IBM StreamSets

Build, deploy and manage real-time streaming data pipelines with an intuitive visual oriented design

Highlights

Streaming data integration coupled with extensive connectivity

Drag-and-drop visual pipeline designer

Data security and privacy with IBM StreamSets

Full-Featured SDK for Python

In today's highly-competitive environment, organizations seeking a strategic advantage need to leverage all the information at their disposal. The pressure to respond to market volatility, customer demands and digital transformation initiatives necessitates the access to continuous, real-time data across the enterprise. To meet these needs, companies must streamline data integration, helping to enable innovation while maintaining control over data flows. This requires implementing resilient, adaptive streaming pipelines that can handle high-velocity data and unexpected changes.

Given the unique needs of enterprises and due to specific use cases, the IBM Data Fabric architecture is entirely fit-for-purpose. Customers can choose from a set of seamlessly integrated data integration products that fit their needs, whether they be for artificial intelligence (AI), business intelligence and analytics, or other industry-specific requirements.

IBM's Data Integration portfolio includes industry-leading tools such as IBM DataStage for moving and transforming mission-critical data with ETL/ELT processing, and IBM StreamSets is a strategic addition to this portfolio. StreamSets enables customers to build real-time streaming data pipelines, complementing the existing capabilities of IBM DataStage. With IBM Databand, the data observability solution for data pipeline monitoring and issue remediation underpinning the entire portfolio, StreamSets is a seamless and comprehensive solution for designing, deploying and managing data pipelines across all data sources and integration patterns.

With StreamSets organizations get an innovative visual-oriented approach to building real-time data pipelines to capture and stream data in real-time, regardless of its structure or complexity, allowing for faster responses to changing business conditions, more informed decisions and greater innovation. Workloads can run in the same physical location where data resides as StreamSets offers hybrid cloud support, integrating data across multiple cloud platforms and on-premises systems.

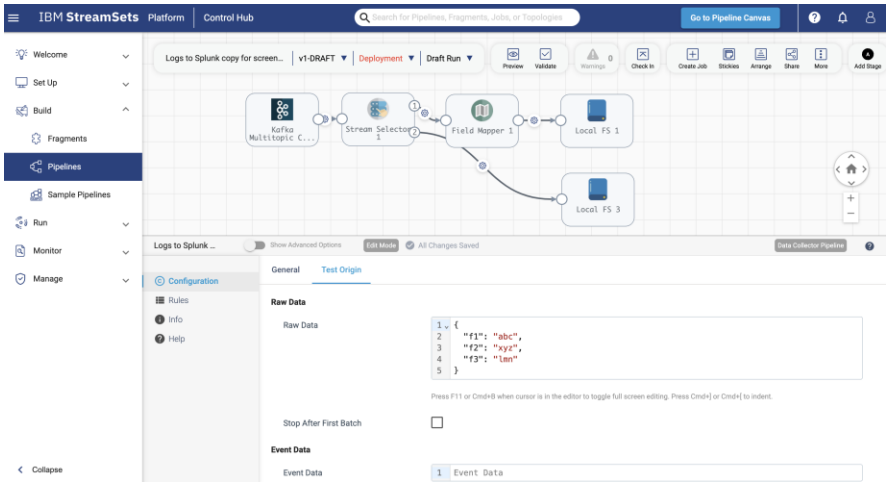
IBM defines real-time data integration as the ability to ingest, process, and write data as soon as it's available, instead of on an intermittent or scheduled basis.

Use cases that benefit from real-time data integration are those for which the extraction of insights with minimal delay within seconds provides business value.

Real-time data integration coupled with extensive connectivity

Real-time data streaming presents several challenges. One primary challenge is ensuring low latency, as delays can negatively impact user experience or decision-making processes. Scalability is another issue, as systems must handle varying loads efficiently without performance degradation. Data consistency and reliability are crucial, especially when dealing with distributed systems where data might arrive out of order or get lost. Managing the infrastructure costs associated with high-throughput and low-latency requirements can be significant. Additionally, ensuring data security and security during transmission and processing is essential to protect against breaches and comply with regulations. Real-time data streaming systems also need robust error handling and fault-tolerance mechanisms to maintain service continuity in the face of failures or network issues.

IBM StreamSets offers a comprehensive data integration solution, connecting with various streaming technologies like Apache Pulsar and Apache Kafka. StreamSets also excels in Change Data Capture (CDC), capturing real-time data changes from sources like Oracle, PostgreSQL, and MongoDB, and delivering them to downstream systems. The platform enhances these capabilities with smart data pipelines, allowing users to schedule and monitor jobs, which define the pipeline and execution engine for data processing. Users can perform transformations securely and efficiently within these pipelines. Furthermore, StreamSets introduces the concept of pipeline fragments, which are reusable sets of connected stages that can be incorporated into multiple pipelines, ensuring consistent processing logic across different data flows and simplifying pipeline design and maintenance.



