

Agenda: Azure Event Grid

- Introduction.
- How it works?
- Key Features of Event Grid.
- Event Source and Event Handlers.
- Route storage events to Azure Functions.
- Monitor VM changes using Event Grid and Logic Apps.
- Comparing with Service Bus Queues and Topics.

Introduction

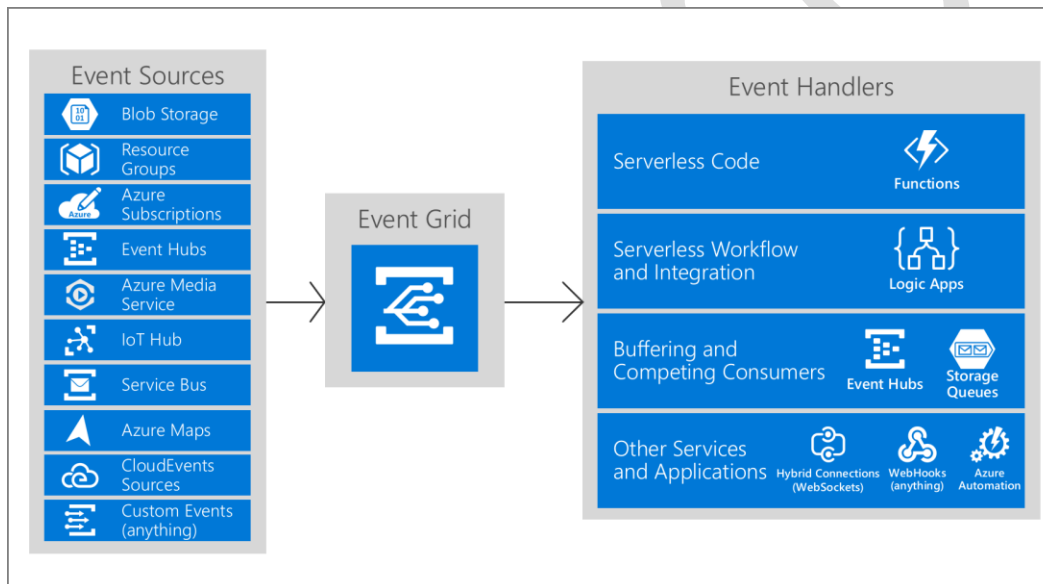
- Event Grid is a fully **managed event service** that enables you to easily manage events across many **different Azure services and applications**. Made for performance and scale, it simplifies building event-driven applications and **serverless architectures**.
- Azure Event Grid allows you to easily build applications with **event-based architectures** where events are delivered in **nearly real time**, so consumers can respond immediately to events as they occur.
- **Eliminate polling** and the associated cost and latency.
- With Event Grid, event publishers are **decoupled** from event subscribers using a pub/sub model and simple HTTP-based event delivery, allowing you to build scalable serverless applications, microservices and distributed systems.
- Event Grid, is an event routing service, made for **performance and massive scale** and responsible for delivery of events within a configurable period of time (up to 24 hours). After that, events are either discarded or **dead lettered**.
- If **archival of event** contents or replayability of event stream are needed, it is possible to facilitate this requirement by setting up an Event Grid **subscription** to the **Event Hub** or a **Azure Storage Queue** where messages can be retained for longer periods and archival of messages is supported.
- Azure Event Grid is deployed to **maximize availability** by natively spreading across multiple **fault domains** in every region, and across **availability zones** (in regions that support them).
- It has the following characteristics:
 - Dynamically scalable.
 - Low cost.
 - Serverless.
 - At least once delivery.

Ensure reliability and performance in your apps

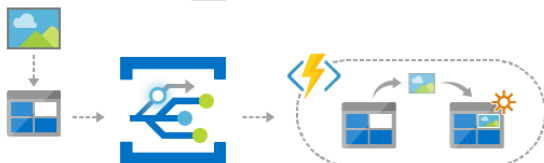


How it Works?

- First, select the Azure resource you would like to subscribe to, and then give the event handler or WebHook endpoint to send the event to.
- Event Grid has built-in support for events coming from Azure services, like storage blobs and resource groups. Event Grid also has support for your **own events**, using **custom topics**.
- You can use **filters** to route specific events to different endpoints, multicast to multiple endpoints, and make sure your events are **reliably** delivered.



Example 1: Event Grid connects data sources and event handlers. For example, use Event Grid to instantly trigger a serverless function to run image analysis each time a new photo is added to a blob storage container.



Deccansoft Software Services.

