Contents

[Logic Apps 1](#_Toc50163426)

[Azure Functions 1](#_Toc50163427)

[Event Grid ( IFTT for Azure service & plugins) 1](#_Toc50163428)

[Terminology 2](#_Toc50163429)

[Event Hub ( Kafka ) 2](#_Toc50163430)

[Key architecture components 3](#_Toc50163431)

[Features and terminology in Azure Event Hubs 4](#_Toc50163432)

[Namespace 4](#_Toc50163433)

[Event Hubs for Apache Kafka 4](#_Toc50163434)

[Event publishers 4](#_Toc50163435)

[Publishing an event 4](#_Toc50163436)

[Capture 4](#_Toc50163437)

[SAS tokens 4](#_Toc50163438)

[Consumer groups 5](#_Toc50163439)

[Event Hub FAQs 5](#_Toc50163440)

[Azure Relay 6](#_Toc50163441)

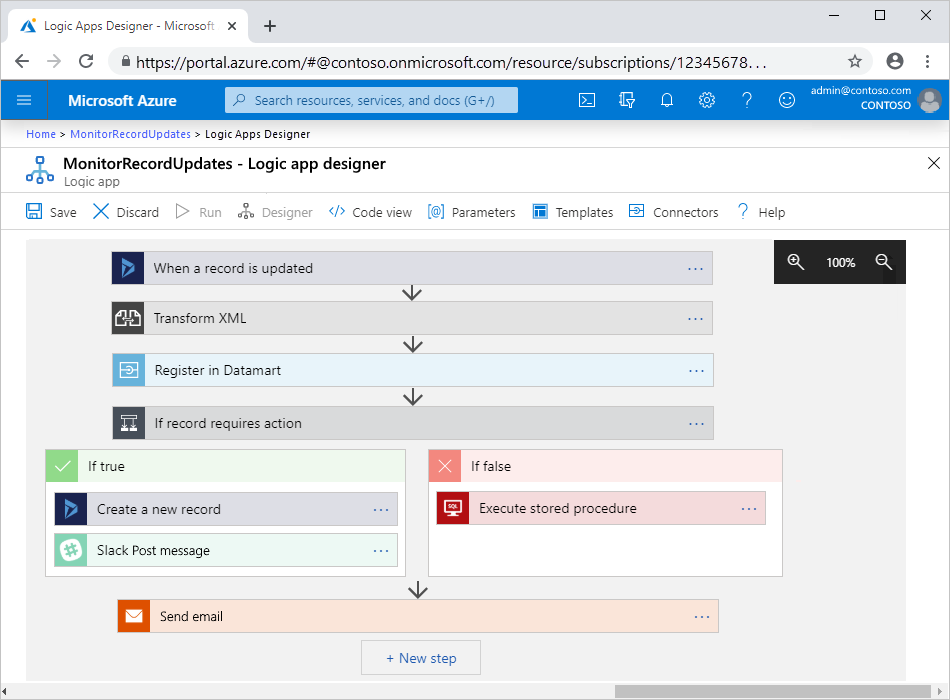
[Service Bus 6](#_Toc50163442)

# Logic Apps

<https://docs.microsoft.com/en-us/azure/logic-apps/logic-apps-overview>

[Azure Logic Apps](https://azure.microsoft.com/services/logic-apps) is a cloud service that helps you schedule, automate, and orchestrate tasks, business processes, and [workflows](https://docs.microsoft.com/en-us/azure/logic-apps/logic-apps-overview#logic-app-concepts) when you need to integrate apps, data, systems, and services across enterprises or organizations.

Every logic app workflow starts with a trigger, which fires when a specific event happens, or when new available data meets specific criteria. Many triggers provided by the connectors in Logic Apps include basic scheduling capabilities so that you can set up how regularly your workloads run.



## Key terms

* **Workflow**: Visualize, design, build, automate, and deploy business processes as series of steps.
* **Managed connectors**: Your logic apps need access to data, services, and systems. You can use prebuilt Microsoft-managed connectors that are designed to connect, access, and work with your data. See [Connectors for Azure Logic Apps](https://docs.microsoft.com/en-us/azure/connectors/apis-list).
* **Triggers**: Many Microsoft-managed connectors provide triggers that fire when events or new data meet specified conditions. For example, an event might be getting an email or detecting changes in your Azure Storage account. Each time the trigger fires, the Logic Apps engine creates a new logic app instance that runs the workflow.
* **Actions**: Actions are all the steps that happen after the trigger. Each action usually maps to an operation that's defined by a managed connector, custom API, or custom connector.
* **Enterprise Integration Pack**: For more advanced integration scenarios, Logic Apps includes capabilities from BizTalk Server. The Enterprise Integration Pack provides connectors that help logic apps easily perform validation, transformation, and more.

