Microsoft

Azure Data Lake Storage – Configuring Notification Alerts

Prepared by

Data SQL Ninja Engineering Team (askdmjfordmtools@microsoft.com)

Note: The detail provided in this document has been harvested as part of a customer engagement sponsored through the Azure Data Services Jumpstart Program.

Table of Contents

Table of Figures

1 Introduction

Azure Data Lake Storage (ADLS) is a secure, durable, cost-effective and highly scalable cloud data lake offering from Microsoft. Besides providing file based I/O operations, it offers a deep integration with Azure PaaS and SaaS offerings such as Azure Data Factory, Azure Databricks, Azure HDInsight, Azure SQL Datawarehouse and PowerBI. To ensure security, ADLS is well integrated with Azure Active Directory service that allow customers to configure ACL-based roles that match their functional requirements.

This document is primarily focused on one of the key operational requirements – be able to receive notifications and alerts whenever things change on ADLS, using one of the following solution approaches:

- Event Notification & Metrics from ADLS using Log Analytics
- Event Notification & Metrics from ADLS using Event Hub
- Event Notification & Metrics from ADLS using Event Grid

2 Pre-requisites

It is important that you have following available or created before proceeding to further sections below:

- Resource Group, for example - `*adlsalerts_**rg***` (`rg` → Resource Group)
- ADLS Gen 1 Resource, for example - `*adlsalerts**gen1***` (`..gen1` → ADLS Gen1)
- Storage Account Resource, for example - `adlsgen1alerts***sa***` (`..sa` → Storage Account)
- Log Analytics Resource, for example - `adlsgen1alerts***LAWS***` (`..LAWS` → Log Analytics Workspace)
- Monitor → Alerts → Action Group, for example - ` ADLSGen1AlertsActionGroup` (see section further below in this documentation on how to go about creating an action group)

Please review Azure Portal documentation here, on how to approach such definitions, except for action group. There's a brief section further below that describes necessary steps with references to additional detail.

3 ADLS Event Notification Alerts using Log Analytics

This section describes the steps involved in gathering diagnostics logs from Azure Data lake account into Azure log analytics workspace and generating webhook and email alert thereafter.

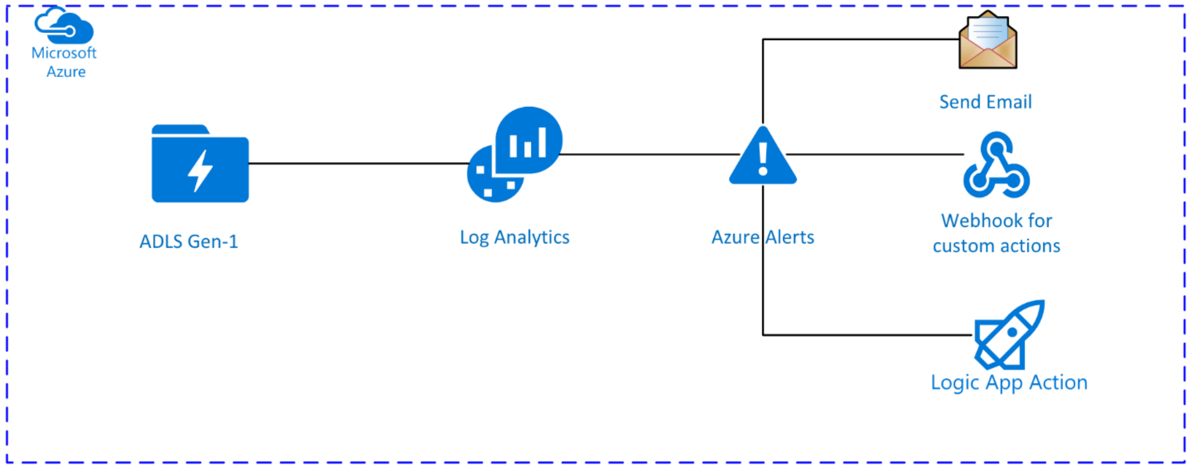


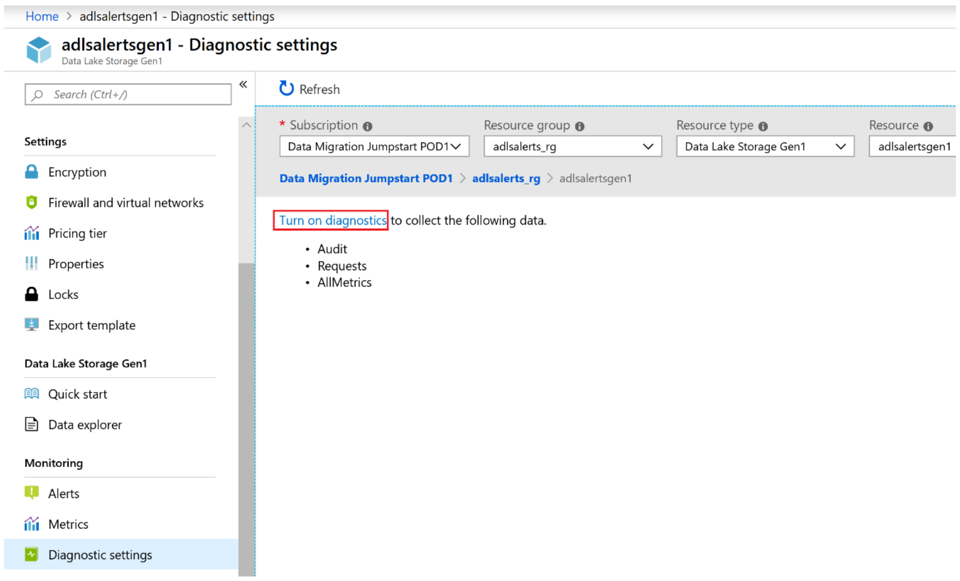
Figure 1: Log Analytics Approach

Following is the high-level outline of steps involved using Log Analytics approach:

- Configure/Enable Diagnostics to point to Azure Log Analytics
- Using Azure Monitor configure alerts from Log Analytics
- Pros and Cons of the approach

3.1 Configure/Enable Diagnostics

As a pre-requisite one must enable or turn-on the diagnostics for the corresponding ADLS Account. Use the Azure Portal Interface to achieve this, as shown in the following visual:



Home > adlsalertsgen1 - Diagnostic settings

adlsalertsgen1 - Diagnostic settings
Data Lake Storage Gen1

Search (Ctrl+/)

Settings
- Encryption
- Firewall and virtual networks
- Pricing tier
- Properties
- Locks
- Export template

Data Lake Storage Gen1
- Quick start
- Data explorer

Monitoring
- Alerts
- Metrics
- Diagnostic settings

Refresh

* Subscription	Resource group	Resource type	Resource
Data Migration Jumpstart POD1	adlsalerts_rg	Data Lake Storage Gen1	adlsalertsgen1

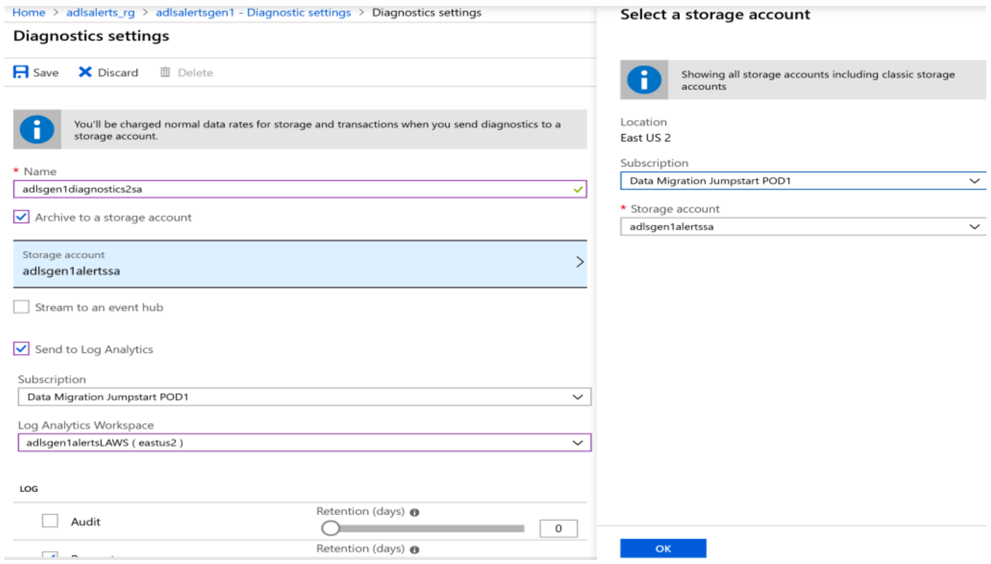
Data Migration Jumpstart POD1 > adlsalerts_rg > adlsalertsgen1

Turn on diagnostics to collect the following data.

- Audit
- Requests
- AllMetrics

Figure 2: Turn-on diagnostics

Click on Turn-on diagnostics. In the subsequent screen as shown below, select storage account and log analytics target. Storage account is used to archive the diagnostic events like create, delete, etc. Log Analytics will collect these events as submitted by ADLS setting and will be available for further triage and reasoning.



