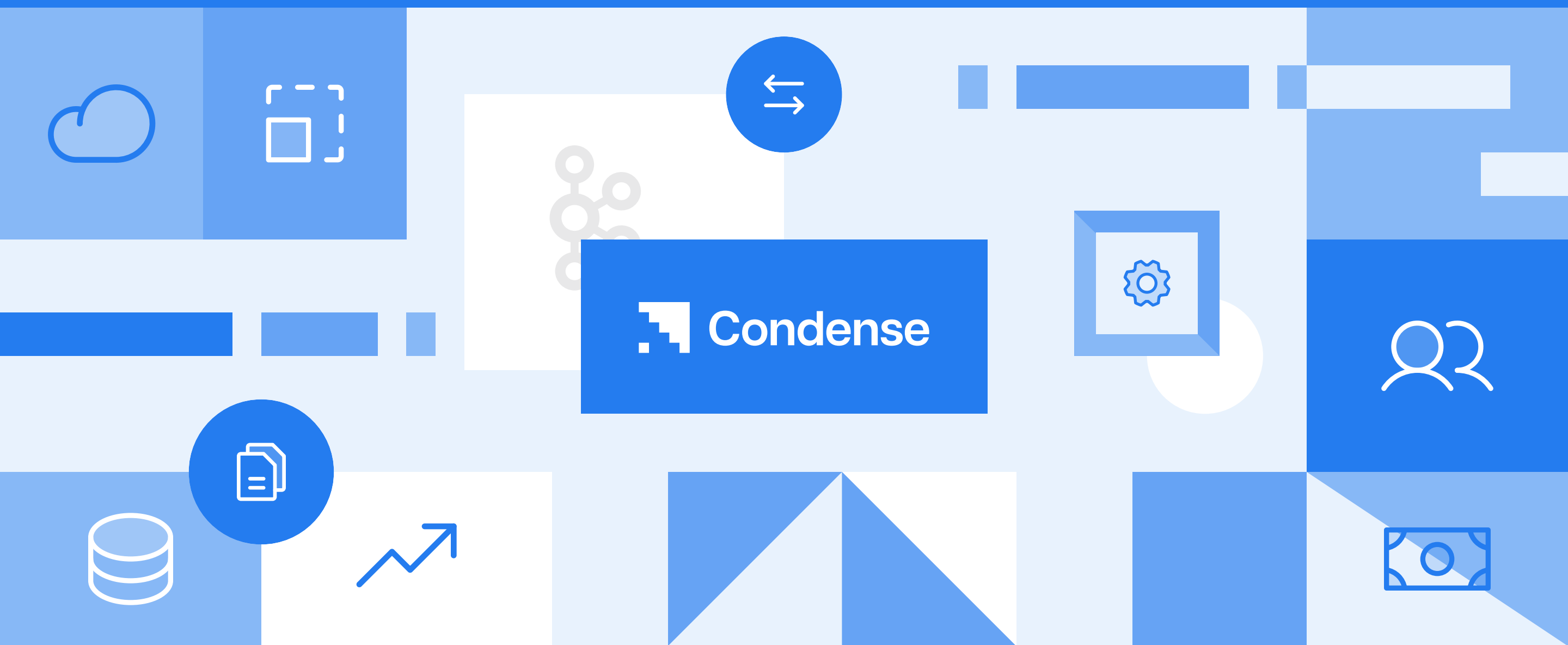


 Comparison

Data in Motion, Simplified at Scale

Evaluating Condense vs Apache Kafka
for Enterprise Real-Time Data Streaming



 **Condense**

ZELIOT

Presented By :
Zeliot, for their platform [Condense](#): a Kafka- native, fully
managed streaming platform deployed in your cloud

 |  |  |  | 

All Rights Reserved 2026 @ Zeliot

How Does **Condense** Compare vs Kafka?

Condense and Apache Kafka (OSS) represent the choice between deploying a finished application and managing raw infrastructure. Apache Kafka OSS is a distributed engine that moves bytes reliably, but it requires you to build, scale, and maintain the entire ecosystem including separate clusters for any data processing. **Condense is a unified streaming platform** that wraps the Kafka engine with an autonomous execution layer, allowing you to run business logic directly on the stream without managing the underlying "plumbing."

