Azure Event Hubs — A big data streaming platform and event ingestion service

06/23/2020 • 4 minutes to read • +4

In this article

Azure Event Hubs is a big data streaming platform and event ingestion service. It can receive and process millions of events per second. Data sent to an event hub can be transformed and stored by using any real-time analytics provider or batching/storage adapters.

The following scenarios are some of the scenarios where you can use Event Hubs:

- Anomaly detection (fraud/outliers)
- Application logging
- Analytics pipelines, such as clickstreams
- Live dashboarding
- Archiving data
- Transaction processing
- User telemetry processing
- Device telemetry streaming

Ingest, buffer, store, and process your stream in real time to get actionable insights. Event Hubs uses a partitioned consumer model, enabling multiple applications to process the stream concurrently and letting you control the speed of processing.

Capture your data in near-real time in an Azure Blob storage or Azure Data Lake Storage for long-term retention or micro-batch processing. You can achieve this behavior on the same stream you use for deriving real-time analytics. Setting up capture of event data is fast. There are no administrative costs to run it, and it scales automatically with Event Hubs throughput units. Event Hubs enables you to focus on data processing rather than on data capture.

Azure Event Hubs also integrates with Azure Functions for a serverless architecture.

Scalable

With Event Hubs, you can start with data streams in megabytes, and grow to gigabytes or terabytes. The Auto-inflate feature is one of the many options available to scale the number of throughput units to meet your usage needs.

Rich ecosystem

Event Hubs for Apache Kafka ecosystems enables Apache Kafka (1.0 and later) clients and applications to talk to Event Hubs. You do not need to set up, configure, and manage your own Kafka clusters.

With a broad ecosystem available in various languages .NET, Java, Python, JavaScript, you can easily start processing your streams from Event Hubs. All supported client languages provide low-level integration. The ecosystem also provides you with seamless integration with Azure services like Azure Stream Analytics and Azure Functions and thus enables you to build serverless architectures.

Key architecture components

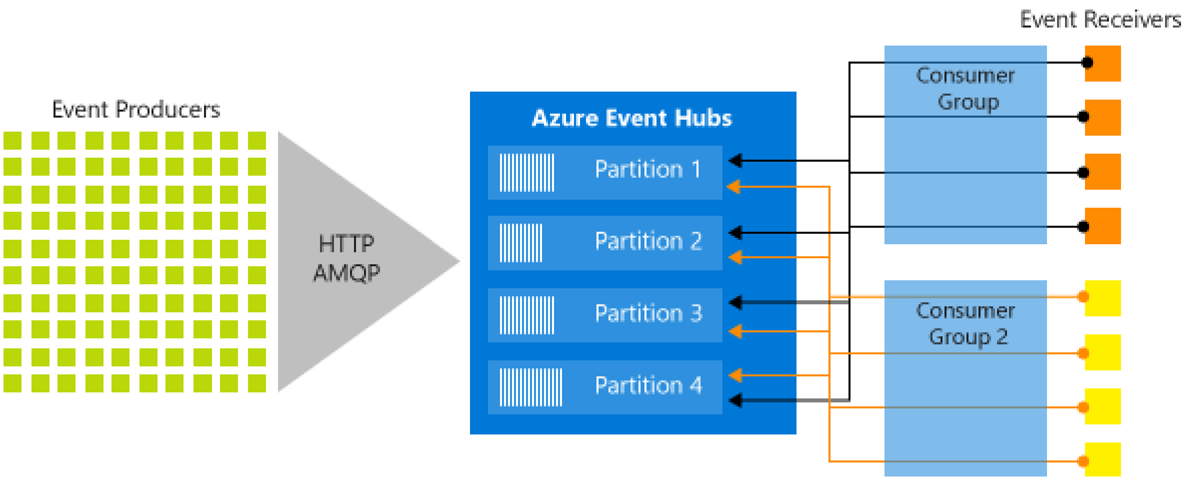
Event Hubs contains the following key components:

- **Event producers**: Any entity that sends data to an event hub. Event publishers can publish events using HTTPS or AMQP 1.0 or Apache Kafka (1.0 and above)
- **Partitions**: Each consumer only reads a specific subset, or partition, of the message stream.
- **Consumer groups**: A view (state, position, or offset) of an entire event hub. Consumer groups enable consuming applications to each have a separate view of

the event stream. They read the stream independently at their own pace and with their own offsets.