Home > adlsalerts_rg > adlsalertsgen1 - Diagnostic settings > Diagnostics settings

Diagnostics settings

Save | Discard | Delete

You'll be charged normal data rates for storage and transactions when you send diagnostics to a storage account.

* Name
adlsgen1diagnostics2sa

[✓] Archive to a storage account

Storage account
adlsgen1alertssa

[ ] Stream to an event hub

[✓] Send to Log Analytics

Subscription
Data Migration Jumpstart POD1

Log Analytics Workspace
adlsgen1alertsLAWS (eastus2)

LOG

[ ] Audit — Retention (days) 0

Retention (days)

Select a storage account

Showing all storage accounts including classic storage accounts

Location
East US 2

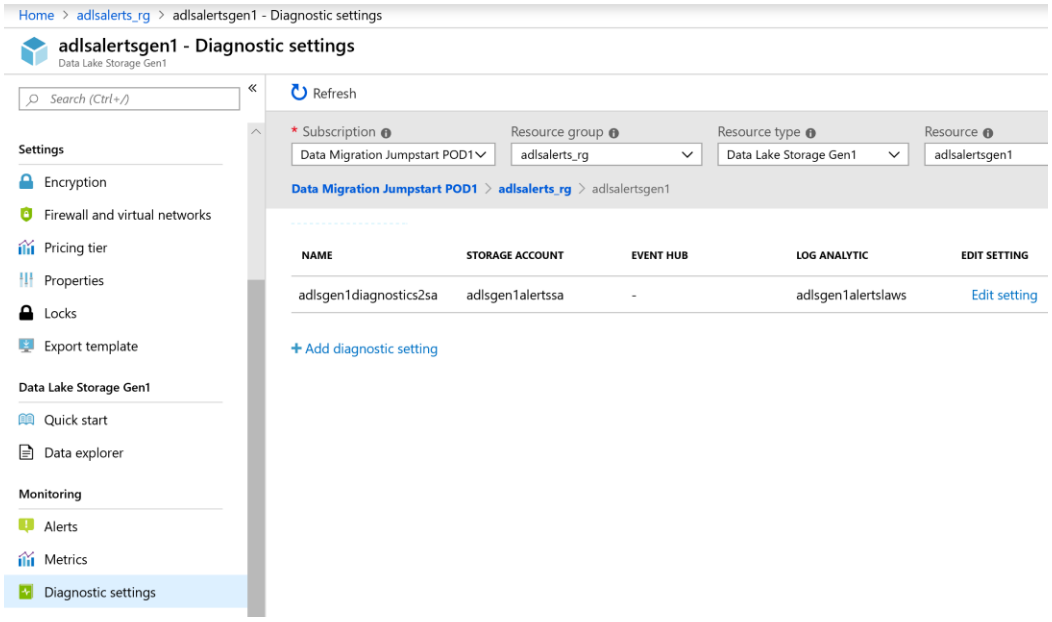
Subscription
Data Migration Jumpstart POD1

* Storage account
adlsgen1alertssa

OK

Figure 3: Turn-on diagnostics - Choose a Storage Account

Hit save once you have made necessary configuration changes. Following screen will now list the diagnostic configuration that you just created:



Home > adlsalerts_rg > adlsalertsgen1 - Diagnostic settings

adlsalertsgen1 - Diagnostic settings
Data Lake Storage Gen1

Subscription: Data Migration Jumpstart POD1
Resource group: adlsalerts_rg
Resource type: Data Lake Storage Gen1
Resource: adlsalertsgen1

Data Migration Jumpstart POD1 > adlsalerts_rg > adlsalertsgen1

NAME	STORAGE ACCOUNT	EVENT HUB	LOG ANALYTIC	EDIT SETTING
adlsgen1diagnostics2sa	adlsgen1alertssa	-	adlsgen1alertslaws	Edit setting

+ Add diagnostic setting

Figure 4: ADLS Gen1 Diagnostics Settings Overview

3.2 Define Action Group

Next step is to configure Action Group! Action groups abstract the alert outcomes as actions. Here is a quick link to Azure Portal documentation that describes how to manage Action Groups.

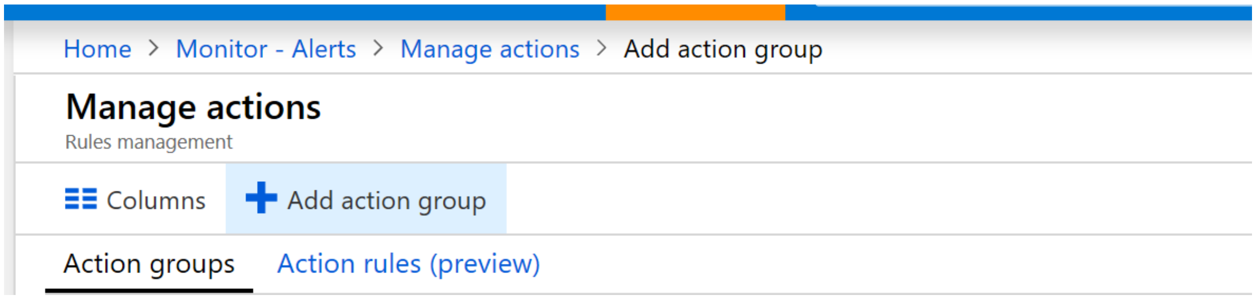
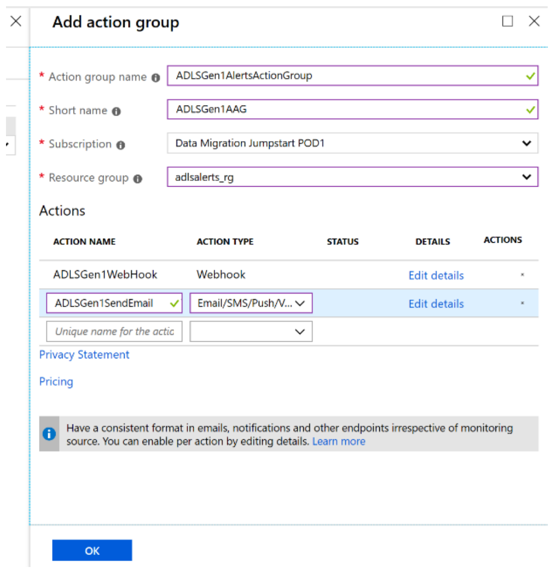


Figure 5: Add action group

Following visual provides the summary detail of actions that this new action group would abstract – an email and a web hook.


Add action group

* Action group name: ADLSGen1AlertsActionGroup
* Short name: ADLSGen1AAG
* Subscription: Data Migration Jumpstart POD1
* Resource group: adlsalerts_rg

Actions

ACTION NAME	ACTION TYPE	STATUS	DETAILS	ACTIONS
ADLSGen1WebHook	Webhook		Edit details	
ADLSGen1SendEmail	Email/SMS/Push/V...		Edit details	
Unique name for the actic				

Privacy Statement

Pricing

Have a consistent format in emails, notifications and other endpoints irrespective of monitoring source. You can enable per action by editing details. Learn more

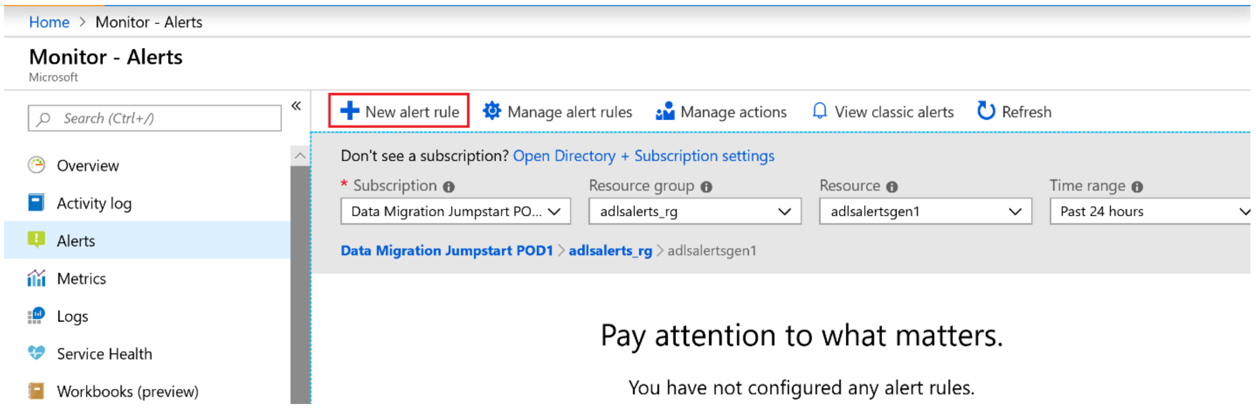
OK

Figure 6: Add Action Group detail

For a test webhook, you can leverage defining a webhook end point using the publicly available developer interface – www.webhook.site URL. A sample URL will look like this - http://webhook.site/45735651-caa0-46a9-9ea1-64109ddeafe2.

