

Optimization Rate

81.6%

335.5 TB



61.74 TB

Archives



Reducing Observability Costs at Scale

How to Optimize Data Ingestion Volumes Without Sacrificing Visibility

Telemetry Data Volumes Are Growing

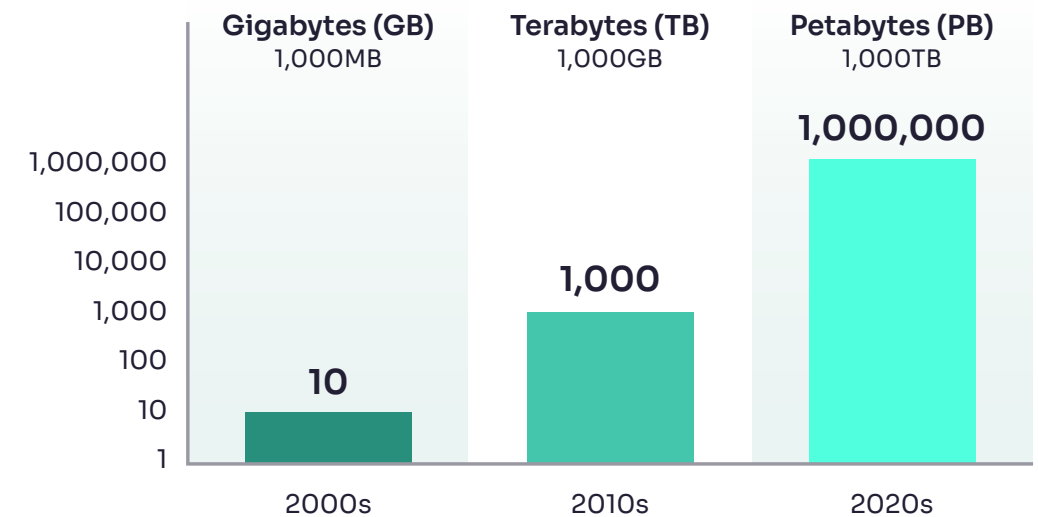
Organizations that leverage distributed and cloud-native architectures are experiencing massive growth of telemetry data. Large enterprises in particular are routinely working with telemetry data at petabyte scale, a significant increase since the turn of the decade.

As data volumes grow, organizations are forced to ship and store more data in their observability platforms — most of which were built in a time when mass data consumption was necessary for effective correlation. However, many of these tools aren't architected to operate efficiently at scale, and costs are skyrocketing as a result.

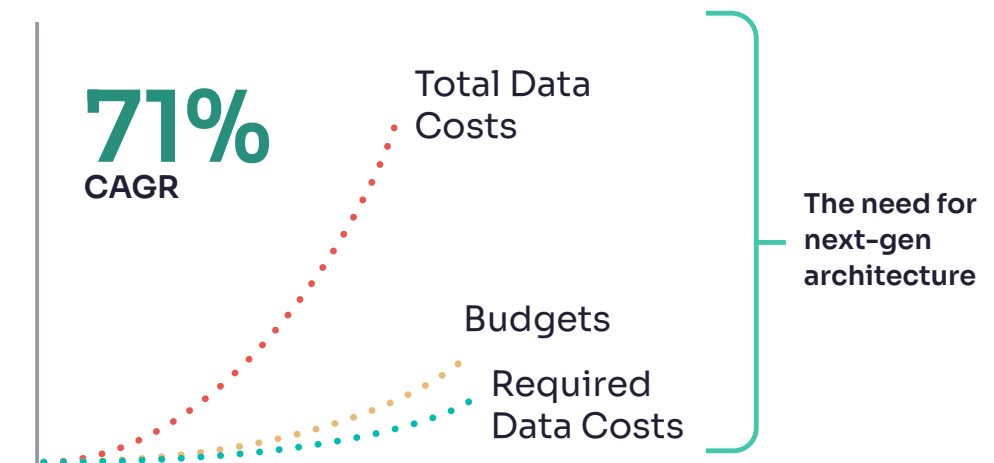
Teams need a fundamentally new strategy for controlling telemetry data volumes — one that enables them to reduce observability costs by optimizing data volumes pre-index, without limiting visibility.

In this guide, we'll explore the benefits of intelligent telemetry pipelines — and show how they empower teams to implement cutting-edge strategies that overcome the limitations of traditional observability platforms.

Increase in Log File Consumption by Large Enterprises
Log Volume Ingested



Source: Gartner, Cool Vendors in Observability and Monitoring for Logging and Containers, Published 27 April 2022. GARTNER is a registered trademark and service mark of Gartner, Inc. and/or its affiliates in the U.S. and internationally and is used herein with permission. All rights reserved.



