Free Guide from Forbytes: Comparison of Data Ingestion Methods

Choosing the right data ingestion method can make all the difference for your analytics, reporting, and business intelligence. Each tool comes with its own strengths, file size limits, delivery modes, and ideal use cases. Knowing which one fits your pipeline saves time, reduces errors, and ensures reliable data flow.

That's why we created this guide to help you compare methods, weigh their pros and cons, and pick the most suitable option for your business needs.

Apache Spark Connector

Strengths: Supports all Spark-compatible formats; unlimited file size; ideal for secure streaming pipelines and preprocessing on Spark.

Weaknesses: Requires Spark cluster; cost considerations; batch vs. streaming setup can be complex.

Use Cases: Existing pipelines, fast Spark-based streaming, high-volume transformations.

Azure Data Factory (ADF)

Strengths: Supports many formats including Excel and XML; handles large files; queued or triggered ingestion.

Weaknesses: Slower ingestion; loads data into memory before processing.

Use Cases: Copying large files from 90+ sources; cloud or on-prem migration.

Event Grid

Strengths: Continuous ingestion; triggers on blob creation or renaming; supports multiple formats.

Weaknesses: Max 6 GB uncompressed; limited to queued ingestion.

Use Cases: Continuous Azure storage ingestion; event-driven pipelines.

Event Hub

Strengths: Supports real-time messages and events; queued or streaming ingestion.

Weaknesses: Requires proper architecture to handle streaming scale.

Use Cases: Real-time event processing, message pipelines.

Get data experience

Strengths: Supports CSV/JSON; direct or queued ingestion; flexible bulk and historical ingestion.

Weaknesses: Limited to 1 GB uncompressed per blob; best suited for smaller datasets.

Use Cases: One-off ingestion, table schema creation, bulk ingestion of up to 5,000 blobs.

IoT Hub

Strengths: Optimized for IoT messages, events, and properties; queued and streaming ingestion.

Weaknesses: Requires IoT-specific architecture; limited flexibility for non-IoT sources.

Use Cases: IoT sensor data, device telemetry, event collection.

Kafka Connector

Strengths: High-volume support; multiple formats (Avro, JSON, CSV, Parquet, ORC); queued or streaming ingestion.
Weaknesses: Java restrictions; setup complexity with multiple producers/consumers.
Use Cases: High-volume pipelines, distributed systems, real-time streaming.

Kusto Client Libraries

Strengths: Customizable programmatic ingestion; queued, streaming, or direct; cost-efficient storage transactions.
Weaknesses: Requires coding knowledge; limited to 1 GB uncompressed per file.
Use Cases: Custom pipelines, event-driven ingestion, optimized cost pipelines.

LightIngest

Strengths: Bulk and historical ingestion; queued or direct; supports multiple formats.
Weaknesses: Case- and space-sensitive; max 1 GB uncompressed.
Use Cases: Data migration, historical backfills, bulk ingestion.

Logic Apps

Strengths: Automates ingestion pipelines; supports multiple formats; queued ingestion.
Weaknesses: Limited to 1 GB uncompressed; less suited for high-volume streaming.
Use Cases: Workflow automation, scheduled or triggered ingestion.

LogStash

Strengths: Mature open-source tool; handles JSON; unlimited file size (Java restrictions); queued ingestion.
Weaknesses: Requires producer/consumer setup; less intuitive for beginners.
Use Cases: High-volume pipelines, existing open-source environments.

Power Automate

Strengths: Automates ingestion commands as part of workflows; queued ingestion; supports multiple formats.
Weaknesses: Limited to 1 GB uncompressed; best for simpler pipelines.
Use Cases: Workflow automation, repetitive ingestion tasks.

Ready to streamline your data ingestion?

Reach out to Forbytes

and start building a smarter, more reliable data pipeline.