3.3 Using Azure Monitor configure alerts from Log Analytics

In the previous step when you have saved the diagnostics configuration, wait for a while, say about few minutes. This will allow the backend to enable a key schema in the log analytics called `*AzureDiagnostics*`. In this section we will go through how to query that schema for events of our interest using Log Analytics query interface. Following visual shows how to navigate to the Alerts configuration feature (path to follow is Home → Monitor → Alerts tab):



Don't see a subscription? Open Directory + Subscription settings

* Subscription: Data Migration Jumpstart PO...
Resource group: adlsalerts_rg
Resource: adlsalertsgen1
Time range: Past 24 hours

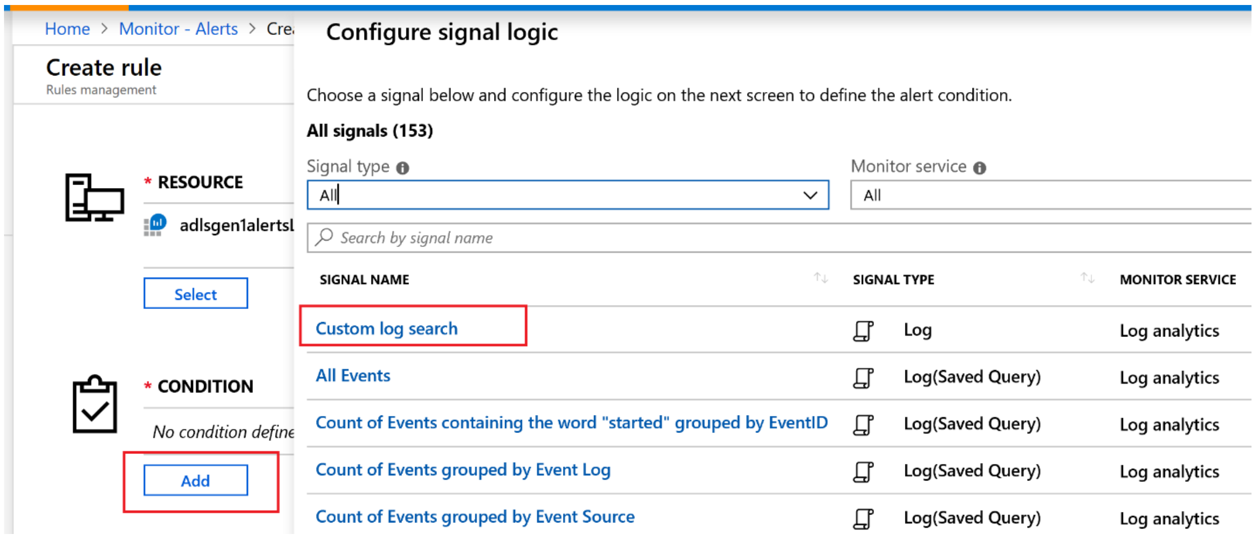
Data Migration Jumpstart POD1 > adlsalerts_rg > adlsalertsgen1

Pay attention to what matters.

You have not configured any alert rules.

Figure 7: Azure Monitor - Alerts View

Click on new alert rule option as highlighted above. In the subsequent screen we will define a condition using a custom log search option, as show in the following visual:


Configure signal logic

Choose a signal below and configure the logic on the next screen to define the alert condition.

All signals (153)

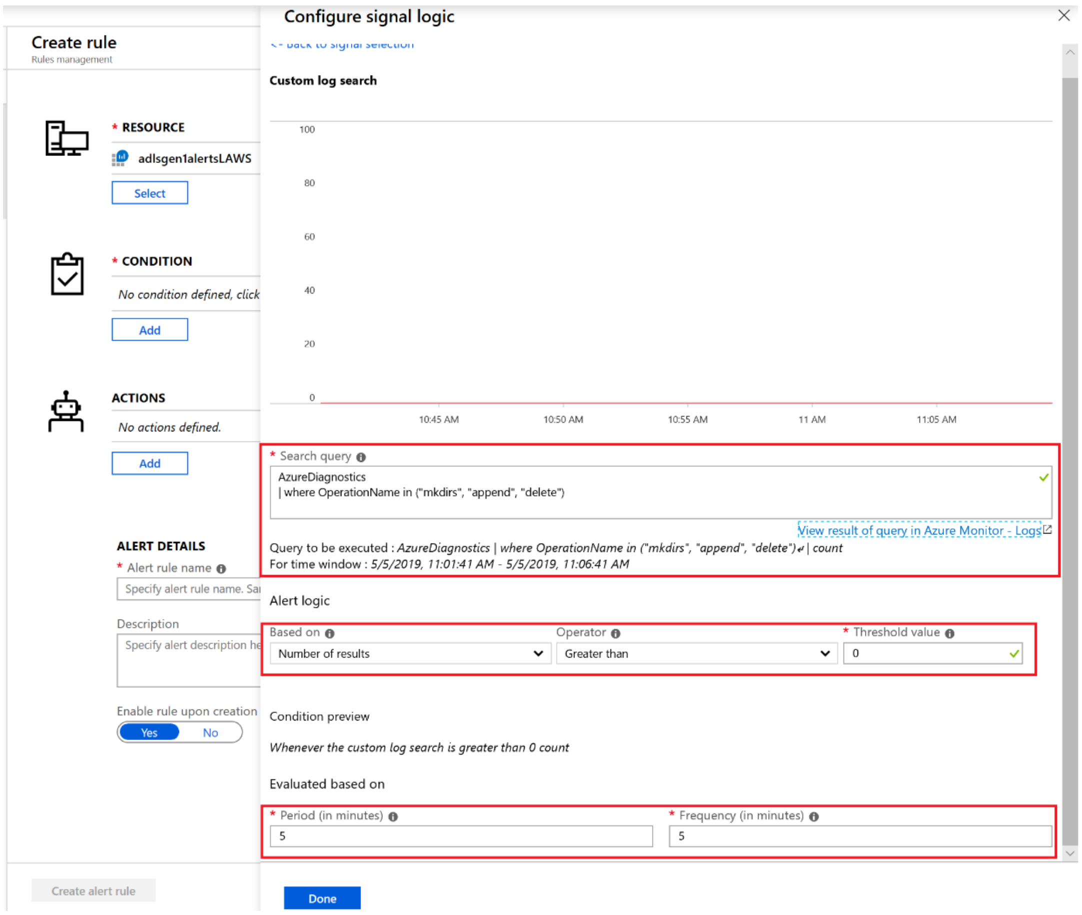
Signal type: All

Monitor service: All

SIGNAL NAME	SIGNAL TYPE	MONITOR SERVICE
Custom log search	Log	Log analytics
All Events	Log(Saved Query)	Log analytics
Count of Events containing the word "started" grouped by EventID	Log(Saved Query)	Log analytics
Count of Events grouped by Event Log	Log(Saved Query)	Log analytics
Count of Events grouped by Event Source	Log(Saved Query)	Log analytics

Figure 8: Custom Log Search Condition

Let us now look at the details of this search criteria:


Configure signal logic

Custom log search

Search query:

```
AzureDiagnostics
| where OperationName in ("mkdirs", "append", "delete")
```

Query to be executed : *AzureDiagnostics | where OperationName in ("mkdirs", "append", "delete") | count*

For time window : *5/5/2019, 11:01:41 AM - 5/5/2019, 11:06:41 AM*

Alert logic

Based on: Number of results
Operator: Greater than
Threshold value: 0

Condition preview

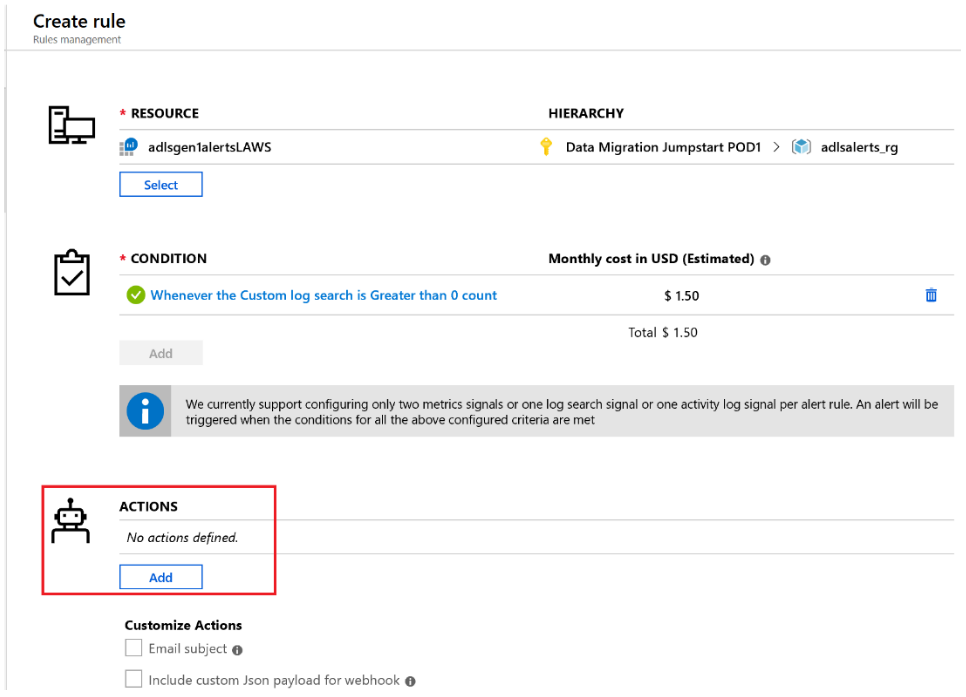
Whenever the custom log search is greater than 0 count

Evaluated based on

Period (in minutes): 5
Frequency (in minutes): 5

Figure 9: Log Analytics - Alert Logic Configuration

It is important to understand, the minimum latency to receive and process alerts is 5 minutes, for up to 24 hours.