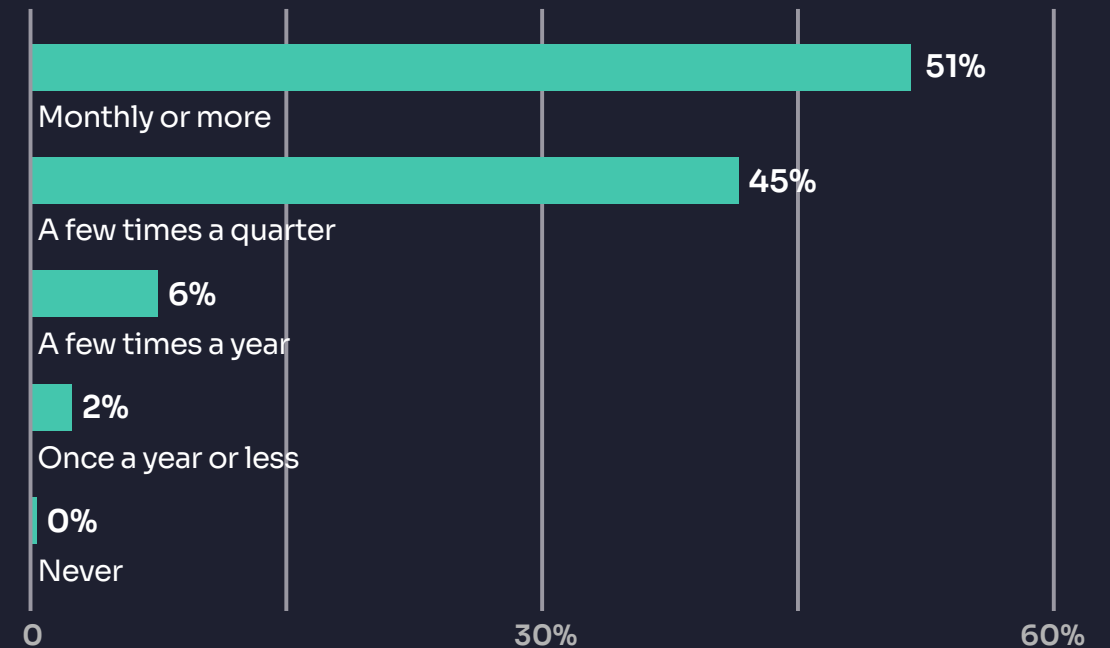
Data costs are growing at a 71% compound annual growth rate (CAGR), far outpacing observability budgets, but the costs associated with the most useful data are rising at a much more manageable pace.

The Core Problem: More Data Than Legacy Observability Platforms Can Handle

Legacy platforms follow a “centralize-then-analyze” telemetry model, where data is shipped directly to a backend system before any processing or analysis begins.

At modern scale, this means queries must process terabytes or even petabytes of unnecessary data during execution, which is both inefficient and incredibly expensive. These platforms charge based on ingestion, indexing, and retention volumes, forcing teams to incur exorbitant baseline costs and overage fees for sending all their data downstream.

Frequency of Overage/Cost Spikes



According to a recent survey, **98% of enterprise companies admit to experiencing overages or unexpected spikes in costs at least a few times a year.**

However, not all telemetry data is created equal.

Broadly speaking, it can be categorized into the following tiers:



High Tier

Critical data required for real-time analysis (e.g., WARN-level logs). Must be routed in full-fidelity to observability platforms.



Middle Tier

Less important data that must remain searchable and accessible, but can be sent to either lower-cost search platforms or to primary observability platforms in a compressed form (e.g., DEBUG-level logs).



Low Tier

Non-critical, low-interaction data (e.g., INFO-level logs). Sent to affordable archival storage in full-fidelity for auditing and compliance purposes.

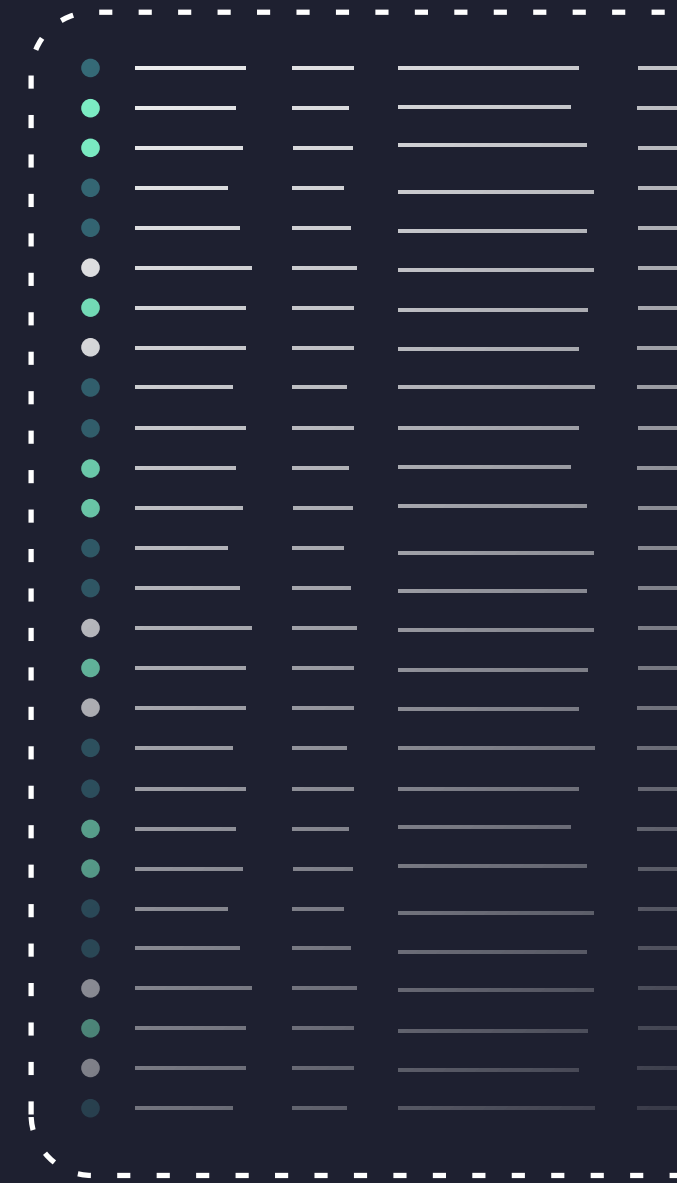
To avoid paying premium rates to index and store less valuable data, teams must filter their telemetry data at the source, accurately identifying which logs, metrics, and traces are worth keeping. The rest should be compressed, dropped, or archived in the appropriate downstream destination. By minimizing noise sent into expensive platforms, teams can more efficiently monitor their environments and investigate issues while significantly reducing costs.

