SAM Overview 3

Streaming Analytics Manager Overview

Date of Publish: 2018-12-18

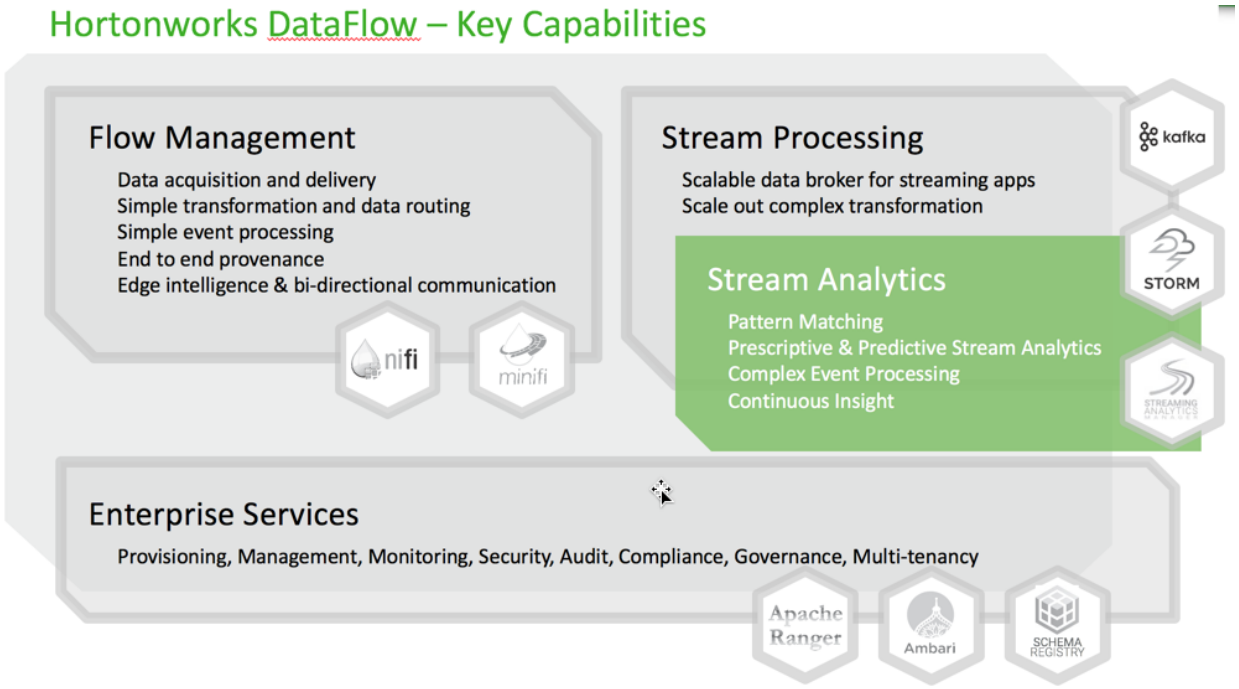
https://docs.hortonworks.com/

Contents

Streaming Analytics Manager Overview

The Hortonworks DataFlow Platform (HDF) provides flow management, stream processing, and enterprise services for collecting, curating, analyzing and acting on data in motion across on-premise data centers and cloud environments.

As the following diagram illustrates, Hortonworks Streaming Analytics Manager (SAM) is an application within the stream processing suite of the HDF platform:

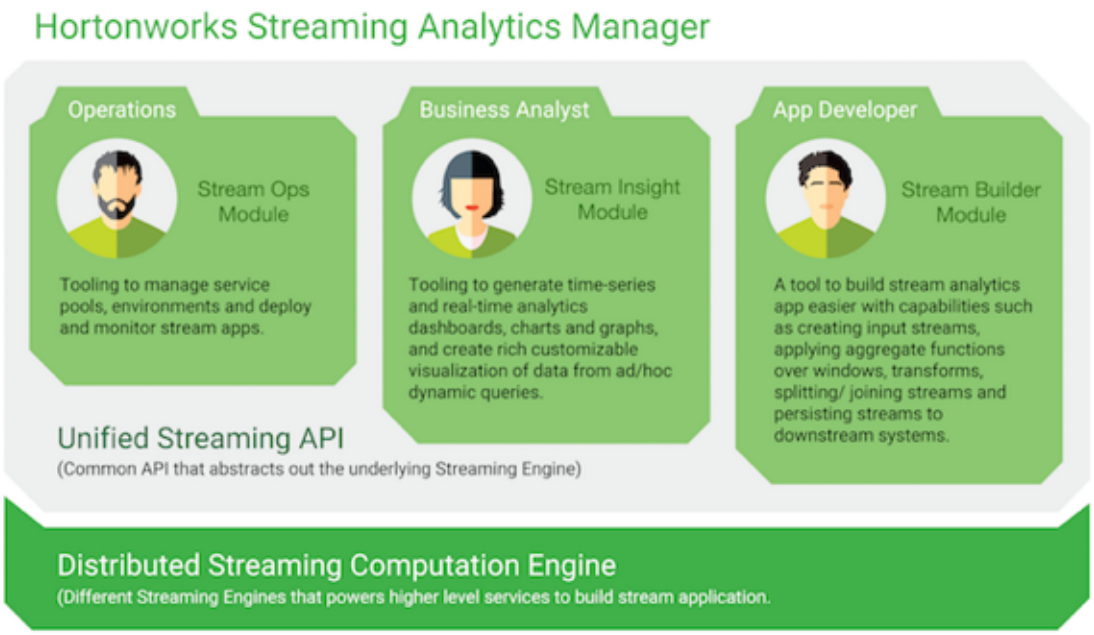


Use Streaming Analytics Manager to design, develop, deploy and manage streaming analytics apps with a drag-and-drop visualization paradigm. Streaming Analytics Manager allows you to build streaming analytics apps for event correlation, context enrichment, complex pattern matching, and analytic aggregations. You can create alerts and notifications when insights are discovered.

Streaming Analytics Manager is agnostic to the underlying streaming engine, and it can support multiple streaming substrates such as Storm, Spark Streaming, Flink, etc. The first streaming engine fully supported is Apache Storm.

Streaming Analytics Manager Modules

Streaming Analytics Manager is composed of several different modules that cater to different user personas. The following diagram illustrates Streaming Analytics Manager modules.



You can think of Streaming Analytics Manager as an application that operates on top of a streaming engine (a "gray box") that allows you build stream apps faster on top of your selected streaming substrate. The unified streaming API provides the abstraction that allows you to plug in different streaming engines.

Streaming Analytics Manager Taxonomy

The following table describes the taxonomy for Streaming Analytics Manager. This taxonomy will be used throughout the rest of this guide.

Term	Description
Streaming Analytics Manager	The name of the graphical application for building, deploying, and managing stream apps.
Stream App	A streaming application built using Streaming Analytics Manager.
My Applications	The landing page for the Streaming Analytics Manager application. The Dashboard has a list of stream apps.
App Tile	Located on the dashboard, an app tile provides metrics, status and lifecyle actions for a stream. Each stream app is displayed as an app tile.
Stream Builder	The Streaming Analytics Manager tool that is used to build stream apps.
Builder Canvas	The canvas of the Stream Builder, on which stream apps are built. The canvas includes a palette of Builder components that can be used to build a stream app.