> Architecture

FEATURE	CONDENSE	APACHE KAFKA
System Design	A unified environment that bundles Kafka, Kubernetes, and the App Runtime into one integrated stack	Distributed design based on a Java/Scala engine that utilizes the JVM, OS page cache, and internal KRaft for cluster metadata
Infrastructure	Managed BYOC: Deploys a fully managed stack (Kafka + K8s + Logic) into your VPC with one click from cloud marketplace of AWS, Azure and GCP	DIY Infrastructure: You must manually provision VMs, networking, storage, and KRaft
Unified Platform	Unified Application Fabric: Merges the Kafka engine with a native event-driven microservice runtime. It is an "Execution Fabric"	Fragmented Infrastructure: Operates as a standalone distributed log. Requires external, manual integration with separate processing layers (Flink, Spark, or custom JVM apps)
Logic Execution	Managed App Runtime: Custom transforms run in a managed, serverless-style environment. Condense handles containerization, state management, and orchestration	External Services: Requires separate Flink, Spark, or Microservices to process data
Scaling Scope	Full-Stack: Automatically scales brokers and custom code in lockstep based on real-time lag	Infrastructure-Only: Manual broker provisioning and time-consuming partition rebalancing
Storage	Native Object Storage: Direct offloading to S3/ GCS/Azure Blob; data stays in your buckets for infinite retention	Local Disk (Default): Requires manual configuration of Tiered Storage and complex cleanup policies
Performance	Serverless Scale and Zero-Hop: Running GBps+ enterprise data workload and autonomous scale for spikes and variations. Application logic runs "local" to the broker, eliminating the network latency of external processors	Network-Heavy: Every transformation usually requires a network hop to an external processor

> Ecosystem

FEATURE	CONDENSE	APACHE KAFKA
Operational Effort	Zero-Ops, platform handles upgrades, patching, and self-healing autonomously	A cloud-native tiered architecture that uses the proprietary Kora engine to disaggregate compute from storage for elastic, multi-tenant scaling
Industry Readiness	Dual-Layer Ecosystem: Access a vast library of generic connectors alongside industry specific domain-aware transforms for Mobility, IoT, and FinTech (e.g., VIN parsing, Geofencing) out of the box	Standard Connect: Generic connectors that require manual configuration and worker management
Deployment Model	BYOC-First: Runs 100% in your private VPC to keep data sovereign and eliminate egress fees	Self-Managed: Runs anywhere, but you own the complexity of the security and network stack
Dev Experience	Integrated AI IDE: Build, test, and deploy Java or Python logic natively with built-in AI for debugging and code generation	Command Line / External: No native UI; requires external tools like Kafka-UI or AKHQ
Observability	Full-Stack: Single dashboard tracing an event from raw ingestion through logic to the sink	Fragmented: Requires stitching together Prometheus, Grafana, and external APM tools

> Support & Compliance

FEATURE	CONDENSE	APACHE KAFKA
Access Experts	Direct Engineering: Built in support widget to raise ticket. Condense offers support priority based TAT for resolution	Community / Consultants: Reliance on StackOverflow or expensive third-party support contracts
Security	Own Cloud -Sovereign: Inherits your IAM and security groups; 100% data localization	You must build and maintain security policies, certificates, and auditing
SLAs	99.95% Full-Stack: Covers the brokers, the processing logic, and the connectors under one SLA	None: Uptime is entirely dependent on your internal DevOps team's expertise
TCO	Predictable BYOC: Pay for compute (vCPU/hr) inside your VPC. No throughput or egress fees; uses your existing cloud credits (EDP)	Kafka is free, but engineering hours for operations are not

Why Switch to Condense?