Create rule
Rules management

* RESOURCE | HIERARCHY
adlsgen1alertsLAWS | Data Migration Jumpstart POD1 > adlsalerts_rg
Select

* CONDITION	Monthly cost in USD (Estimated)
Whenever the Custom log search is Greater than 0 count	$ 1.50
	Total $ 1.50

Add

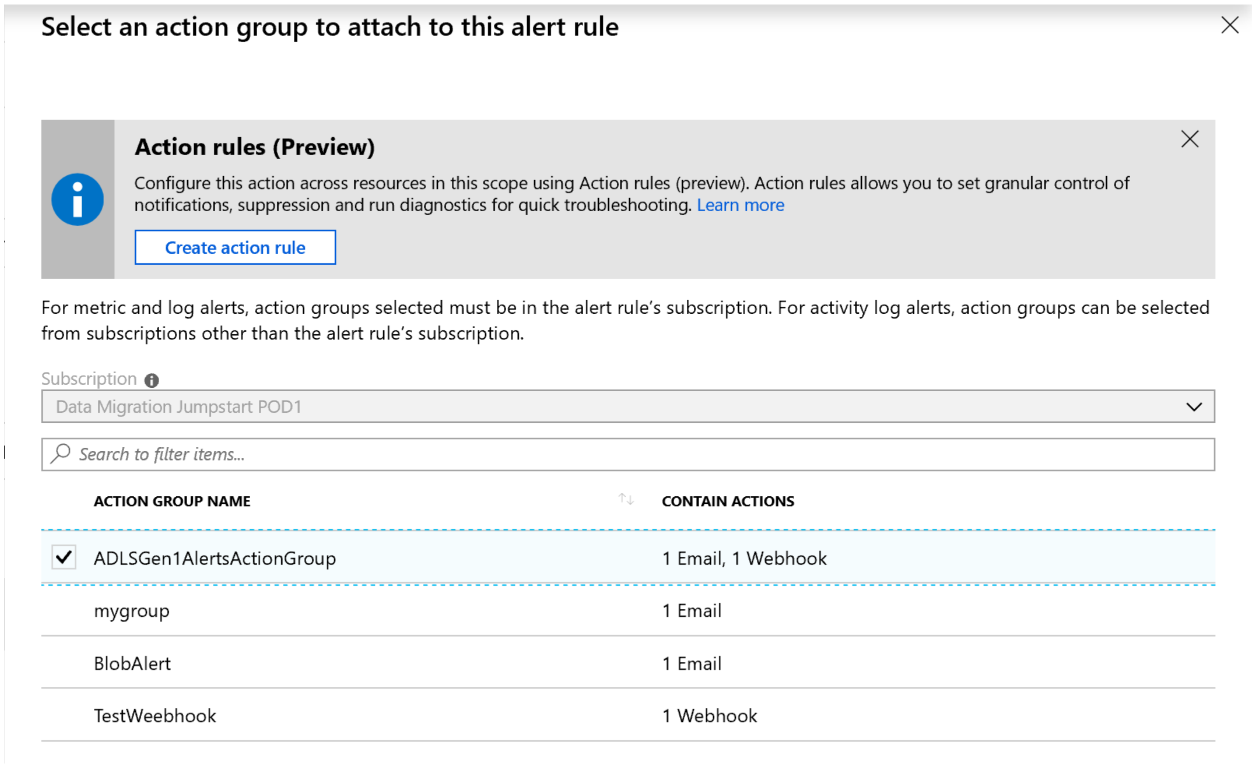
We currently support configuring only two metrics signals or one log search signal or one activity log signal per alert rule. An alert will be triggered when the conditions for all the above configured criteria are met

ACTIONS
No actions defined.
Add

Customize Actions
☐ Email subject
☐ Include custom Json payload for webhook

Figure 10: Action Group Assignment

When you click on Add above, it will open an interface to select the sample Action Group that we have defined in prior sections. Here's the visual:


Select an action group to attach to this alert rule

Action rules (Preview)
Configure this action across resources in this scope using Action rules (preview). Action rules allows you to set granular control of notifications, suppression and run diagnostics for quick troubleshooting. Learn more
Create action rule

For metric and log alerts, action groups selected must be in the alert rule's subscription. For activity log alerts, action groups can be selected from subscriptions other than the alert rule's subscription.

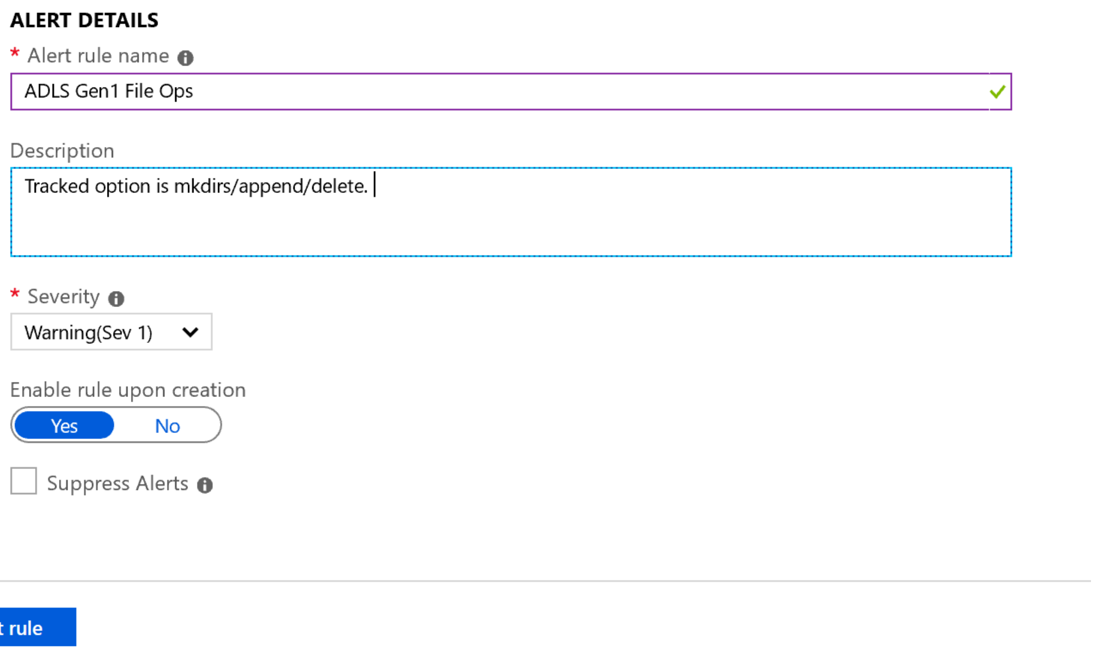
Subscription
Data Migration Jumpstart POD1

Search to filter items...

	ACTION GROUP NAME	CONTAIN ACTIONS
✓	ADLSGen1AlertsActionGroup	1 Email, 1 Webhook
	mygroup	1 Email
	BlobAlert	1 Email
	TestWeebhook	1 Webhook

Figure 11:Action Group Selection

Once you have added the Action Group under the Actions option of the Create Rule interface, the next step is to add Alert details on the same interface. This will allow you complete the last step in the sequence and submit to create the alert rule.



ALERT DETAILS

* Alert rule name: ADLS Gen1 File Ops

Description: Tracked option is mkdirs/append/delete.