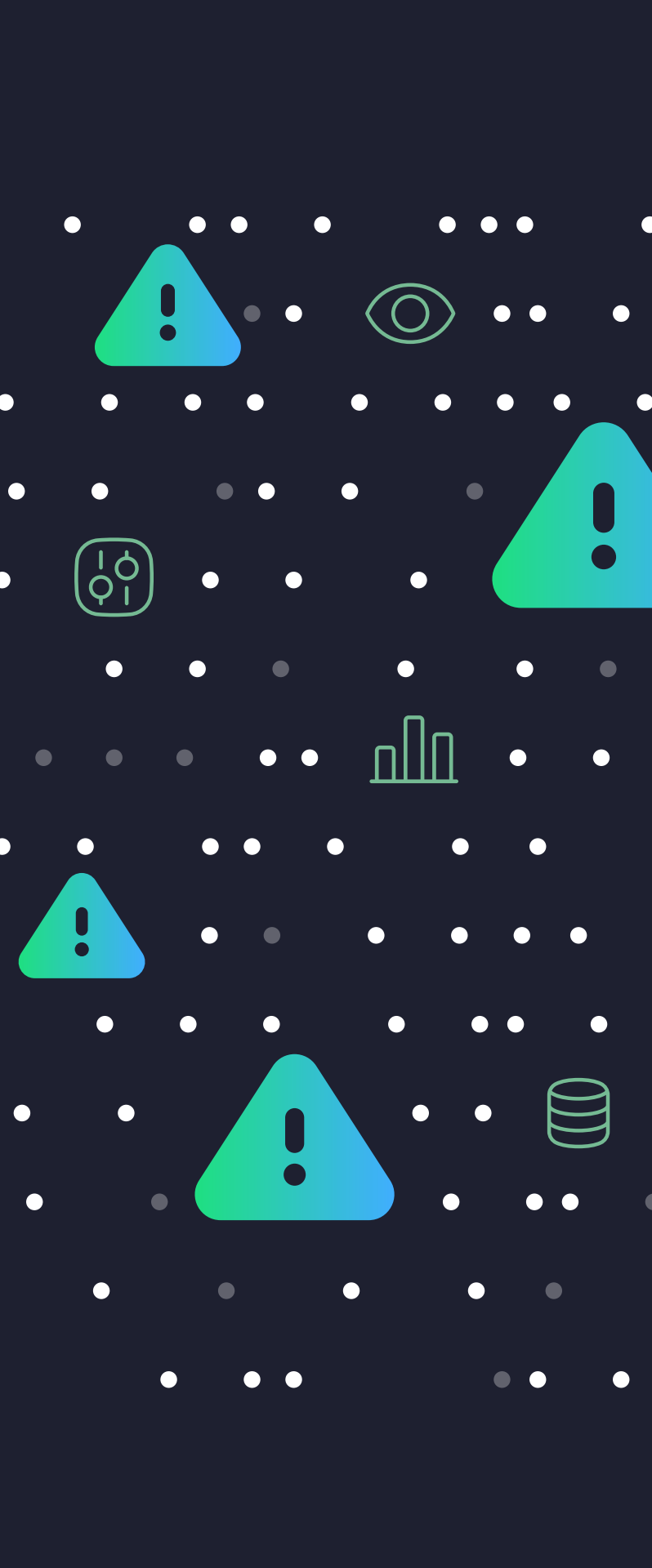
The caveat? Without a purpose-built solution, this approach to data reduction is difficult to implement, requiring a mix of guesswork and time-consuming manual processes that can introduce critical blind spots. We'll explore these challenges in more detail in the following section.