- **Throughput units**: Pre-purchased units of capacity that control the throughput capacity of Event Hubs.
- **Event receivers**: Any entity that reads event data from an event hub. All Event Hubs consumers connect via the AMQP 1.0 session. The Event Hubs service delivers events through a session as they become available. All Kafka consumers connect via the Kafka protocol 1.0 and later.

The following figure shows the Event Hubs stream processing architecture:



Event Hubs on Azure Stack Hub

Event Hubs on Azure Stack Hub allows you to realize hybrid cloud scenarios. Streaming and event-based solutions are supported, for both on-premises and Azure cloud processing. Whether your scenario is hybrid (connected), or disconnected, your solution can support processing of events/streams at large scale. Your scenario is only bound by the Event Hubs cluster size, which you can provision according to your needs.

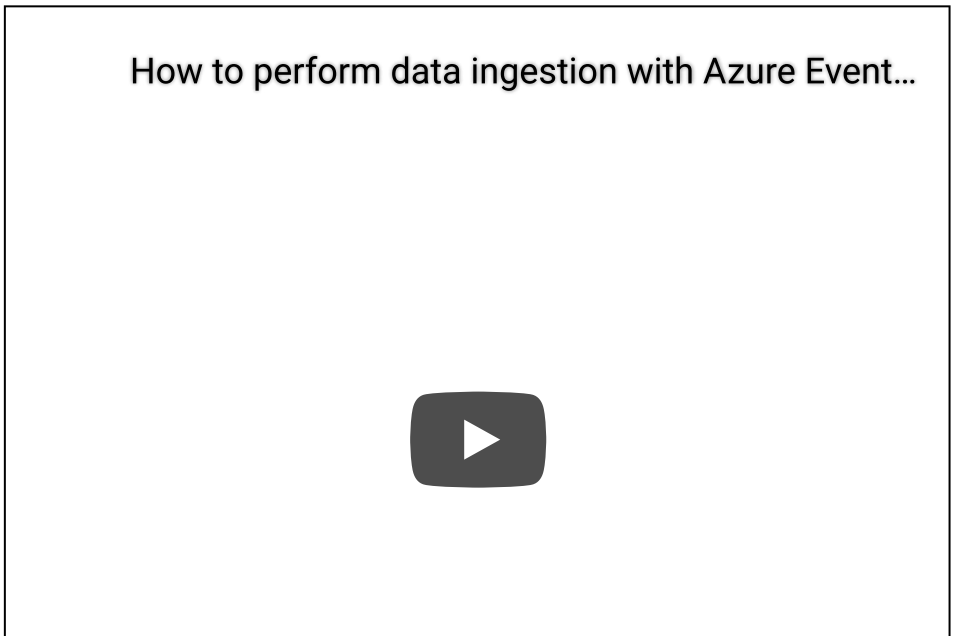
The Event Hubs editions (on Azure Stack Hub and on Azure) offer a high degree of feature parity. This parity means SDKs, samples, PowerShell, CLI, and portals offer a similar experience, with few differences.

Event Hubs on Stack is free during public preview. For more information, see Event Hubs on Azure Stack Hub overview.

Next steps

To get started using Event Hubs, see the **Send and receive events** tutorials:

- .NET Core



Why use Event Hubs?

Data is valuable only when there is an easy way to process and get timely insights from data sources. Event Hubs provides a distributed stream processing platform with low latency and seamless integration, with data and analytics services inside and outside Azure to build your complete big data pipeline.

Event Hubs represents the "front door" for an event pipeline, often called an *event ingestor* in solution architectures. An event ingestor is a component or service that sits between event publishers and event consumers to decouple the production of an event stream from the consumption of those events. Event Hubs provides a unified streaming platform with time retention buffer, decoupling event producers from event consumers.

The following sections describe key features of the Azure Event Hubs service:

Fully managed PaaS

Event Hubs is a fully managed Platform-as-a-Service (PaaS) with little configuration or management overhead, so you focus on your business solutions. Event Hubs for Apache Kafka ecosystems gives you the PaaS Kafka experience without having to manage, configure, or run your clusters.

Support for real-time and batch processing

- Java
- Python
- JavaScript
- Go
- C (send only)
- Apache Storm (receive only)

To learn more about Event Hubs, see the following articles:

- Event Hubs features overview
- Frequently asked questions.