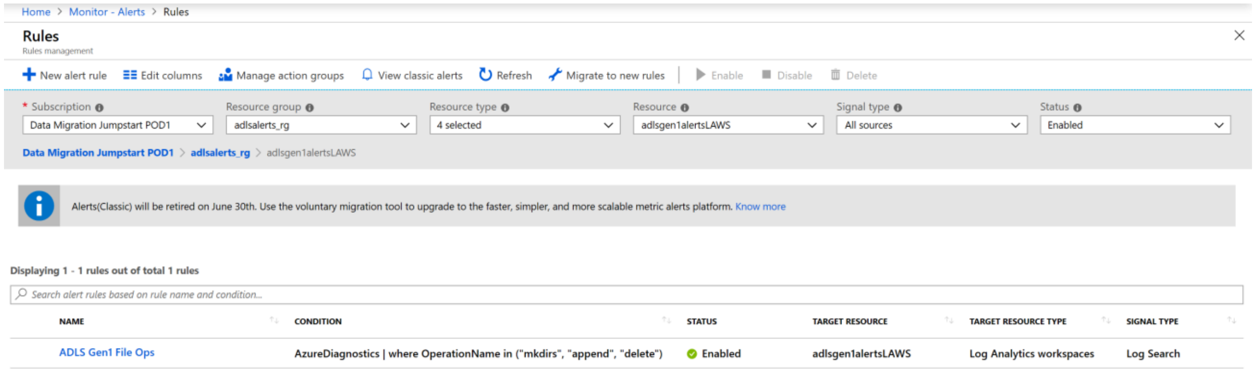
* Severity: Warning(Sev 1)

Enable rule upon creation: [✓ Yes] No

☐ Suppress Alerts

Figure 12: Create Rule/Alert Details

You can verify the newly defined Alert rule, by navigating to the Home → Monitor – Alerts tab and by selecting the tab Manage Rules:



* Subscription: Data Migration Jumpstart POD1
Resource group: adlsalerts_rg
Resource type: 4 selected
Resource: adlsgen1alertsLAWS
Signal type: All sources
Status: Enabled

Data Migration Jumpstart POD1 > adlsalerts_rg > adlsgen1alertsLAWS

Alerts(Classic) will be retired on June 30th. Use the voluntary migration tool to upgrade to the faster, simpler, and more scalable metric alerts platform. Know more

Displaying 1 - 1 rules out of total 1 rules

Search alert rules based on rule name and condition...

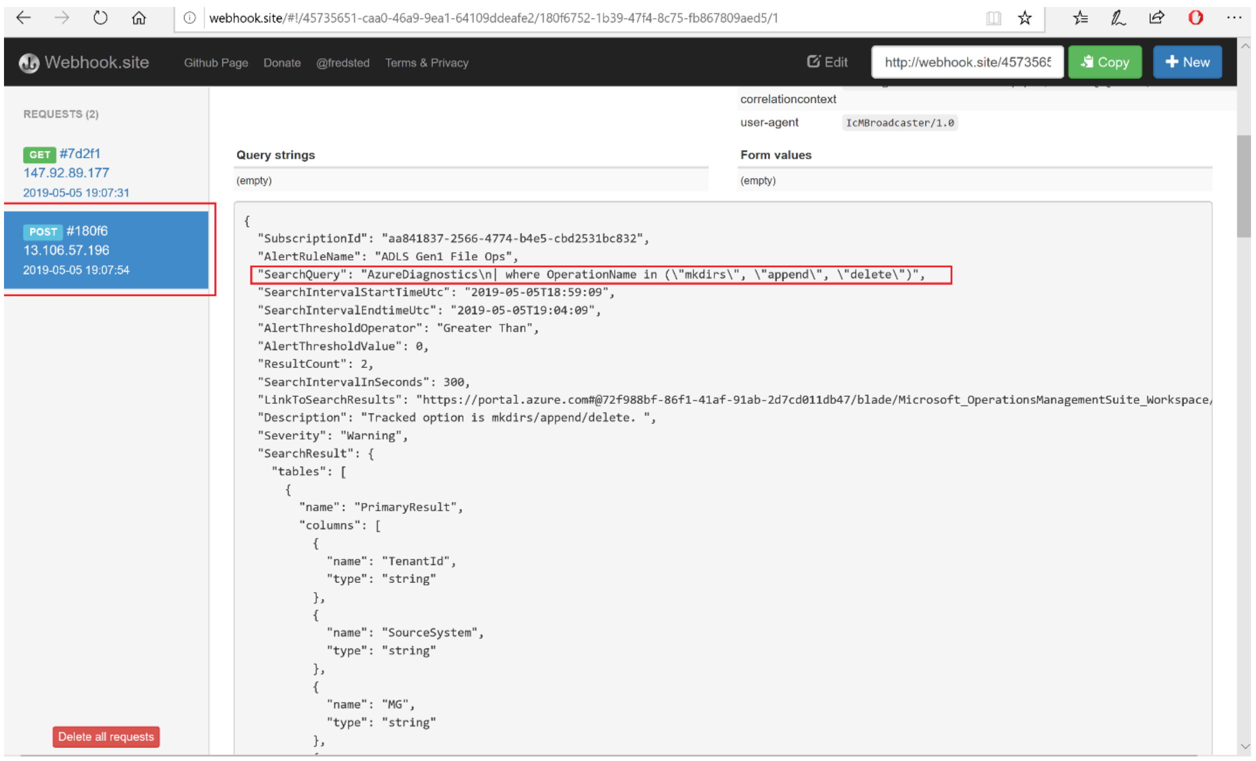
NAME	CONDITION	STATUS	TARGET RESOURCE	TARGET RESOURCE TYPE	SIGNAL TYPE
ADLS Gen1 File Ops	AzureDiagnostics | where OperationName in ("mkdirs", "append", "delete")	✓ Enabled	adlsgen1alertsLAWS	Log Analytics workspaces	Log Search

Figure 13: Manage Rules/Verify New Rule

3.4 Verify the Alerts

Recommend that you wait about 5 minutes, to start observing the action outcomes. Remember the SLA set above to 5 minutes minimum latency defined with an event duration of since last 5 minutes up to a maximum of past 24 hours. To trigger an alert you can use Azure Data Explorer client and create few folders under the same subscription POD and the ADLS Gen1 resource that is part of the resource group where you have defined the alerts. The ADLS resource must be same as configured to be monitored by the alert's action group.

Here's the sample of the webhook result (received as a POST request):



correlationcontext
user-agent IcMBroadcaster/1.0

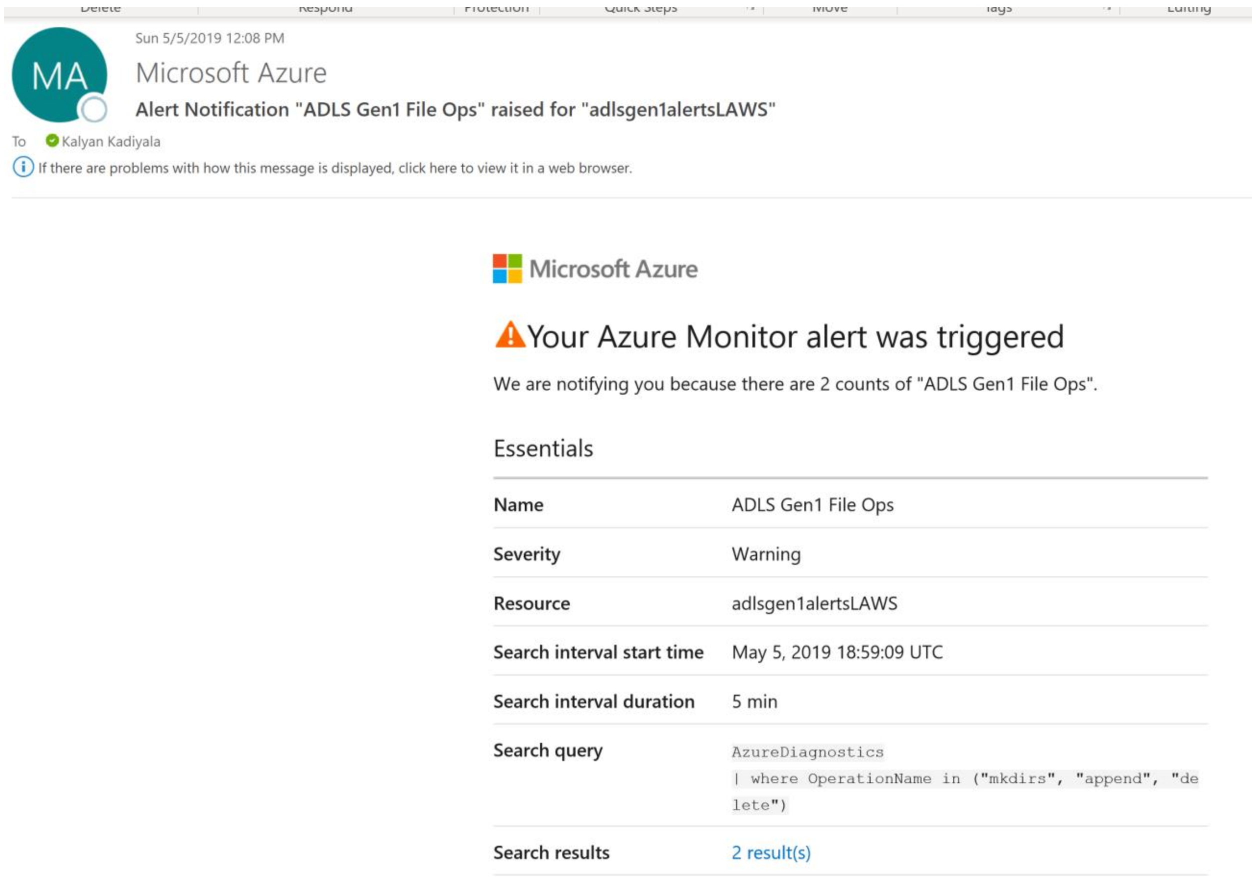
Query strings
(empty)