Data Reduction Strategies

There are several strategies teams can leverage to reduce telemetry data volumes:

Dropping Events	100% of data items are discarded.
Sampling Events	1 out of every N data items are delivered, the rest are discarded.
Dynamic Sampling	1 out of every N data items are delivered, where N increases proportionally with data volume. As volume increases, the percentage of dropped data increases, and vice versa.
Suppression	No more than N copies of this data type are delivered per unit of time.
Parsing + Trimming Events	Removing unnecessary, unwanted, or overly verbose parts of an event.
Tail Sampling	1 out of every N trace items are discarded, based on holistic evaluation of spans within the item.





While these methods can be highly effective, they must be implemented with care.

Without a telemetry pipeline solution, teams are forced to stitch together processing workflows using standard log shippers or ingestion agents — neither of which were designed for effective telemetry control at petabyte scale.

This manual approach introduces a number of challenges, including:

- **Blind spots:** These tools provide limited visibility into what's being dropped or transformed, making it difficult to troubleshoot or verify data accuracy.
- **Fragmented control:** Data must be filtered and routed across multiple systems with inconsistent syntax and behavior, making standardization tricky.
- **High overhead:** Maintaining and syncing configs across environments adds operational complexity and drains engineering time.
- **Volume management:** Team leads must institute quotas to ensure data volumes remain under control, and manually roll back changes when data throughput crosses pre-defined thresholds.

It's therefore essential to adopt a solution that provides full control over data processing, routing flexibility, and real-time visibility into data flows pre-index — without significant time or resource investments.

Leveraging Telemetry Pipelines for Efficient Data Management

Telemetry pipelines are transforming observability architectures by providing a fundamentally new way to collect, process, and route telemetry data from source to destination:



Collection: Pipelines ingest data from a variety of sources — including networks, applications, servers, endpoints, and more.



Processing: Next, they apply processing rules to transform, enrich, filter and reduce data.



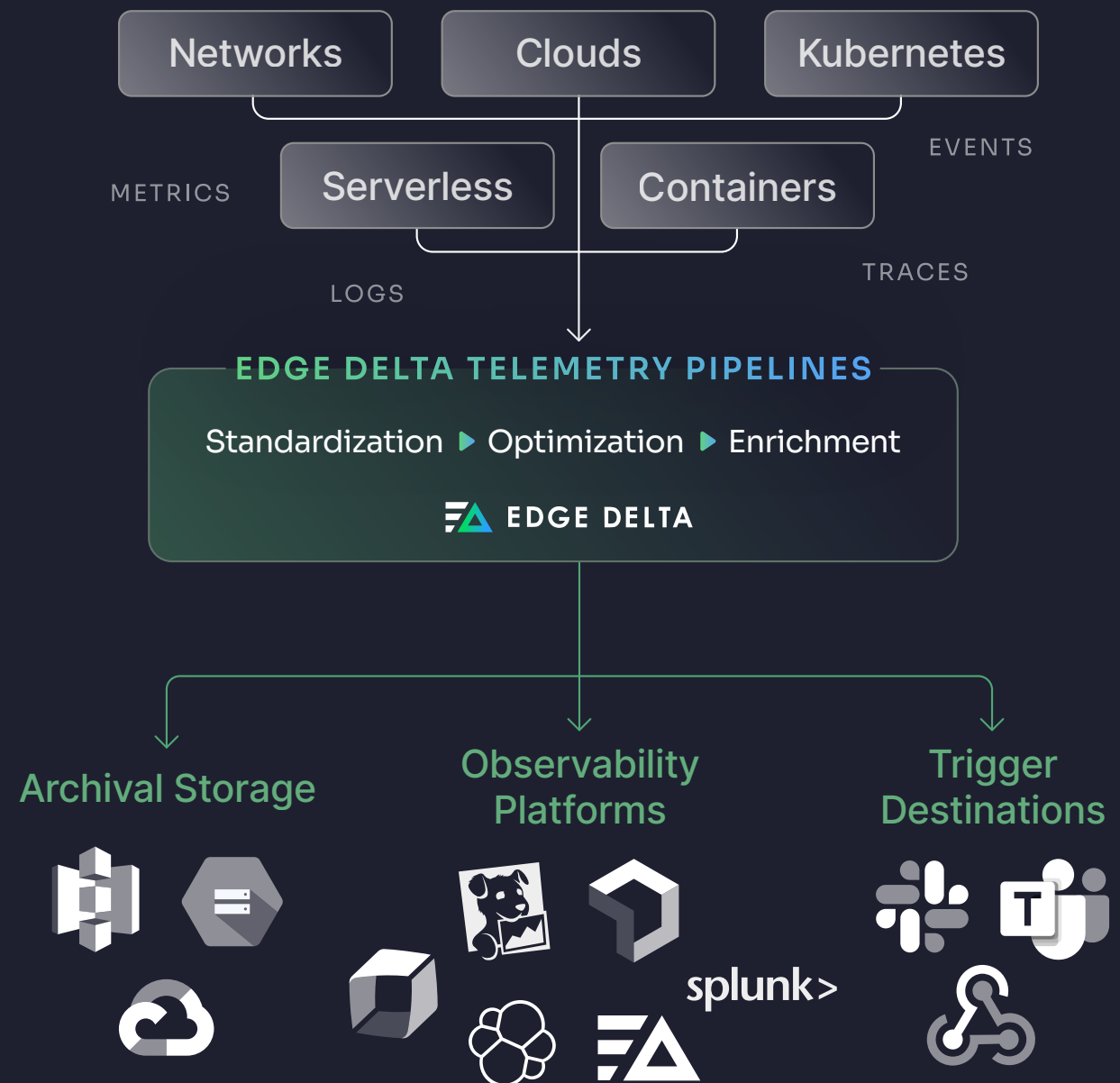
Routing: Finally, pipelines ship data to one or more destinations — including observability backends (Datadog, New Relic, Splunk, Grafana, Edge Delta), alerting tools (Slack, PagerDuty, custom webhooks), and archival storage (S3, Azure Blob, GCP).