Error! Hyperlink reference not valid. | <https://www.bestdotnettraining.com/>

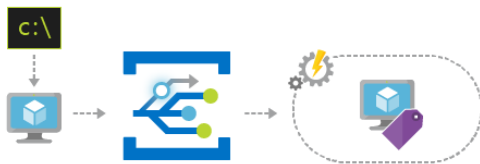
<https://www.AzureA2Z.com>

Phone: +91 40 2784 1517 OR +91 8008327000 (INDIA)

You should opt for **Event Grid trigger** instead of the **Blob storage trigger** when dealing with the following requirements:

- **Blob delete events:** blob delete events are not supported by blob storage triggers.
- **Reduced latency:** if your function app is on the **Consumption plan**, there can be up to a **10-minute delay** in processing new blobs if a function app has gone idle. To avoid this latency, you can **use an Event Grid trigger** or switch to an **App Service plan** with the **Always On** setting enabled.
- **High scale:** High scale can be loosely defined as containers that have more than 100,000 blobs in them or storage accounts that have more than 100 blob updates per second.

Example 2: Event Grid allows you to speed automation and simplify policy enforcement. For example, Event Grid can notify Azure Automation when a virtual machine is created or a SQL Database is spun up. These events can be used to automatically check that service configurations are compliant, put metadata into operations tools, tag virtual machines or file work items.



Here are some of the key features of Azure Event Grid:

- **Simplicity** - Point and click to aim events from your Azure resource to any event handler or endpoint.
- **Advanced filtering** - Filter on event type or event publish path to make sure event handlers only receive relevant events.
- **Fan-out** - Subscribe several endpoints to the same event to send copies of the event to as many places as needed.
- **Reliability** - 24-hour retry with exponential backoff to make sure events are delivered.
- **Pay-per-event** - Pay only for the amount you use Event Grid.
- **High throughput** - Build high-volume workloads on Event Grid with support for **millions of events per second**.
- **Built-in Events** - Get up and running quickly with resource-defined built-in events.
- **Custom Events** - Use Event Grid route, filter, and reliably deliver custom events in your app.

There are five concepts in Azure Event Grid that let you get going:

1. **Events** - What happened.

Every event has common information like: source of the event, time the event took place, and unique identifier. Every event also has specific information that is only relevant to the specific type of event. For example, an event about a new file being created in Azure Storage has details about the file, such as the lastTimeModified value.

Each event is limited to **64 KB** of data.

2. **Event sources** - Where the event took place.
3. **Topics** - The endpoint where publishers send events.
4. **Event subscriptions** - The endpoint or built-in mechanism to route events, sometimes to more than one handler. Subscriptions are also used by handlers to intelligently filter incoming events.
5. **Event handlers** - The app or service reacting to the event.

Example: Route storage events to Azure Functions

1. **Search Subscription → Resource providers → Search Event Grid → Register.**
2. Create an Azure Function App → Add an **Azure Event Grid Trigger** function to it.

```
#r "Microsoft.Azure.EventGrid"
#r "Newtonsoft.Json"
using Newtonsoft.Json;
using Microsoft.Azure.EventGrid.Models;
public static void Run(EventGridEvent eventGridEvent, ILogger log)
{
    dynamic json = JsonConvert.DeserializeObject(eventGridEvent.Data.ToString());
    string url = json.url;
    log.LogInformation(url);
}
```

3. Create a Storage Account with Standard **Storage v2 (v1 is not supported)**
4. Select Storage Account → **Events** → Click on **Add Event Grid Subscription**
 - a. Basic Tab
 - i. Name = BlobAddedListener
 - ii. Topic Types = Storage Accounts
 - iii. Resource = <Existing Storage Name>
 - iv. Event Types = Check **Blob Created** and **Blob Deleted**

- v. Endpoint Type = Azure Function, Select an endpoint (Function created earlier) → Confirm Selection
- b. Filters Tab
 - i. Enable subject filtering
 - ii. Subject Ends With = ".jpg"
 - iii. Save
- 5. Add a file to Blob (image with ext. ".jpg")
- 6. Look at the Log of Azure Function and note the JSON output.