87%

organizations require data to be ingested and analyzed within 1 day or faster[1]

Drag-and-drop visual pipeline designer

IBM StreamSets provides a comprehensive web-based user interface (UI) in which users can design, execute and monitor their pipelines. At its core is the pipeline canvas, a workspace dedicated to these tasks. Users can view and adjust the properties of a pipeline or its selected stages, with panels that can be resized, minimized, or maximized for convenience. The Preview panel shows data entering and exiting specific stages or groups of stages, along with stage properties and preview configurations, while the Monitor panel delivers real-time metrics and statistics for running pipelines. The UI also allows users to view a list of all available pipelines and related information, with filtering options such as Running Pipelines.

A graphical data integration methodology offers numerous business benefits, including ease of use and enhanced productivity, as it simplifies complex processes and makes them accessible to users with varying technical expertise. The intuitive drag-and-drop interfaces and visual workflows streamline the creation and management of data pipelines, speeding up development time and allowing for quicker adaptation to changing requirements. These tools also improve collaboration by facilitating better communication among team members and stakeholders, reducing errors through visual representations that make it easier to identify and correct issues. Additionally, GUI-based tools offer scalability to handle growing data volumes and complexity, enhanced monitoring and management capabilities for real-time visibility into data flows, and cost efficiency by lowering development and maintenance costs. They provide flexibility and integration with a wide range of data sources and existing systems, support compliance and governance needs, and empower users with self-service capabilities, reducing dependency on IT departments and fostering a culture of data-driven decision-making.

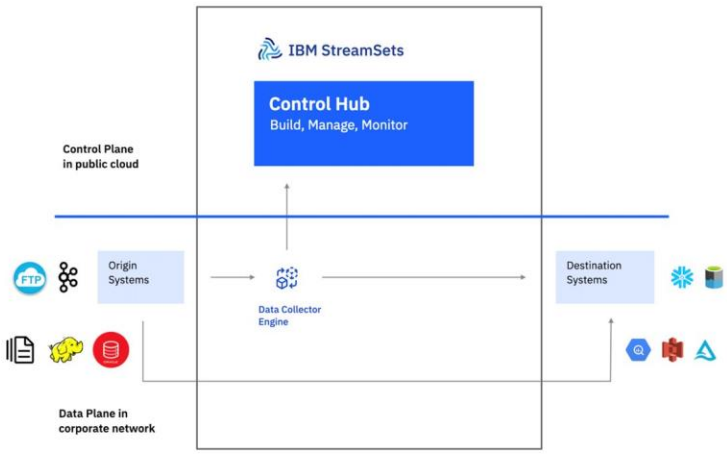
IBM StreamSets

provides powerful, flexible, and secure data integration solutions that cater to the diverse needs of modern enterprises

Ensuring data security and privacy with IBM StreamSets

IBM StreamSets has has a ”secure by design" architecture in which a cloud-based control plane is completely separated from the customer-deployed data plane. The control plane, also known as Control Hub, is a cloud-native application that serves as the central management system for StreamSets' data integration platform. This powerful tool enables users to create, schedule, manage and monitor pipelines, engines and jobs from a single, unified interface. Designed specifically for cloud environments and managed by StreamSets, Control Hub offers a comprehensive solution for orchestrating data flows across an organization. In contrast, the Data Plane houses the actual ingestion and transformation engines, operating securely in either on-premise or cloud environments, providing flexibility to meet diverse infrastructure requirements. The StreamSets' service offering includes a unique feature that allows customers to snapshot pipeline data for configuration and debugging purposes. It's important to note that this feature is not enabled by default and is entirely under the customer's control. For organizations with stringent firewall policies, it offers the option to implement a direct connection between the user's browser and the data engines, further enhancing data security.

StreamSets' visibility into customer data is carefully delineated to balance functionality with privacy and security concerns. Organizations can access to three main categories of data: configuration data, usage data, and telemetry data. Configuration data encompasses information about the customer's assets, developed pipelines, and administrative settings such as user roles and permissions. This data is crucial for tailoring the instance to each customer's specific implementation. Given its sensitive nature, this data is securely maintained within the application. For customers using the SaaS version or DataOps platform, this data resides in a public cloud with robust security measures. Usage data, primarily in the form of metering information, is collected for billing purposes and to help customers estimate costs for potential usage expansion. Telemetry data is gathered to support and improve StreamSets' services. This information helps the company prioritize feature development and enhancements based on customer usage patterns, such as identifying widely used integrations or stages. Additionally, telemetry data contributes to improving overall performance, efficiency, and uptime of stages, pipelines, and services. By collecting and analyzing this data.