This novel approach enables organizations to pair end-to-end data control with full visibility and flexibility. With telemetry pipelines in place, teams can de-risk data optimizations by confirming new processing rules work as intended, which safely and effectively reduces downstream costs.

Manage and Reduce Data Volume with Edge Delta's Next-Generation Telemetry Pipelines

Edge Delta's Telemetry Pipelines are designed to provide maximum data control, visibility, and flexibility at scale.

The architecture is built around lightweight, Go-based agents that process data in real time. With it, teams can reduce, enrich, and route petabytes of telemetry data accurately and efficiently.



With Edge Delta's Telemetry Pipelines, teams gain:

Intelligent Filtering and Sampling:

Edge Delta Pipelines automatically analyze data in motion and provide intelligent processing recommendations, helping users reduce data volume without losing context.

The screenshot displays the Edge Delta Application Logs interface, which is designed for monitoring and processing telemetry data. The interface is divided into three main sections:

- Application Logs:** This section at the top left shows a live tail of logs. It includes a search bar and a list of log entries. The first log entry is an error message: `{ "timestamp": "2025-07-01T17:27:52.522Z", "level": "ERROR", "msg": "...", "resource": { "ed.conf.id": "46aa874e-b2b8-46ff-b998-42af434d1b78", "ed.demo": "demo_template_input_e289", "ed.org.id": "db3e3bbc-7c92-49c8-b17b-0d918a7da1e7", "ed.source.name": "demo_template_input_e289", "ed.source.type": "demo_template_input", "ed.tag": "prod-pipeline-cloud", "host.ip": "10.51.64.194", "host.name": "default-deployment-65fc84c575-dk7mp", "service.name": "demo-demo_template_input_e289" }, "attributes": {} }`.
- Processors:** This central section provides a list of recommended processors for the logs. The recommendations include:
 - Parse JSON:** "We detected unparsed JSON in your logs. For best performance l..."
 - Mask PII:** "We detected possible PII data."
 - Log to Pattern:** (experimental)
- Processed output:** This section on the right shows the result of applying the processors. The first log entry is the same as the original, but it has been processed according to the recommendations.

Real-Time Processing Visibility:

With Live Capture, teams can observe data as it flows through the pipeline and test processing steps locally, gaining clear insight into what's being dropped, transformed, and routed.

The screenshot displays the 'Application Logs' interface, which is designed for real-time monitoring and processing of data. The interface is divided into three main sections:

- Live tail samples:** This section shows a stream of log entries. The top entry is an error log with a timestamp of 10:27:53 AM and a message: `{ "timestamp": "2025-07-01T17:27:52.522Z", "level": "ERROR", "msg": "... }`. Below this, there is a detailed view of the log entry, showing fields like `_type`, `timestamp`, `body`, `resource`, `ed.conf.id`, `ed.demo`, `ed.org.id`, `ed.source.name`, `ed.source.type`, `ed.tag`, `host.ip`, `host.name`, `service.name`, and `attributes`.
- Processors:** This section lists the processing steps applied to the logs. The current processors are: `Parse JSON`, `Mask PII`, and `Update body`, all of which are turned on. There is also an `Add a processor` button and a list of recommendations including `Parse Severity Fields`, `Parse Timestamp`, `Delete Empty Fields`, and `Log to Pattern` (experimental).
- Processed output:** This section shows the result of the processing. The top entry is a log entry with a timestamp of 10:27:53 AM and a message: `Molestiae inventore repudiandae inventore reiciendis.`. Below this, there is a detailed view of the processed log entry, showing fields like `_type`, `timestamp`, `body`, `resource`, `ed.conf.id`, `ed.demo`, `ed.org.id`, `ed.source.name`, `ed.source.type`, `ed.tag`, `host.ip`, `host.name`, `service.name`, and `attributes`. The output shows that the `body` field has been transformed into a string, and the `host.ip` field has been redacted.