<https://docs.microsoft.com/en-us/azure/logic-apps/quickstart-create-first-logic-app-workflow>

# Azure Functions

In Azure Functions, a function app provides the execution context for your individual functions. Function app behaviors apply to all functions hosted by a given function app. All functions in a function app must be of the same [language](https://docs.microsoft.com/en-us/azure/azure-functions/supported-languages).

|  | **Durable Functions** | **Logic Apps** |
| --- | --- | --- |
| **Development** | Code-first (imperative) | Designer-first (declarative) |
| **Connectivity** | [About a dozen built-in binding types](https://docs.microsoft.com/en-us/azure/azure-functions/functions-triggers-bindings#supported-bindings), write code for custom bindings | [Large collection of connectors](https://docs.microsoft.com/en-us/azure/connectors/apis-list), [Enterprise Integration Pack for B2B scenarios](https://docs.microsoft.com/en-us/azure/logic-apps/logic-apps-enterprise-integration-overview), [build custom connectors](https://docs.microsoft.com/en-us/azure/logic-apps/custom-connector-overview) |
| **Actions** | Each activity is an Azure function; write code for activity functions | [Large collection of ready-made actions](https://docs.microsoft.com/en-us/azure/logic-apps/logic-apps-workflow-actions-triggers) |
| **Monitoring** | [Azure Application Insights](https://docs.microsoft.com/en-us/azure/azure-monitor/app/app-insights-overview) | [Azure portal](https://docs.microsoft.com/en-us/azure/logic-apps/quickstart-create-first-logic-app-workflow), [Azure Monitor logs](https://docs.microsoft.com/en-us/azure/logic-apps/monitor-logic-apps) |
| **Management** | [REST API](https://docs.microsoft.com/en-us/azure/azure-functions/durable/durable-functions-http-api), [Visual Studio](https://docs.microsoft.com/en-us/visualstudio/azure/vs-azure-tools-resources-managing-with-cloud-explorer?view=vs-2019) | [Azure portal](https://docs.microsoft.com/en-us/azure/logic-apps/quickstart-create-first-logic-app-workflow), [REST API](https://docs.microsoft.com/en-us/rest/api/logic/), [PowerShell](https://docs.microsoft.com/en-us/powershell/module/az.logicapp), [Visual Studio](https://docs.microsoft.com/en-us/azure/logic-apps/manage-logic-apps-with-visual-studio) |
| **Execution context** | Can run [locally](https://docs.microsoft.com/en-us/azure/azure-functions/functions-runtime-overview) or in the cloud | Runs only in the cloud |

# Event Grid ( IFTT for Azure service & plugins)

<https://medium.com/microsoftazure/azure-event-grid-the-whole-story-4b7b4ec4ad23>

[Azure Event Grid](https://jlik.me/e1r) is a cloud service that provides infrastructure for event-driven computing. Event Grid focuses on **events** ( messages ) that declare, “something happened.”

Most Azure services automatically send messages through Event Grid and many can directly consume messages “out of the box.” Azure Event Grid supports a “push model.” In many cases it can also lead to cost savings because it removes the overhead of polling on a regular basis and instead triggers code only when it is needed to consume an event.