UNIFIED PLATFORM VS INFRA SPRAWL

OSS Kafka is a "broker-only" technology. To build a real-world application, you must manually integrate and manage separate clusters for Kafka Connect, Schema Registry, Flink/Spark (for logic), and Prometheus/Grafana (for monitoring)

- 💡 Condense collapses these silos into a single Unified Application Fabric. It merges the Kafka engine with a native, event-driven microservice runtime. You don't "glue" tools together; the tools are built into the fabric

"ZERO-HOP" PERFORMANCE

In OSS Kafka processing typically happens in external microservices or Flink clusters. Data must travel from the broker over the network to the processor and often back again, adding significant network latency and egress costs

- 💡 Condense allows you to deploy your Java, Python, or Go logic local to the data. By running code on the same substrate as the broker, Condense eliminates the network "hop," delivering sub-millisecond end-to-end latency for mission-critical alerts

MANAGED BYOC (SOVEREIGNTY W/O THE OPS)

In OSS Kafka you own the code, but you also own the 3 AM "broker down" phone calls. Scaling, patching, and rebalancing are manual, high-risk operations

- 💡 Condense offers a Bring Your Own Cloud (BYOC) model. It deploys directly into your AWS/GCP/Azure account via the marketplace leveraging your cloud credits and IAM, but with fully managed platform without headache of operations and maintenance

VERTICAL INTELLIGENCE (THE MOBILITY / IOT ADVANTAGE)

OSS Kafka is a generic "dumb pipe." It has no idea what a VIN (Vehicle Identification Number), a geofence, or a telematics packet is. You have to write every line of that domain logic from scratch

- 💡 Condense ships with Vertical Blueprints. It includes pre-built, domain-aware modules for Mobility (Trip Formation, Driver Scoring) and IoT (VSS Mapping). This allows teams to go live in days rather than the months required to build these patterns on OSS





TCO REDUCTION

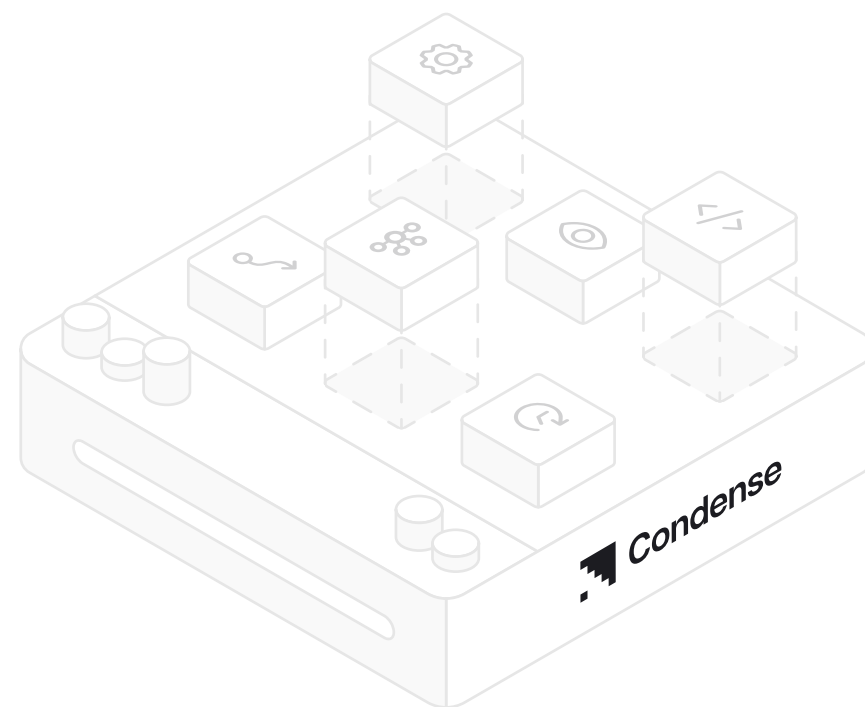
In OSS Kafka the license is "free," the Total Cost of Ownership is massive. You need a dedicated team of 3-5 specialized SREs to manage a production-grade Kafka ecosystem

- 💡 Condense reduces TCO by up to 60%. It eliminates the need for a specialized Kafka Ops team and uses a flat, predictable vCPU-based pricing model. There are no "hidden taxes" for throughput, partitions, or data egress within your VPC

