# Tecton Feature Store Sink I/O for Beam

*Proof of Concept:* *#36078*
*Tracking Beam Issue:* *#35046*
*Last updated:* *Sep 12, 2025*
*Authors:* *Mohamed Awnallah*
*Status:* ***Draft*** *> Current > Needs update > Obsolete*

## Overview

The Tecton Stream Ingest Sink I/O for Apache Beam enables both streaming and batch pipelines to persist features into Tecton's online and offline feature stores. This component seamlessly integrates Tecton's capabilities into Beam workflows, allowing pipelines to store newly computed or updated feature data. It is essential for enabling real-time feature serving, continuous feature engineering, and managing the full lifecycle of feature data in modern machine learning systems.

## Objective

Build a configurable and efficient Beam Sink I/O for Tecton that supports writing records to a feature view. It should enable seamless integration of feature engineering pipelines with Tecton for real-time and batch feature serving.

## Background

Beam currently has Vertex AI and Feast feature store enrichment handlers [1] [2], which enable pipelines to retrieve features for enrichment. However, the Beam ecosystem lacks a native Sink I/O for feature stores, which is essential for use cases where newly computed or updated features need to be stored back into feature stores for serving or continuous learning pipelines. On the first iteration of development and evolution we support Tecton as another feature store. Future iterations of development can add Sink I/Os for Vertex AI and Feast.

# Design and Implementation

The Tecton Stream Ingest Sink for Beam is designed to efficiently write records into Tecton's feature store using the Stream Ingest API [3]. This service, an endpoint for ingesting or updating features with sub-second latency, ensures records are immediately available for training and inference. The Sink transforms incoming Beam elements into a format compatible with Tecton's API, and batch inserts to optimize performance.

## Constraints

- Record Batching Limit: Each stream source can be mapped to a maximum of 10 records to be ingested in a single request.
- Predefined Stream Source and Feature View: Both the Stream Source and its associated Stream Feature View must be predefined in Tecton prior to pipeline execution [4]. The Sink operates on these existing abstract layers and does not handle their creation or modification.
- Predefined Schema: The target Stream Source in Tecton, including its defined schema, must exist prior to pipeline execution. The Sink does not create or modify Tecton Stream Sources.

## Connection/Sink Parameters

The Tecton Stream Write Sink defines its configuration through the TectonStreamWriteConfig data class, which encapsulates all necessary parameters for connecting to and writing data to Tecton's Stream Ingest API. This configuration is passed to the WriteToTectonStream PTransform.

| Parameter | Type | Description | Default |
|---|---|---|---|
| tecton_instance | str | Tecton instance domain (e.g., 'explore.tecton.ai') | *Required* |

| | | | |
|---|---|---|---|
| api_key | str | Tecton API key for authentication | *Required* |
| workspace_name | str | Name of the Tecton workspace | *Required* |
| stream_source_name | str | Name of the Stream Source | *Required* |
| batch_size | int | Number of records to batch together. | 10 |
| max_retries | int | Maximum number of retries for failed requests. | 3 |
| timeout | float | Request timeout in seconds. | 30.0 |

The API endpoint for the Stream Ingest API is https://preview.{instance}/ingest.

## Metrics

The Tecton Sink I/O will expose several metrics using Beam's metrics API, providing insights into its performance and operational health. These metrics will be accessible through Beam's monitoring interfaces (e.g., Dataflow Monitoring UI, Flink UI).

| Metric | Description |
|---|---|
| records_written | A counter that tracks the total number of records successfully written to Tecton, providing a high-level view of data throughput. |
| write_requests | A counter that increments each time a batch of records is sent as an HTTP request to the Tecton Stream Ingest API, monitoring API call frequency. |
| write_errors | A counter that tracks the number of failed write requests, crucial for identifying and |

| | |
|---|---|
| | troubleshooting data ingestion issues. |

# Testing Framework

## Infrastructure

For development, a free tier version on explore.tecton.ai can be used to test and evaluate the Sink I/O.

For deployment, a GitHub secret can be used to store the Tecton API key so that integration tests can be run against a live Tecton instance in the CI environment.

## Unit Testing

The unit tests for the Tecton Sink I/O focus on validating user-provided data and verifying the conversion logic from Beam row elements to a format compatible with Tecton. These tests are isolated and do not require a running Tecton instance, making them fast, lightweight, and suitable for both local development and continuous integration environments.

## Integration Testing

Integration tests validate the behavior of the Tecton Sink I/O when interacting with a live Tecton instance under various real-world conditions. These tests are designed to align with Beam I/O integration testing standards [5].

| Category | Scenario | Expected Outcome |
|---|---|---|
| Insertion | Stream Source not found | Error (404) |
| | Workspace not found | Error (404) |
| | Invalid API Key | Error (401/403) |
| | Schema mismatch | Error (400) |
| | Steam Source exists | Success (200) |
| | Batching behavior under configured thresholds | Records are grouped and flushed (200) |

| | | |
|---|---|---|
| Error Handling | Network interruption | Retries and eventual success or failure |
| | Tecton API rate limiting | Retries with backoff (429) |
| | Malformed record data | Error and/or skipping of malformed records (400) |
| | Timeout exceeded | Error |

## Notebook Example

A Jupyter notebook example showcasing Tecton Sink I/O is planned to be included as part of the official Beam documentation [16]. This notebook will demonstrate key usage patterns, configuration options, and end-to-end data ingestion workflows using the Tecton connector.

It is typically included as a follow-up pull request (PR) to the I/O implementation to ensure the documentation remains functional and aligned with the latest code.

## Documentation

To reflect the addition of Tecton Sink I/O, updating the release notes of the current *unreleased* Beam version to signal that Beam now includes built-in support for Beam as a sink connector.

In a follow-up PR, including a new row in the Beam I/O connectors table [17] to list Tecton I/O as an officially supported built-in connector.

## References and Resources

[1] Vertex AI Enrichment Handler | Apache Beam Source Code

[2] Feast Enrichment Handler | Apache Beam Source Code

[3] Tecton Streaming Ingestion Service | Tecton HTTP API Documentation

[4] Tecton Streaming Ingestion Service | Tecton Documentation

[5] I/O Standards | Apache Beam Documentation