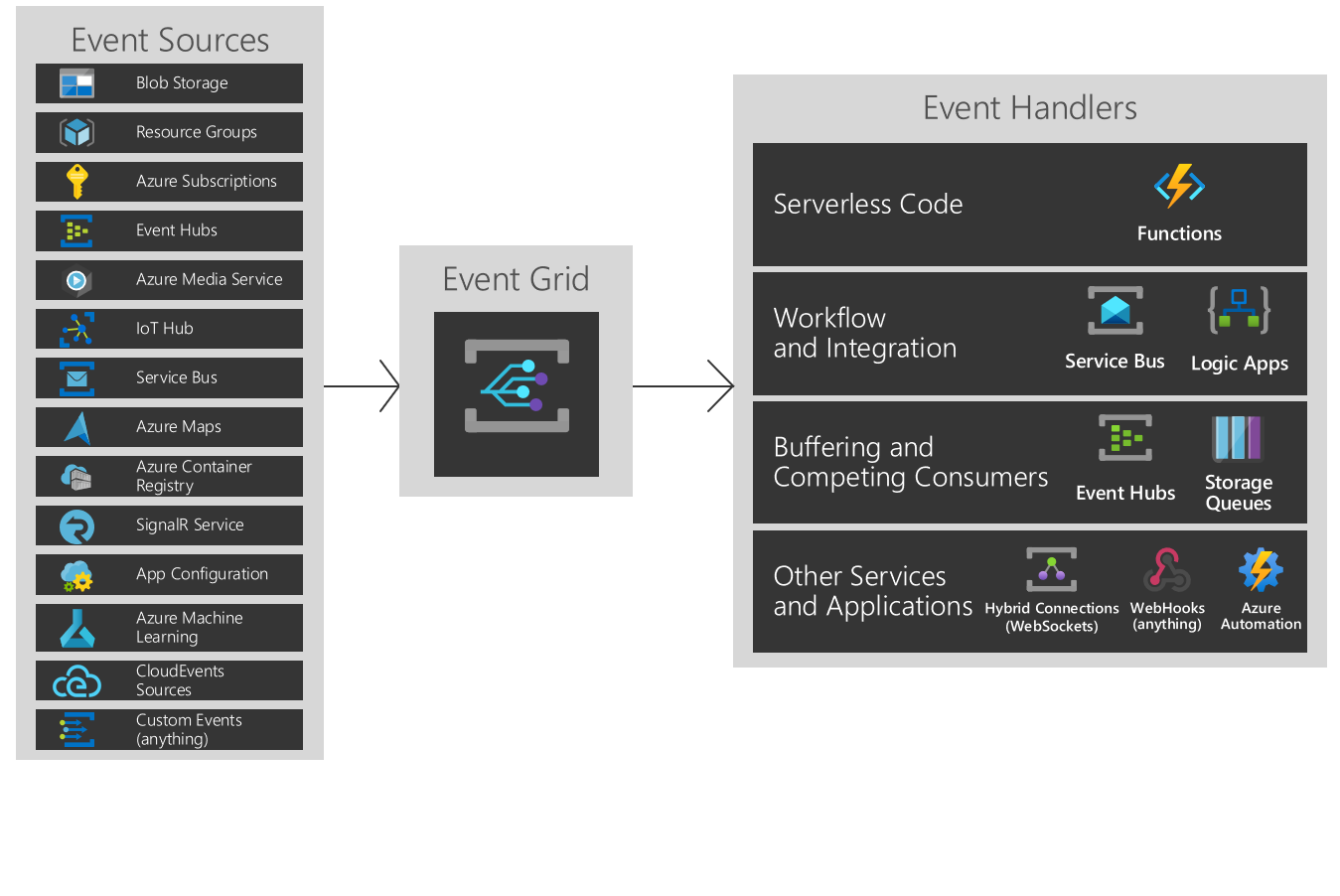
Here’s a list of the major key features of Azure Event Grid:

* Fully managed event routing service
* Built to support event-driven and serverless applications

### Terminology

There are several concepts that are useful to understand when working with Event Grid.

* **Events** — what happened (i.e. “file was uploaded” or “SKU was added”)
* **Event Publishers** — where the event happened (i.e. “web app” or “blob storage” or “CLI tool”)
* **Topics** — a channel for related events (i.e. “storage events” or “inventory events”)
* **Event Subscriptions** — how to receive events. A subscription informs Event Grid that an event should be routed to a handler. A single event can have multiple subscriptions, and subscriptions are named so they can be unsubscribed later if need be.
* **Event Handlers** — the app or service that receives and responds to the event (i.e. “Azure Function” or “Azure Logic App” or “my custom Ruby on Rails app”)



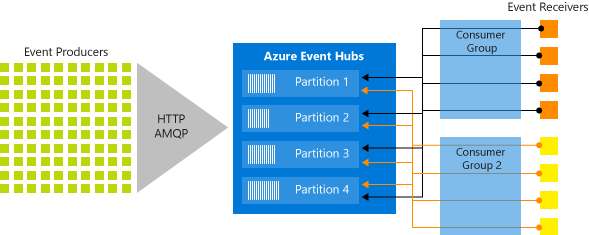
# Event Hub ( Kafka )

<https://docs.microsoft.com/en-us/azure/event-hubs/event-hubs-about>

Azure Event Hubs is a big data streaming platform and event ingestion service. It can receive and process millions of events per second.