Note: *We don't see system topics in our Azure subscription because the publisher owns the topics, but we can subscribe to them.*

Example: Monitor virtual machine changes by using Azure Event Grid and Logic Apps

- Create a logic app that monitors events from an event grid.
 - Add a condition that specifically checks for virtual machine changes.
 - Send email when your virtual machine changes.
1. Create a Black Logic App (Name = "EventGridDemoLogicApp"), Template="Blank"
 2. Add **Event Grid Trigger**: When a resource event occurs
 - a. Authenticate as a member of the Azure Active Directory that's associated with your Azure subscription, for example, **user-name@emailoutlook.onmicrosoft.com**.
 3. Now subscribe your logic app to events from the publisher. Provide the details about your event subscription as described in the following table, for example:

When a resource event occurs

* Subscription: Pay-As-You-Go

* Resource Type: Microsoft.Resources.ResourceGroups

* Resource Name: My-Virtual-Machine-RG

Event Type Item - 1: Microsoft.Resources.ResourceActionSuccess

Event Type Item - 2: Microsoft.Resources.ResourceDeleteSuccess

Event Type Item - 3: Microsoft.Resources.ResourceWriteSuccess

+ Add new item

Subscription Name: fabrikam-monitor-virtual-machine-event-subscription

Add new parameter

Connected to Sophia.Owen@fabrikam.com. [Change connection.](#)

Deccansoft Software Services.

Error! Hyperlink reference not valid. | <https://www.bestdotnettraining.com/>
<https://www.AzureA2Z.com>

Phone: +91 40 2784 1517 OR +91 8008327000 (INDIA)

Note:

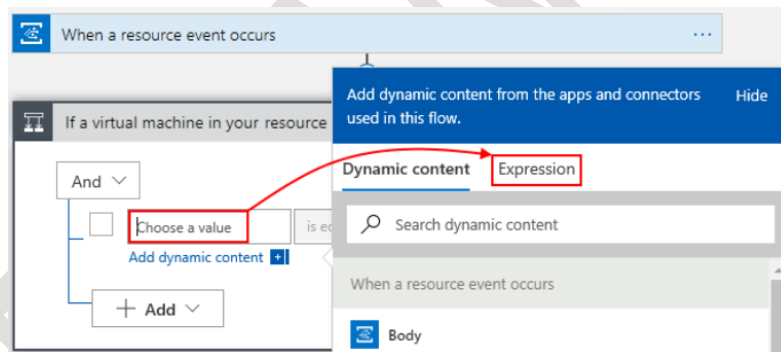
When you save your logic app with an event grid trigger, Azure automatically creates an event subscription for your logic app to your selected resource. So when the resource publishes an event to the event grid, that event grid automatically pushes the event to your logic app. This event triggers your logic app, then creates and runs an instance of the workflow that you define in these next steps.

Add a condition

If you want your logic app to run only when a specific event or operation happens, add a condition that checks for the **Microsoft.Compute/virtualMachines/write** operation. When this condition is true, your logic app sends you email with details about the updated virtual machine.

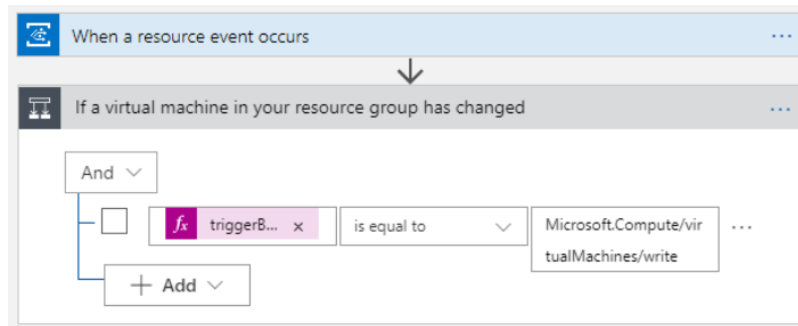
4. Add an empty **condition** to your workflow (Name = "If a virtual machine in your resource group has changed")
5. Create a condition that checks the event body for a data object where the operationName property is equal to the Microsoft.Compute/virtualMachines/write operation.
 - a. In the expression editor, enter this expression, which returns the operation name from the trigger, and select **OK**:

- a. **triggerBody()?['data']['operationName']**



- b. In the middle box, keep the operator **is equal to**.
 - c. In the right box, enter this value, which is the specific operation that you to monitor:
 - a. **Microsoft.Compute/virtualMachines/write**

Your finished condition now looks like this example:



Send email Notification

6. In the condition's **If true** box, select **Add an action** = "Send an email"
7. Provide information about the email from the previous action:

The screenshot shows the configuration of the "Send email when virtual machine updated" action. The form includes the following fields:

- To:** Sophia.Owen@fabrikam.com
- Subject:** Resource updated: [icon] Subject x
- Body:** Resource: [icon] Topic x, Event type: [icon] Event Type x, Event ID: [icon] ID x, Time: [icon] Event Time x

At the bottom, there is a section for "Add new parameter" and a status bar indicating "Connected to Sophia.Owen@fabrikam.com. Change connection."

Add an action

Test your logic app workflow

8. Update your VM – You can resize your VM and after few moments you should get an email.