Stronger Optimization Techniques:

Edge Delta provides unique processing features like log to pattern conversion, which leverages a proprietary clustering algorithm to automatically detect repeated patterns in log data at the edge.



Reduce Costs by Dynamically Tiering Data with Edge Delta

By intelligently processing data pre-index, Edge Delta users gain the ability to easily implement an important cost-reduction strategy: data tiering.

Instead of constantly sending all telemetry data downstream in full fidelity into a single, expensive platform, teams can dynamically route different portions of data to the most appropriate and cost-effective destination in real time. When an incident occurs, they can use a method called flow control to automatically trigger a change in their pipeline configuration, temporarily routing data in full fidelity into their primary platform until the issue is resolved.

For instance, data can be tiered across downstream destinations as follows:



Baseline Data Flow

High-value, frequently accessed data → Sent to primary observability platform (e.g., Splunk, Datadog)

Less-critical, moderately accessed data → Sent to primary observability platform in compressed form (e.g., Splunk, Datadog)

Compliance and audit logs → Stored in cost-efficient cold storage or archive (e.g., S3, GCP)



Incident Investigation

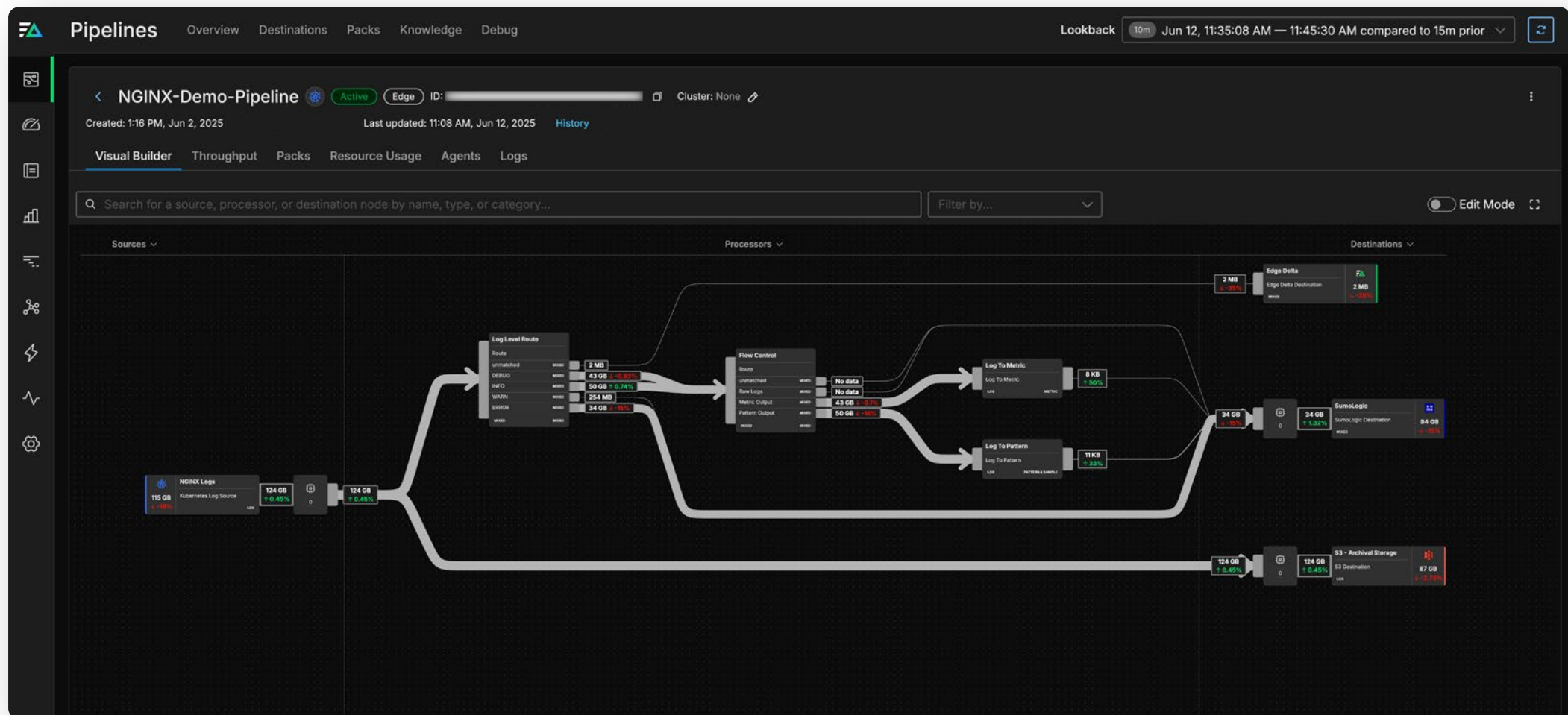
All relevant data (frequently and moderately accessed) → Temporarily sent into observability platform in full fidelity using flow control for real-time context during incident remediation