Term	Description
Builder Components	Building blocks available on the Builder Canvas palette, which can be used to build stream apps. There are four types of Builder components: • Source Builder Component: for creating streams from data sources such as Kafka topics or HDFS files. • Processor Builder Component: for manipulating and processing events in a stream, such as routing, applying transformations, performing windowing operations, and applying rules. • Sink Builder Component: for sending events to other systems such as HBase, HDFS, and Kafka. • Custom Builder Component: for creating custom requirements and adding them to the canvas palette.
Tile component	A component tile that has been moved onto the Builder Canvas, configurable for use in a specific stream app.
Connectors	Define connections between component tiles, directing a flow of tuples and how they flow (such as shuffle grouping).
Stream Operation	A view showing a running stream app, providing metrics for the app. After you use Stream Builder to build and deploy a stream app, the Stream Operation view allows you to monitor the running app.
Service Pool	A pool of services that can be used to create different environments. Services can come from two sources: • Ambari-managed cluster: if you specify an Ambari URL, a service pool is populated with all of the services managed by that Ambari Instance; for example, Storm, HDFS, HBase, and Kafka. • Custom service pool: for services not managed by Ambari, you can create a custom service and add that to a pool. Examples include Elastic Service and the Schema Registry Service.
Environment	A set of services you choose from one or more service pools. The environment is then associated with a stream app, which uses those services in that environment for various configurations.
Stream Insight Superset	The name of the Stream Insight module within SAM for Business Analysts to create dashboards and visualizations
Insight Data Source	A analytics cube powered by Druid where events can be streamed into for rollups/aggregations/analytics
Insight Slice	An insight visualization that can be created from a cube. An insight can be added to a dashboard
Insight Dashboard	Consists of a set of insight slices. Dashboards are created by the Business analyst in the Stream Insights Superset module.

Streaming Analytics Manager Personas

Four main modules within Streaming Analytics Manager offer services to different personas in an organization:

User Persona	Module	Module Features and Functionality
IT Engineer, Operations Engineer, Platform Engineer, Platform Operator	Stream Management	• Create service pools and environments. • Provision, manage and monitor stream apps. • Scale out or scale in stream apps based on resource consumption.

User Persona	Module	Module Features and Functionality
Application Developer	Stream Builder	• The Stream Builder tool assists in building analytic-focused stream apps. • The tool creates streams for event correlation, context enrichment, complex pattern matching, and aggregation. It can create alerts and notifications when patterns are detected and insights are discovered. • The interface uses a drag-and-drop visual programming paradigm.
Business Analyst, Data Analyst	Stream Insight Superset	• The Stream Insight tool assists in generating time-series and real-time analytics dashboards, charts, and graphs of metrics, alerts and notifications. • The tool provides interactive, ad-hoc analytics. You can issue ad-hoc queries, perform multidimensional analyses, and visualize the results in rich configurable dashboards. • The tool offers a self-service ability to create alerts and notification dashboards based on insights derived from the real-time streaming data flows.
SDK Developer	Unified Streaming API	• The unified streaming API abstracts out the underlying streaming engine, making it more straightforward to implement custom components. Initial support is for Storm.

The following subsections describe responsibilities for each persona. For additional information, see the following chapters in this guide:

Persona	Chapter Reference
IT Engineer, Operations Engineer, Platform Engineer, Platform Operator	Installing and Configuring Streaming Analytics Manager Managing Stream Apps
Application Developer	Running the Sample App Building an End-to-End Stream App
Business Analyst, Data Analyst	Creating Visualizations: Insight Slices
SDK Developer	Adding Custom Builder Components

Platform Operator Persona

A platform operator typically manages the Streaming Analytics Manager platform, and provisions various services and resources for the application development team. Common responsibilities of a platform operator include:

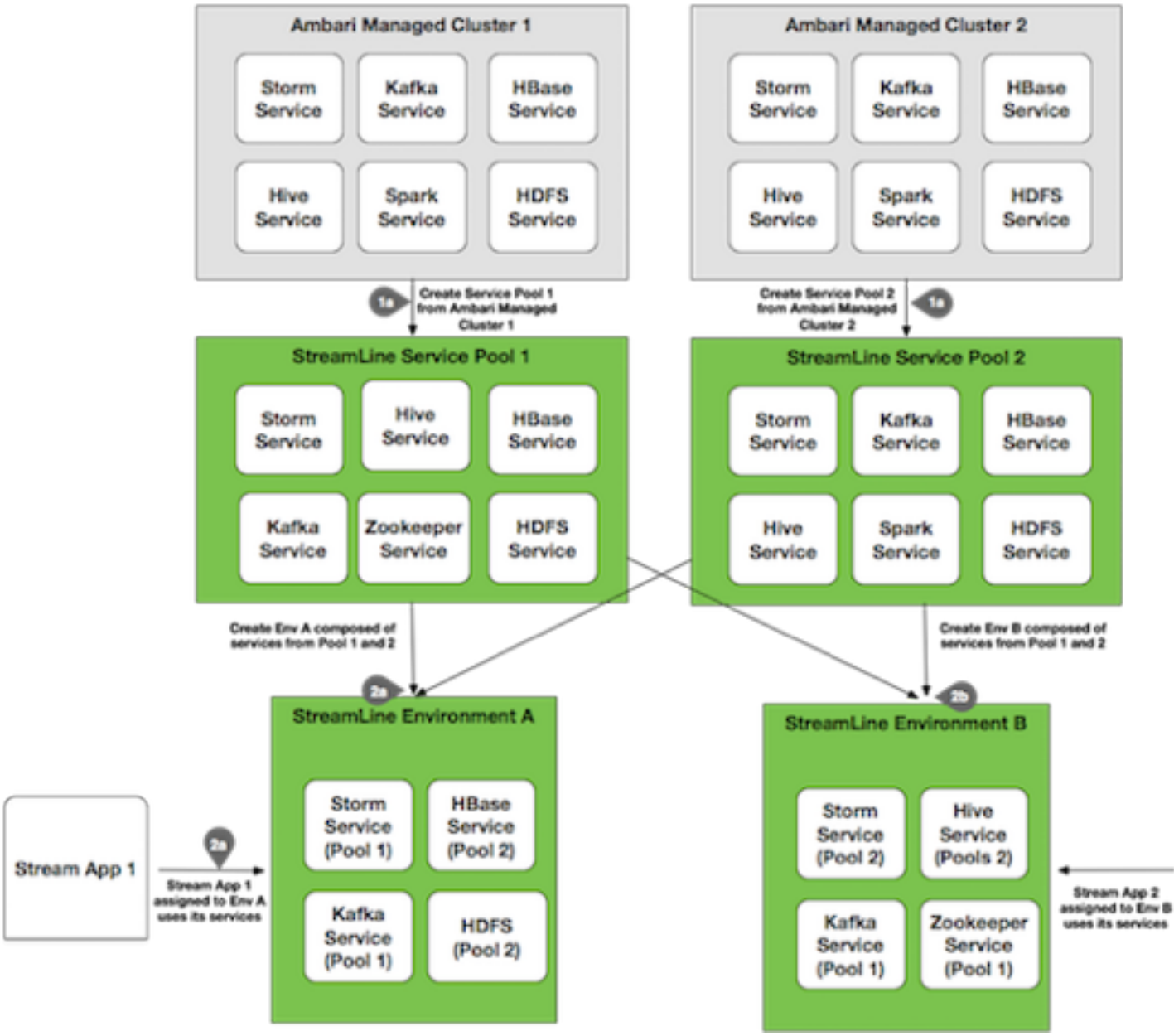
- Installing and managing the Streaming Analytics Manager application platform.
- Provisioning and providing access to services (e.g. big data services like Kafka, Storm, HDFS, HBase) for use by the development team when building stream apps.
- Provisioning and providing access to environments such as development, testing, and production, for use by the development team when provisioning stream apps.

Services, Service Pools and Environments

To perform these responsibilities, a platform operator works with three important abstractions in Streaming Analytics Manager:

- Service is an entity that an application developer works with to build stream apps. Examples of services could be a Storm cluster that the stream app will be deployed to, a Kafka cluster that is used by the stream app to create a streams, or a HBase cluster that the stream app writes to.
- Service Pool is a set of services associated with an Ambari managed cluster
- Environment is a named entity that represents a set of services chosen from different service pools. A stream app is assigned to an environment and the app can only use the services associated with an environment.

The following diagram illustrates these constructs:



The Service, Service Pool, and Environment abstractions provide the following benefits:

1. Simplicity and ease of use: An application developer can use the Service abstraction without needing to focus on configuration details. For example, to deploy a stream app to a Storm cluster, the developer does not need to consider how to configure the Storm cluster (Nimbus host, ports, and so on). Instead, the developer simply

selects the Storm service from the environment associated with the app. The service abstract out all the details/ complexities.

