGETHER

RapidCanvas on Databricks

Every engagement runs on the Hybrid Approach™: the RapidCanvas agentic AI platform builds the pipelines, agents and Data Apps, and RapidCanvas experts work alongside the customer's team on data modeling, validation and the last mile into production.

RAPIDCANVAS BENCHMARKS ACROSS ENGAGEMENTS

4-8 weeks

to first production value

10x

faster time to value than a traditional custom build

Up to 80%

lower cost than a traditional custom build

Ship Code, Not Manual Jobs

Databricks Asset Bundles live in the customer's solution repository and deploy through the Databricks CLI, with separate development and production targets reviewed like any other code change.

Read The Lakehouse In Place

In the reference pattern, RapidCanvas services call the Databricks SQL Statement Execution API on a SQL warehouse, so Unity Catalog permissions govern every query. A native connector is not yet in the platform.

Build On Delta Layers

The delivered pipeline lands raw files in a Unity Catalog volume and builds bronze, silver and gold Delta tables, so downstream teams query governed tables instead of raw drops.

Run In Any Cloud

RapidCanvas runs as SaaS or privately on AWS, Azure or Google Cloud, sitting alongside the customer's Databricks workspace rather than requiring a migration.

Databricks Services in These Stories

Databricks Asset Bundles

Packages and deploys ingestion jobs with the solution

Databricks Jobs

Orchestrates bronze, silver and gold tasks in order

Databricks file-arrival triggers

Starts the job once new files land in a volume

Unity Catalog

Governs catalogs, schemas and volumes as the landing zone

Unity Catalog external locations

Connects the job to cloud storage without embedding credentials

Delta tables

Stores bronze, silver and gold data across the pipeline

Databricks SQL Statement Execution API

Lets hosted agents query gold Delta tables directly

Databricks Model Serving

Scores lead propensity for the sales-agent pattern

Databricks CLI

Deploys Asset Bundles from the solution repository

Customer names are withheld. Each story describes a real RapidCanvas deployment, verified against the delivered code. Delivery status is shown as of September 2026.

NEXT STEP

See How RapidCanvas Works on Databricks

See how RapidCanvas works with Databricks, or book a discovery call to assess fit for your lakehouse.

[Book a Discovery Call](#)

rapidcanvas.ai

Real AI transformation, accelerated.