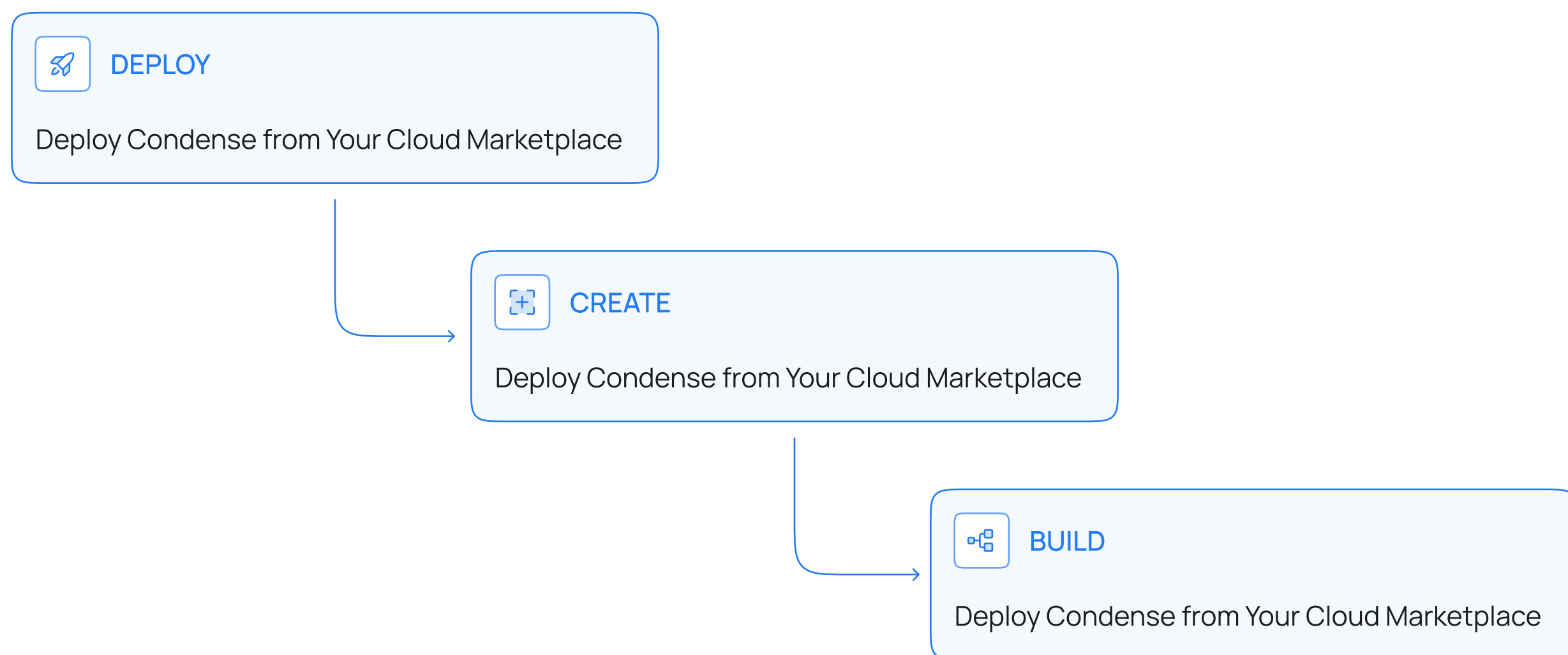
Vertical Data Platform for Real-Time Businesses

Condense understands vertical data by design and connects directly to real time data sources like vehicles or GDS platforms. Its purpose-built industry specific connectors and transformations accelerate the development of real-time vertical use case

-  Accelerate the path from idea to production
-  Rapidly realize and deploy industry use cases
-  Enterprise scale with native governance
-  TCO ↓ through vertical industry-first approach



Get Started with Condense in 3 Simple Steps



Most platforms focus on streaming data, Condense focuses on understanding it. It's not just managed Kafka; it's Kafka with context. Infused with domain intelligence, **Condense transforms raw streams into decisions, automations, and impact**, because the right Kafka platform should understand what it's moving

Hope this analysis sparked a new vision for your data architecture!

Stick Around, It Gets Better



[Read Our Blogs](#) →



[Read Our Documentation](#) →



[Read Our eBooks](#) →

We're Ready When
You Are. Let's Talk.

[Schedule a Meeting](#)

Don't Miss a
Thing - Come Along