Rehydrated full-fidelity data → Loaded from archival storage into observability platform on an as-needed basis for post-mortem investigation (e.g., Splunk, Datadog)

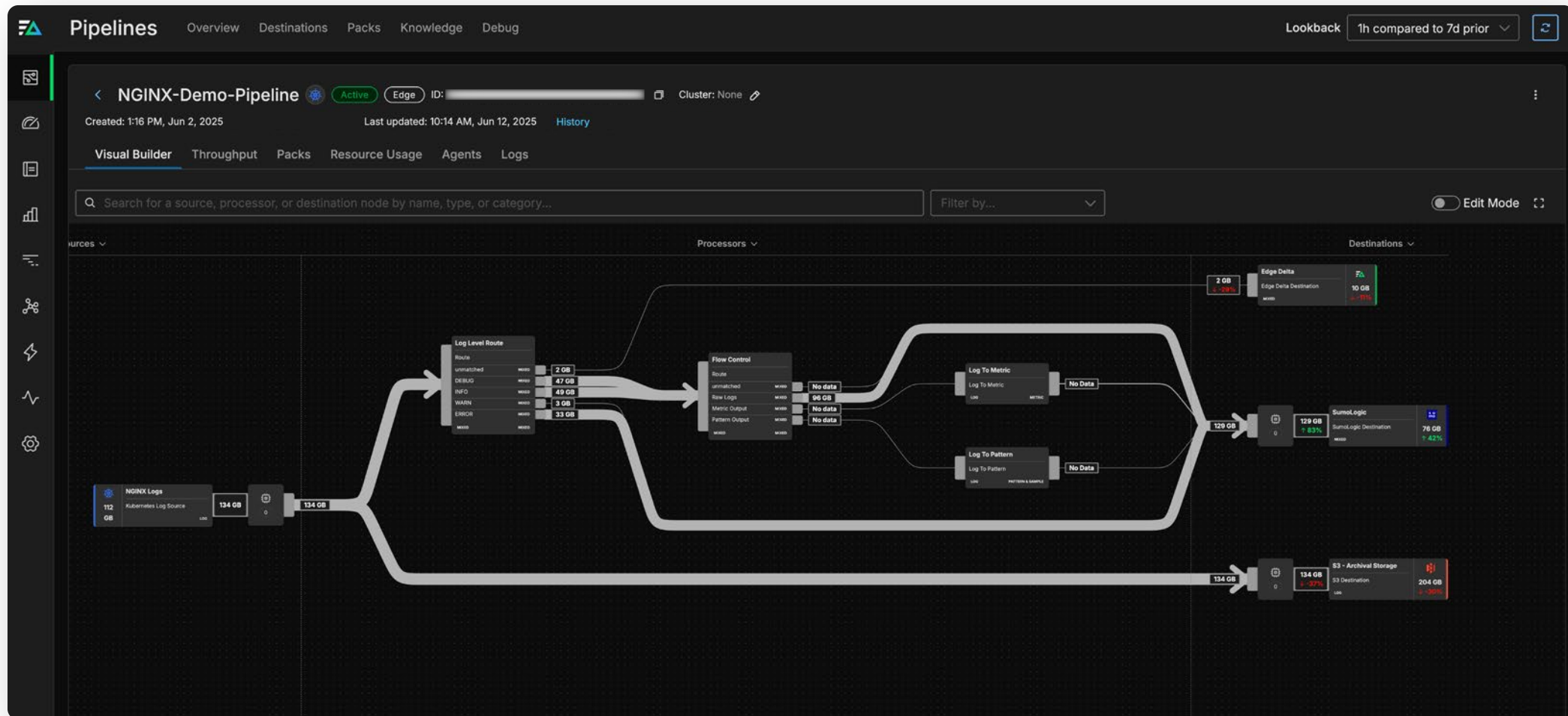
Compliance and audit logs → Remain in cold storage or archive (e.g., S3, GCP)

For example, let's say you're collecting a high volume of NGINX logs, and are struggling to index them all in your SumoLogic instance. With Edge Delta, you can segment, process, and route the data separately based on importance:

1. Route WARN and ERROR messages in full fidelity to SumoLogic for real-time investigation
2. Compress INFO and DEBUG messages into patterns and metrics respectively, and ship them into SumoLogic to feed dashboards and monitors
3. Send a full copy of all raw data into S3 for auditing and compliance



When an incident occurs, Edge Delta Pipelines can dynamically modify data flowing into downstream destinations. In this example, the pipeline temporarily routes INFO and DEBUG logs in full fidelity into SumoLogic to provide full context around the impacted services or component, until the issue is resolved.



If additional information is needed, users can extract telemetry data from their archive and route it directly to their primary observability platform through a process known as rehydration. This rehydrated data is automatically associated with incident windows to provide deeper insight during post-mortem analysis.

With this approach, you retain 100% of your data for troubleshooting and investigations, reduce data noise in your primary and secondary analysis tools, and cut ingestion and retention costs significantly. Edge Delta's vendor-neutral pipelines make this process seamless, offering flexible routing to any destination without adding operational overhead.