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



## Key architecture components

Event Hubs contains the following [key components](https://docs.microsoft.com/en-us/azure/event-hubs/event-hubs-features):

* **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.

## Features and terminology in Azure Event Hubs

<https://docs.microsoft.com/en-us/azure/event-hubs/event-hubs-features>

### Namespace

An Event Hubs namespace provides a unique scoping container, referenced by its [fully qualified domain name](https://en.wikipedia.org/wiki/Fully_qualified_domain_name), in which you create one or **more event hubs or Kafka topics**.

### Event Hubs for Apache Kafka

[This feature](https://docs.microsoft.com/en-us/azure/event-hubs/event-hubs-for-kafka-ecosystem-overview) provides an endpoint that enables customers to talk to Event Hubs using the Kafka protocol. This integration provides customers a Kafka endpoint. **Event Hubs for Apache Kafka supports Kafka protocol 1.0 and later.**

### Event publishers

Any entity that sends data to an event hub is an event producer, or event publisher. Event publishers can publish events using HTTPS or AMQP 1.0 or Kafka 1.0 and later. Event publishers use a Shared Access Signature (SAS) token to identify themselves to an event hub, and can have a unique identity, or use a common SAS token.

### Publishing an event

You can publish an event via AMQP 1.0, Kafka 1.0 (and later), or HTTPS. A single publication (event data instance) **has a limit of 1 MB**, regardless of whether it is a single event or a batch. Publishing events larger than this threshold results in an error. It is a best practice for publishers to be unaware of partitions within the event hub and to only specify a partition key (introduced in the next section), or their identity via their SAS token.

### Capture

[Event Hubs Capture](https://docs.microsoft.com/en-us/azure/event-hubs/event-hubs-capture-overview) enables you to automatically capture the streaming data in Event Hubs and save it to your choice of either a Blob storage account, or an Azure Data Lake Service account. You can enable Capture from the Azure portal, and specify a minimum size and time window to perform the capture.

### SAS tokens

Event Hubs uses Shared Access Signatures, which are available at the namespace and event hub level. A SAS token is generated from a SAS key and is an SHA hash of a URL, encoded in a specific format. Using the name of the key (policy) and the token, Event Hubs can regenerate the hash and thus authenticate the sender.

### Consumer groups

There can be at most 5 concurrent readers on a partition per consumer group; however **it is recommended that there is only one active receiver on a partition per consumer group**. Within a single partition, each reader receives all of the messages. If you have multiple readers on the same partition, then you process duplicate messages. You need to handle this in your code, which may not be trivial. However, it's a valid approach in some scenarios.

## Event Hub FAQs

<https://docs.microsoft.com/en-us/azure/event-hubs/event-hubs-faq>

1. When do I create a new namespace vs. use an existing namespace?

Capacity allocations ([throughput units (TUs)](https://docs.microsoft.com/en-us/azure/event-hubs/event-hubs-faq#throughput-units)) are billed at the namespace level. A namespace is also associated with a region.

You may want to create a new namespace instead of using an existing one in one of the following scenarios:

* You need an Event Hub associated with a new region.
* You need an Event Hub associated with a different subscription.
* You need an Event Hub with a distinct capacity allocation (that is, the capacity need for the namespace with the added event hub would exceed the 40 TU threshold and you don't want to go for the dedicated cluster)

1. What is the difference between Event Hubs Basic and Standard tiers?

The Standard tier of Azure Event Hubs provides features beyond what is available in the Basic tier. The following features are included with Standard:

Longer event retention

Additional brokered connections, with an overage charge for more than the number included

More than a single consumer group

Capture

Kafka integration

1. What is the maximum retention period for events?

Event Hubs Standard tier currently supports a maximum **retention period of seven days**.

1. How many partitions do I need?

The number of partitions is specified at creation and must be between 2 and 32. The partition count isn't changeable, so you should consider long-term scale when setting partition count.

# Azure Relay

# Service Bus