Form values
(empty)

```
{
  "SubscriptionId": "aa841837-2566-4774-b4e5-cbd2531bc832",
  "AlertRuleName": "ADLS Gen1 File Ops",
  "SearchQuery": "AzureDiagnostics\n| where OperationName in (\"mkdirs\", \"append\", \"delete\")",
  "SearchIntervalStartTimeUtc": "2019-05-05T18:59:09",
  "SearchIntervalEndtimeUtc": "2019-05-05T19:04:09",
  "AlertThresholdOperator": "Greater Than",
  "AlertThresholdValue": 0,
  "ResultCount": 2,
  "SearchIntervalInSeconds": 300,
  "LinkToSearchResults": "https://portal.azure.com#@72f988bf-86f1-41af-91ab-2d7cd011db47/blade/Microsoft_OperationsManagementSuite_Workspace/
  "Description": "Tracked option is mkdirs/append/delete. ",
  "Severity": "Warning",
  "SearchResult": {
    "tables": [
      {
        "name": "PrimaryResult",
        "columns": [
          {
            "name": "TenantId",
            "type": "string"
          },
          {
            "name": "SourceSystem",
            "type": "string"
          },
          {
            "name": "MG",
            "type": "string"
          },
```

Figure 14: Webhook Alert Sample

Here is the email sample of the same alert:


Sun 5/5/2019 12:08 PM

Microsoft Azure

Alert Notification "ADLS Gen1 File Ops" raised for "adlsgen1alertsLAWS"

To Kalyan Kadiyala

If there are problems with how this message is displayed, click here to view it in a web browser.

Microsoft Azure

Your Azure Monitor alert was triggered

We are notifying you because there are 2 counts of "ADLS Gen1 File Ops".

Essentials

Name	ADLS Gen1 File Ops
Severity	Warning
Resource	adlsgen1alertsLAWS
Search interval start time	May 5, 2019 18:59:09 UTC
Search interval duration	5 min
Search query	`AzureDiagnostics | where OperationName in ("mkdirs", "append", "delete")`
Search results	2 result(s)

Figure 15: Email Alert Sample

3.5 Log Analytics Approach – Things to remember!

As you observe the approach offers a pure configuration-based approach to define notification alerts for corresponding ADLS operations – mkdir, append and delete. The query interface supports a wide variety of query language features. Here is the link to learn more! The only caveat you need to assert for your applicability is – is the minimum lag duration of 5 minutes is agreeable for your notification alerts use case? Else, this is a great way to setup and receive Notification alerts using one of the available channels – Email, Webhook or SMS alerts.

4 ADLS Event Notification Alerts using Event Hub

The next solution approach that we will review is the ability to use Event Hub to receive notification alerts for select events generated by activities on your configured ADLS Gen1 resource. Here’s the empirical view of the flow:

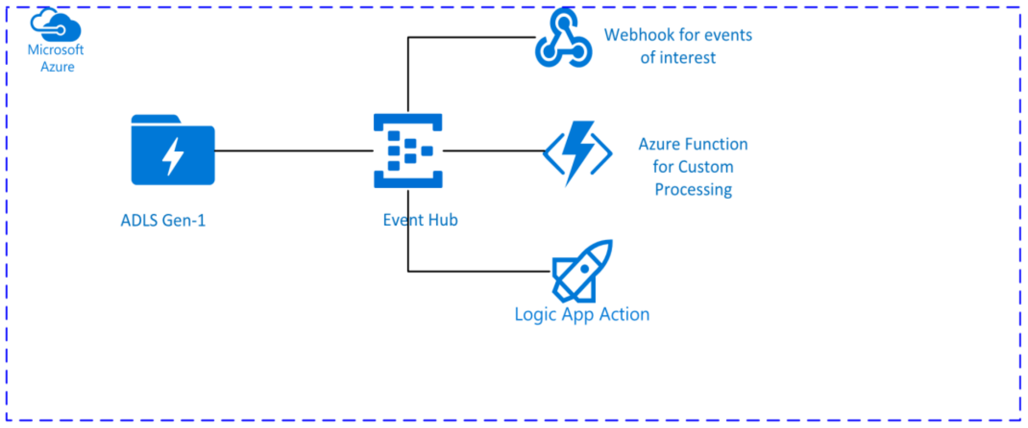

Microsoft Azure
ADLS Gen-1
Event Hub
Webhook for events of interest
Azure Function for Custom Processing
Logic App Action

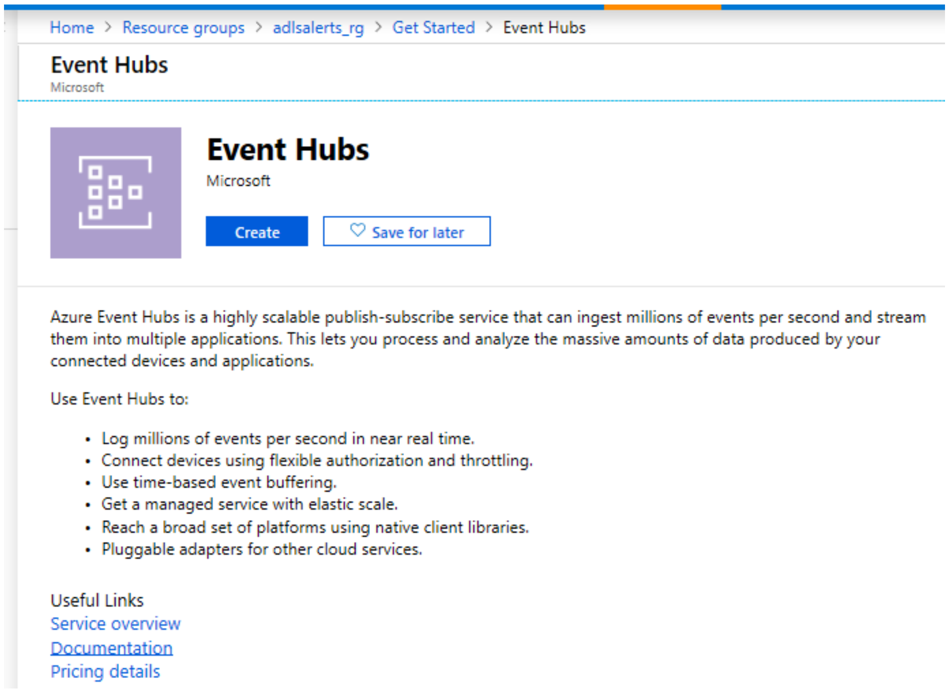
Figure 16: Event Notification Alerts using Event Hub

Briefly, the steps involved in this approach include:

- Configure ADLS Diagnostics to point to Event Hub as target destination to log events.
    - Pre-requisite step here is to define the Event Hub namespace.
- Create Azure Function to ingest and filter ADLS logs.
- (Optional) Create a Logic App with Event Hub as Trigger.

4.1 Configure Event Hub namespace

In this section we will see the approach to configure an Event Hub to capture events from ADLS resource. For the context, let us name our Event Hub namespace as *adlsgen1alerts2ehns*. From your corresponding Resource Group (for example – *adlsalerts_rg*) blade i.e., pane where resources of a Resource Group can be viewed, click on Add a resource icon on the top navigation pane.


Home > Resource groups > adlsalerts_rg > Get Started > Event Hubs
Event Hubs
Microsoft
Event Hubs
Microsoft
Create
Save for later

Azure Event Hubs is a highly scalable publish-subscribe service that can ingest millions of events per second and stream them into multiple applications. This lets you process and analyze the massive amounts of data produced by your connected devices and applications.