2. Ease of propagating a stream app between environments: With Service as an abstraction, it is easy for the stream operator or application developer to move a stream app from one environment to another. They simply export the stream app and import it into a different environment.

Application Developer Persona

The application developer uses the Stream Builder component to design, implement, deploy, and debug stream apps in Streaming Analytics Manager.

The following subsections describe component building blocks and schema requirements.

Component Building Blocks

Stream Builder offers several building blocks for stream apps: sources, processors, sinks, and custom components.

Sources

Source builder components are used to create data streams. SAM has the following sources:

- Kafka
- Azure Event Hub
- HDFS

Processors

Processor builder components are used to manipulate events in the stream.

The following table lists processors that are available with Streaming Analytics Manager.

Processor Name	Description
Join	• Joins two streams together based on a field from each stream. • Two join types are supported: inner and left. • Joins are based on a window that you can configure based on time or count.

Rule	• Allows you to configure rule conditions that route events to different streams.
	• Standard conditional operators are supported for rules.
	• Configuring a rule has two modes:
	  • General: Guided rule creation using drop-down menus.
	  • Advanced: Write complex SQL to construct a rule.
	• Rules are translated to SQL to be applied on the stream.
	• An event goes through all the conditions and if it matches multiple rules the event is sent to all the matching output streams.

Aggregate	• Performs functions over windows of events. • Two types of windows are supported: tumbling and sliding. • You can create window criteria based on time interval and count. • Window functions supported out of the box include: stddev, stddevp, variance, variancecep, avg, min, max, sum, count. The system is extensible to add custom functions as well.
Projection	• Applies transformations to the events in the stream • Extensive set of OOO functions and the ability to add your own functions
Branch	• Performs a standard if-else construct for routing. • The even is routed to the first rule it matches. Once an event has matched a rule, no further condition search is performed.
PMML	• Executes a PMML model that is stored in the Model Registry. PMML has been minimally tested as part of the Tech Preview, and should not be used.

Sinks

Sink builder components are used to send events to other systems.

Streaming Analytics Manager supports the following sinks:

- Kafka
- Druid
- HDFS
- HBase
- Hive
- JDBC
- OpenTSDB
- Notification (OOO support Kafka and the ability to add custom notifications)
- Cassandra
- Solr

Custom Components

For more information about developing custom components, see *SDK Developer Persona*.

Schema Requirements

Unlike NiFi (the flow management service of the HDF platform), Streaming Analytics Manager requires a schema for stream apps. More specifically, every Builder component requires a schema to function.

The primary data stream source is Kafka, which uses the HDF Schema Registry.

The Builder component for Apache Kafka is integrated with the Schema Registry. When you configure a Kafka source and supply a Kafka topic, Streaming Analytics Manager calls the Schema Registry. Using the Kafka topic as the key, Streaming Analytics Manager retrieves the schema. This schema is then displayed on the tile component, and is passed to downstream components.

Analyst Persona

A business analyst uses the Streaming Analytics Manager Stream Insight module to create time-series and real-time analytics dashboards, charts and graphs; and create rich customizable visualizations of data.

Stream Insight Key Concepts

The following table describes key concepts of the Stream Insights module.

Stream Insight Concept	Description
Analytics Engine	• Stream Insight analytics engine is powered by Druid, an open source data store designed for OLAP queries on event data. • Data can be streamed into the Analytics engine via the Druid/ Analytics Engine Sink that app developers can use when building streaming apps. The analytics engine sink can stream data into new/existing insight cubes.
Insight Data Source	• A insight data source is powered by Druid that represents the store for streaming data. The cube can be queried to do rollups, aggregations and other powerful analytics
Insight Slice	• A visualization that can be created from asking questions of the data source. An insight can be added to the dashboard
Dashboard	• Consists of a set of slices. Dashboards are created by the Business analysts to perform descriptive analytics

A business analyst can create a wide array of visualizations to gather insights on streaming data.

The platform supports over 30+ visualizations the business analyst can create.

SDK Developer Persona

Streaming Analytics Manager supports the development of custom functionality through the use of its SDK.