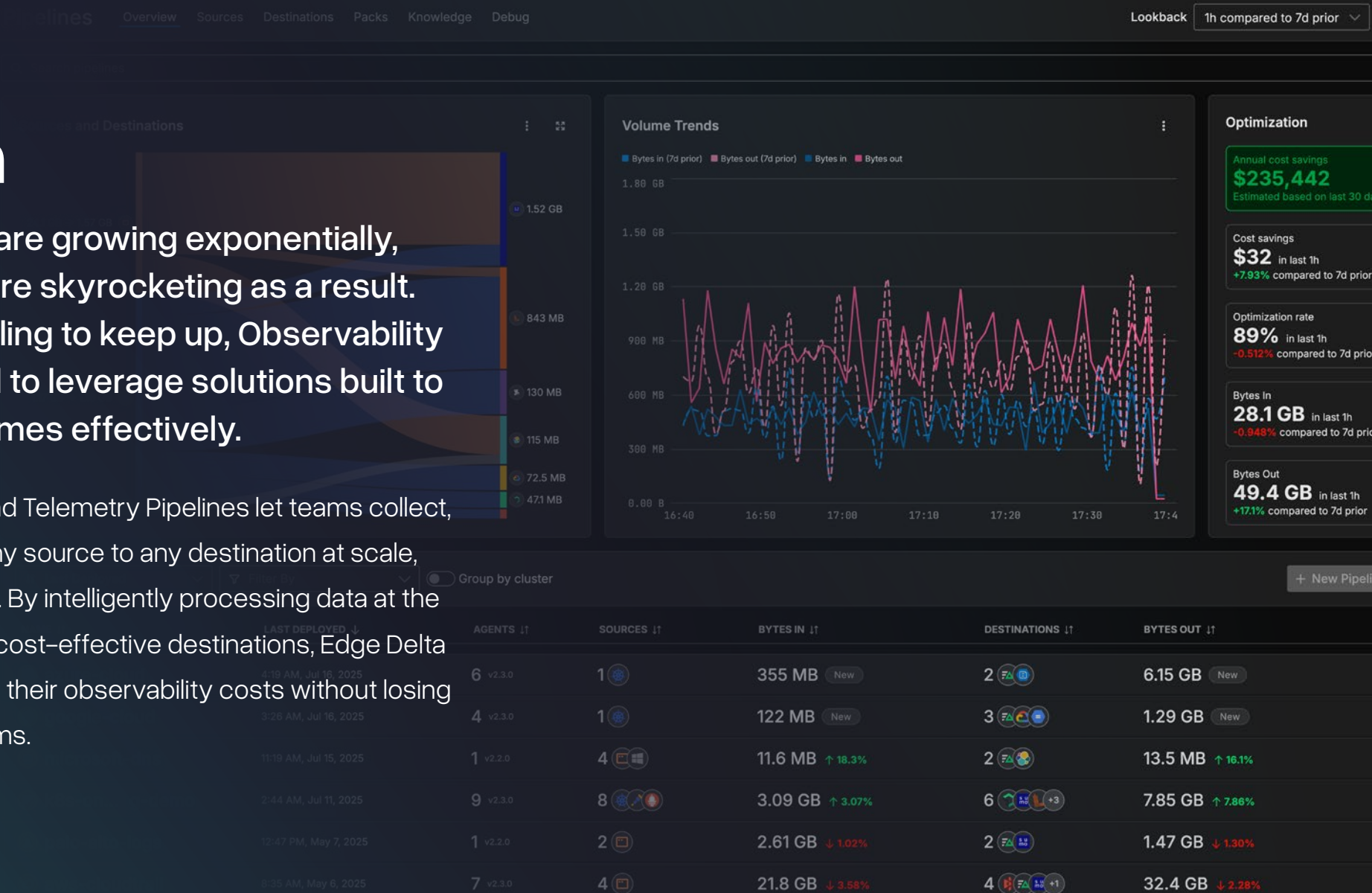
“The older vendors had this thought of, ‘first I’m going to collect everything, and then I’ll be able to do a lot of great things for you.’ And that works, up until the point where you hit scale, where the process of collection becomes its own problem. Edge Delta’s not just about doing what you did in the past and doing it a little bit better — this is about a new way to see this world of collecting and managing observability data.”

Ben Kus, CTO, Box, on reducing log ingestion by 80% with Edge Delta's Telemetry Pipelines

Conclusion

Telemetry data volumes are growing exponentially, and observability costs are skyrocketing as a result. With legacy platforms failing to keep up, Observability and Security teams need to leverage solutions built to handle modern data volumes effectively.

Edge Delta's intelligent, end-to-end Telemetry Pipelines let teams collect, process, and stream data from any source to any destination at scale, and observe it flowing in real time. By intelligently processing data at the source and routing it to the most cost-effective destinations, Edge Delta customers can drastically reduce their observability costs without losing insight or visibility into their systems.





©2025 Edge Delta, Inc. All Rights Reserved.