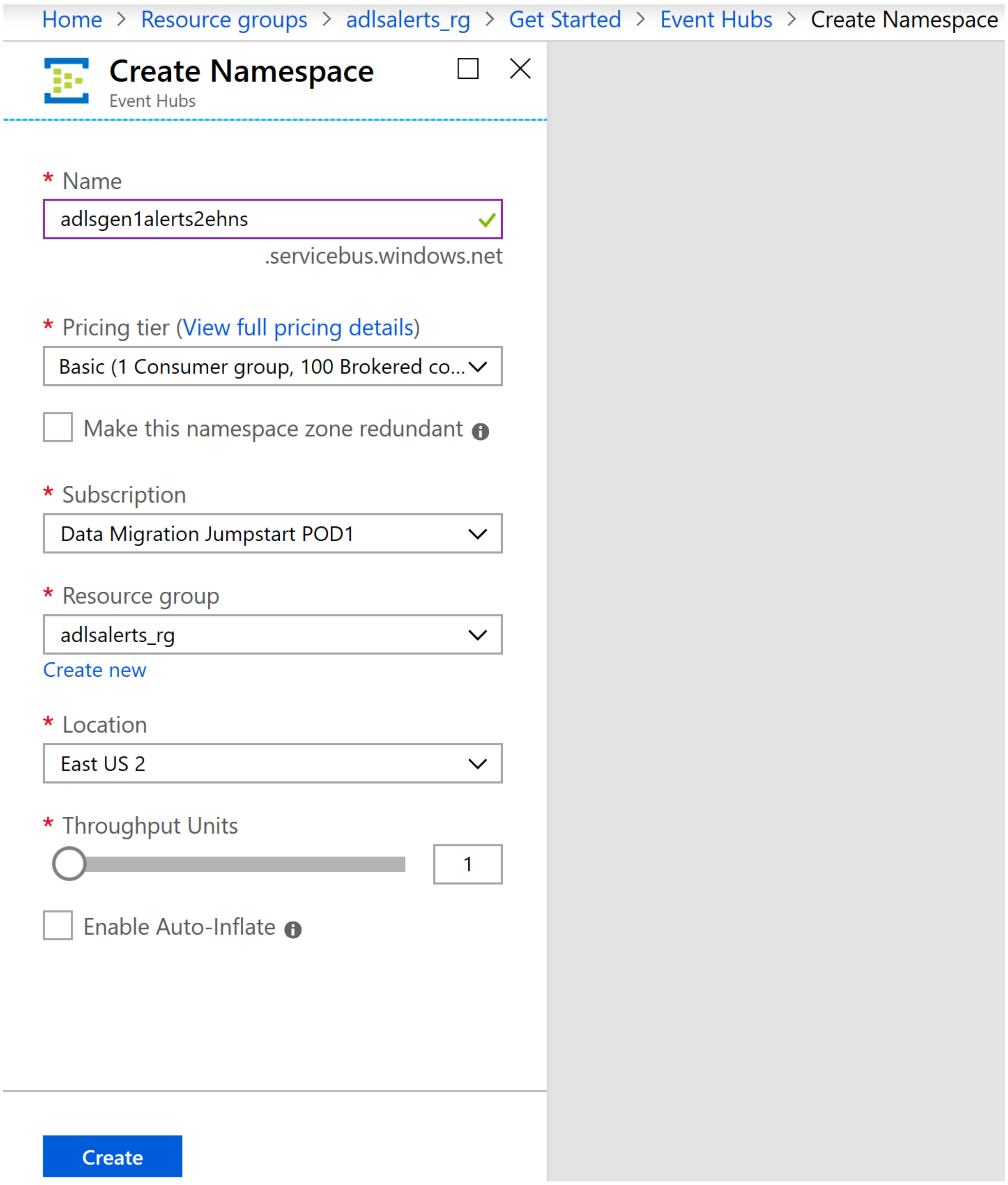
Use Event Hubs to:

- Log millions of events per second in near real time.
- Connect devices using flexible authorization and throttling.
- Use time-based event buffering.
- Get a managed service with elastic scale.
- Reach a broad set of platforms using native client libraries.
- Pluggable adapters for other cloud services.

Useful Links
Service overview
Documentation
Pricing details

Figure 17: Event Hubs

Click on create action button in the interface, like the one shown in the above visual. Fill in the necessary configuration as in the visual below:


Home > Resource groups > adlsalerts_rg > Get Started > Event Hubs > Create Namespace

Create Namespace
Event Hubs

* Name
adlsgen1alerts2ehns
.servicebus.windows.net

* Pricing tier (View full pricing details)
Basic (1 Consumer group, 100 Brokered co...

☐ Make this namespace zone redundant

* Subscription
Data Migration Jumpstart POD1

* Resource group
adlsalerts_rg
Create new

* Location
East US 2

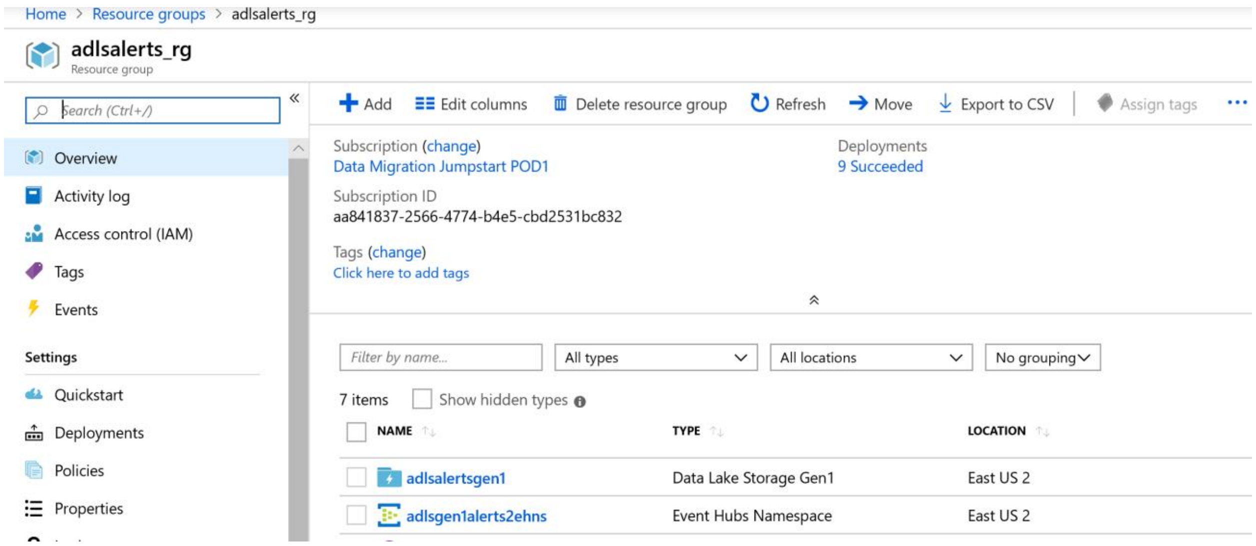
* Throughput Units
1

☐ Enable Auto-Inflate

Create

Figure 18: Create Event Hub Namespace

Upon successful creation of the event hub, you can check the event hub resource from the following Resource Group overview page as in the following visual:



Subscription (change)
Data Migration Jumpstart POD1

Subscription ID
aa841837-2566-4774-b4e5-cbd2531bc832

Tags (change)
Click here to add tags

Deployments
9 Succeeded

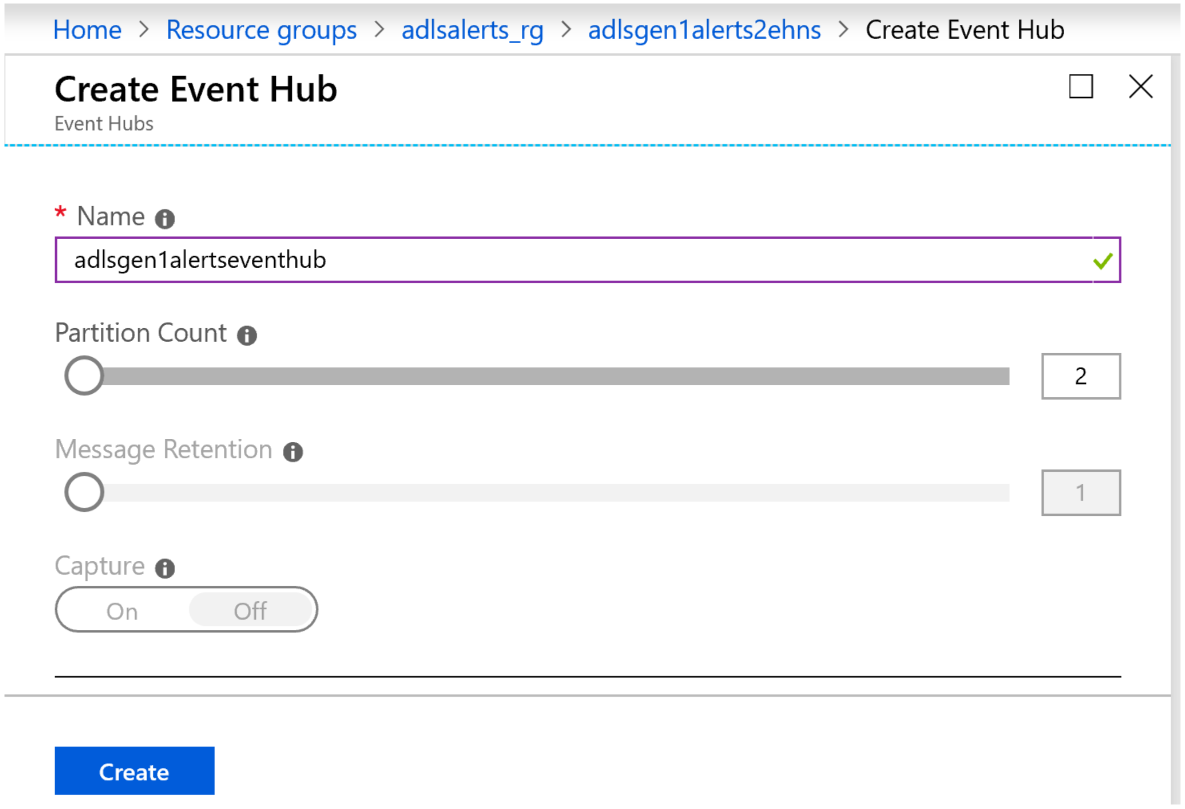
Filter by name... | All types | All locations | No grouping