Python

is the second most popular programming language and has continued to strengthen its position. As a primary language, it outperformed Java and has closed the gap with JavaScript[2]

Full-featured SDK for Python

With the IBM StreamSets SDK for Python, a user can templatize data pipelines for scale, easily creating hundreds of pipelines with just a few lines of code. The Python SDK integrates with the UI-based tool for programmatic creation and management of pipelines and jobs. A user can create and import a functional pipeline in less than 10 lines of code:

There are several advantages to using the Python SDK by StreamSets -

Flexibility:

The StreamSets Python SDK empowers developers to interact with functionalities programmatically, offering a greater degree of control and flexibility. Developers can define pipeline stages, data sources, and destinations using Python code, enabling the expression of complex data integration logic in a familiar programming language. This programmability facilitates the implementation of custom business logic, transformations, and integrations, providing enhanced flexibility and customization options.

Reproducibility and automation:

With the Python SDK, developers can automate the creation, configuration, and management of data pipelines through code. This ensures reproducibility and consistency across different environments and deployments. By encapsulating the pipeline logic in code, it becomes easier to version, test, and deploy changes, promoting a more streamlined and efficient development process. Additionally, automation reduces the potential for manual errors, enhances pipeline scalability, and allows for easier maintenance and updates.

Monitoring and alerting capabilities:

The Python SDK provides real-time monitoring and alerting capabilities, allowing developers to proactively monitor the health and performance of data pipelines. With built-in metrics and statistics, developers can track pipeline execution, identify bottlenecks, and optimize performance. The SDK also supports event-driven notifications and alerts, enabling timely responses to pipeline issues or anomalies, leading to improved data pipeline reliability and performance.

```
# Import the ControlHub module from the SDK.
from streamsets.sdk import ControlHub

# Connect to the Control Hub instance you want to interact with, and show that the ControlHub object was su
# created by displaying it's URL.
sch = ControlHub("https://cloud.streamsets.com", username="username@org", password="password")

# Instantiate the PipelineBuilder instance to build the pipeline, and add stages to the pipeline.
pipeline_builder = sch.get_pipeline_builder()
s3_origin = pipeline_builder.add_stage('Amazon S3', type='origin')
expression_eval = pipeline_builder.add_stage('Expression Evaluator')
stream_selector = pipeline_builder.add_stage('Stream Selector')
s3_destination = pipeline_builder.add_stage('Amazon S3', type='destination')
localfs_destination = pipeline_builder.add_stage('Local FS')

# Configure the S3 origin stage
s3_origin.access_key_id = 'ABCDEF123456'
s3_origin.secret_access_key = '98765ZYXWV1234LMNOP'
s3_origin.bucket = 'files-to-process'
# Read all CSV files found in this directory by setting the prefix_pattern to the expression '*.csv',
s3_origin.prefix_pattern = '*.csv'
# Configure the pipeline to read CSV files by setting the data_format to 'DELIMITED'.
s3_origin.data_format = 'DELIMITED'

# Configure the Expression Evaluator to write True or False to the 'paymentRequired' field based on a condi
# statement written in expression language.
expression_eval.field_expressions = [{'fieldToSet': '/paymentRequired', 'expression': '${(record:value(\'/i
                                     {'fieldToSet': '/processedAt', 'expression': '${time:now()}'}]

# Configure the S3 destination stage
s3_destination.access_key_id = 'ABCDEF123456'
s3_destination.secret_access_key = '98765ZYXWV1234LMNOP'
s3_destination.bucket = 'invoices-unpaid'
# Set the object suffix for the files created on S3.
s3_destination.object_name_suffix = 'csv'
# Configure the pipeline to write CSV files by setting the data_format to 'DELIMITED'.
s3_destination.data_format = 'DELIMITED'

# Configure the directory_template to point to the filepath where the output records should be written on t
# local filesystem.
localfs_destination.directory_template = '/data/paid-invoices'
# Configure the data_format to be CSV.
localfs_destination.data_format = 'DELIMITED'

# Connect the stages of the pipeline together
s3_origin >> expression_eval >> stream_selector >> s3_destination
stream_selector >> localfs_destination
```

For more information

To learn more about IBM StreamSets, contact your IBM representative or IBM Business Partner, give it a try or visit here to sign up for the latest updates.

1. Forrester, “The Future Of Data: Make It Fast”, 2018
2. Jet Brains, “The State of Developer Ecosystem 2022”, 2002

IBM Corporation
New Orchard Road
Armonk, NY 10504

Produced in the
United States of America
August 2024