

Use Azure Event Hubs from Apache Kafka applications

07/20/2020 • 4 minutes to read • 

In this article

[What does Event Hubs for Kafka provide?](#)

[Other Event Hubs features available for Kafka](#)

[Features that are not yet supported](#)

[Next steps](#)

Event Hubs provides a Kafka endpoint that can be used by your existing Kafka based applications as an alternative to running your own Kafka cluster. Event Hubs supports [Apache Kafka protocol 1.0 and later](#), and works with your existing Kafka applications, including MirrorMaker.

Go serverless: Big data processing with Azure Event Hubs for Apac...



What does Event Hubs for Kafka provide?

The Event Hubs for Kafka feature provides a protocol head on top of Azure Event Hubs that is binary compatible with Kafka versions 1.0 and later for both reading from and writing to Kafka topics. You may start using the Kafka endpoint from your applications with no code change but a minimal configuration change. You update the connection string in configurations to point to the Kafka endpoint exposed by your event hub instead of pointing to your Kafka cluster. Then, you can start streaming events from

your applications that use the Kafka protocol into Event Hubs. This integration also

<https://docs.microsoft.com/en-us/azure/event-hubs/event-hubs-for-kafka-ecosystem-overview>

supports frameworks like [Kafka Connect](#), which is currently in preview.

Conceptually Kafka and Event Hubs are nearly identical: they're both partitioned logs built for streaming data. The following table maps concepts between Kafka and Event Hubs.

Kafka and Event Hub conceptual mapping

Kafka Concept	Event Hubs Concept
Cluster	Namespace
Topic	Event Hub
Partition	Partition
Consumer Group	Consumer Group
Offset	Offset

Key differences between Kafka and Event Hubs

While [Apache Kafka](#) is software, which you can run wherever you choose, Event Hubs is a cloud service similar to Azure Blob Storage. There are no servers or networks to manage and no brokers to configure. You create a namespace, which is an FQDN in which your topics live, and then create Event Hubs or topics within that namespace. For more information about Event Hubs and namespaces, see [Event Hubs features](#). As a cloud service, Event Hubs uses a single stable virtual IP address as the endpoint, so clients don't need to know about the brokers or machines within a cluster.

Scale in Event Hubs is controlled by how many throughput units you purchase, with each throughput unit entitling you to 1 MB per second, or 1000 events per second of ingress. By default, Event Hubs scales up throughput units when you reach your limit with the [Auto-Inflate](#) feature; this feature also works with the Event Hubs for Kafka feature.

Security and authentication

Every time you publish or consume events from an Event Hubs for Kafka, your client is trying to access the Event Hubs resources. You want to ensure that the resources are

accessed using an authorized entity. When using Apache Kafka protocol with your


clients, you can set your configuration for authentication and encryption using the SASL mechanisms. When using Event Hubs for Kafka requires the TLS-encryption (as all data in transit with Event Hubs is TLS encrypted). It can be done specifying the SASL_SSL option in your configuration file.

Azure Event Hubs provides multiple options to authorize access to your secure resources.

- OAuth
- Shared access signature (SAS)


OAuth

Event Hubs integrates with Azure Active Directory (Azure AD), which provides a **OAuth** 2.0 compliant centralized authorization server. With Azure AD, you can use role-based access control (RBAC) to grant fine grained permissions to your client identities. You can use this feature with your Kafka clients by specifying **SASL_SSL** for the protocol and **OAUTHBEARER** for the mechanism. For details about RBAC roles and levels for scoping access, see [Authorize access with Azure AD](#).

XML	 Copy
<pre>bootstrap.servers=NAMESPACENAME.servicebus.windows.net:9093 security.protocol=SASL_SSL sasl.mechanism=OAUTHBEARER sasl.jaas.config=org.apache.kafka.common.security.oauthbearer.OAuthBearerLoginModule required; sasl.login.callback.handler.class=CustomAuthenticateCallbackHandler;</pre>	

Shared Access Signature (SAS)

Event Hubs also provides the **Shared Access Signatures (SAS)** for delegated access to Event Hubs for Kafka resources. Authorizing access using OAuth 2.0 token-based mechanism provides superior security and ease of use over SAS. The built-in roles can also eliminate the need for ACL-based authorization, which has to be maintained and managed by the user. You can use this feature with your Kafka clients by specifying **SASL_SSL** for the protocol and **PLAIN** for the mechanism.

XML	 Copy
<pre>bootstrap.servers=NAMESPACENAME.servicebus.windows.net:9093 security.protocol=SASL_SSL sasl.mechanism=PLAIN sasl.jaas.config=org.apache.kafka.common.security.plain.PlainLoginModule</pre>	

```
required username="$ConnectionString" password="{YOUR.EVENTHUBS.CONNECTION.STRING}";
```

Samples

For a **tutorial** with step-by-step instructions to create an event hub and access it using SAS or OAuth, see [Quickstart: Data streaming with Event Hubs using the Kafka protocol](#).

For more **samples** that show how to use OAuth with Event Hubs for Kafka, see [samples on GitHub](#).

Other Event Hubs features available for Kafka

The Event Hubs for Kafka feature enables you to write with one protocol and read with another, so that your current Kafka producers can continue publishing via Kafka, and you can add readers with Event Hubs, such as Azure Stream Analytics or Azure Functions. Additionally, Event Hubs features such as [Capture](#) and [Geo Disaster-Recovery](#) also work with the Event Hubs for Kafka feature.

Features that are not yet supported

Here is the list of Kafka features that are not yet supported:

- Transaction
- Compression
- Size-based retention
- Log compaction
- HTTP Kafka API support
- Kafka Streams

Next steps

This article provided an introduction to Event Hubs for Kafka. To learn more, see [Apache Kafka developer guide for Azure Event Hubs](#).

Is this page helpful?

 Yes  No