7 items Show hidden types

NAME	TYPE	LOCATION
adlsalertsgen1	Data Lake Storage Gen1	East US 2
adlsgen1alerts2ehns	Event Hubs Namespace	East US 2

Figure 19: Resources Overview - Event Hub Namespace Listing

Once you have the Event Hub Namespace defined, the next step is to define event hub where our events from the ADLS gen1 can be captured. Here's the reference visual to help you define an Event Hub, under this newly defined Event Hub Namespace:



Home > Resource groups > adlsalerts_rg > adlsgen1alerts2ehns > Create Event Hub
Create Event Hub
Event Hubs
Create

* Name: adlsgen1alertseventhub

Partition Count: 2

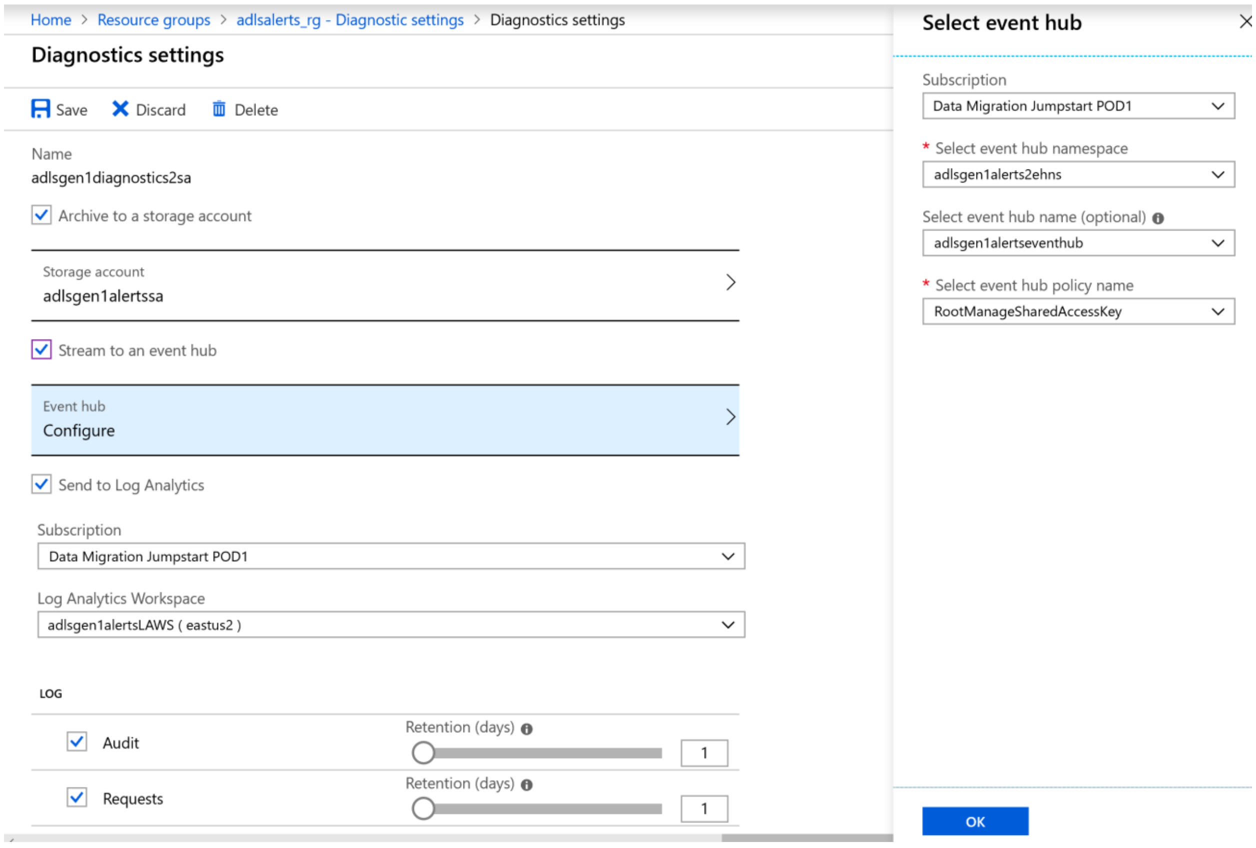
Message Retention: 1

Capture: On [✓ Off]

Figure 20: Create Event Hub

4.2 Configure ADLS Diagnostics

Now that we have an Event Hub name space and Event Hub defined, let's configure diagnostics to also stream events to the event hub. Here's the visual for a quick reference:


Home > Resource groups > adlsalerts_rg - Diagnostic settings > Diagnostics settings
Save
Discard
Delete
OK

Diagnostics settings

Name
adlsgen1diagnostics2sa

☑ Archive to a storage account

Storage account
adlsgen1alertssa

☑ Stream to an event hub

Event hub
Configure

☑ Send to Log Analytics

Subscription
Data Migration Jumpstart POD1

Log Analytics Workspace
adlsgen1alertsLAWS (eastus2)

LOG

☑ Audit	Retention (days)	1
☑ Requests	Retention (days)	1

Select event hub

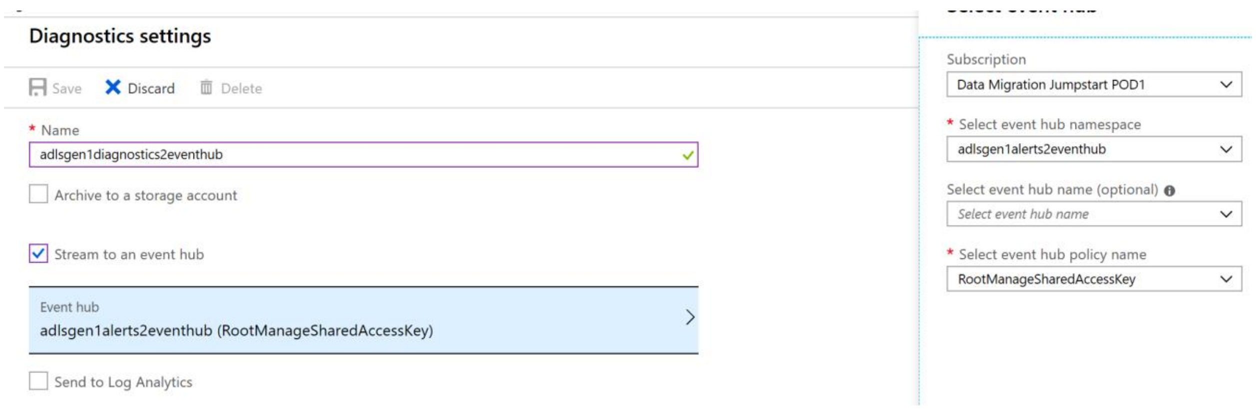
Subscription
Data Migration Jumpstart POD1

* Select event hub namespace
adlsgen1alerts2ehns

Select event hub name (optional)
adlsgen1alertseventhub

* Select event hub policy name
RootManageSharedAccessKey

Now that we have an Event Hub namespace defined, let's proceed to define a diagnostic setting to point ADLS events to the newly created event hub. Navigate the Azure Portal Path - Home → Monitor – Diagnostics Settings and add a new diagnostic setting. Here's the visual for reference:



Diagnostics settings

* Name
adlsgen1diagnostics2eventhub

☐ Archive to a storage account

☑ Stream to an event hub

Event hub
adlsgen1alerts2eventhub (RootManageSharedAccessKey)

☐ Send to Log Analytics

Subscription
Data Migration Jumpstart POD1

* Select event hub namespace
adlsgen1alerts2eventhub

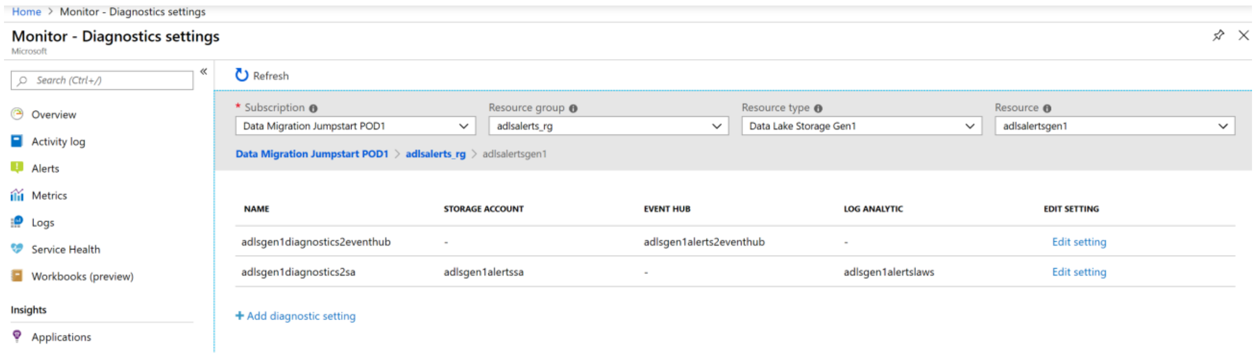
Select event hub name (optional)
Select event hub name

* Select event hub policy name
RootManageSharedAccessKey

Figure 21: Selecting Event Hub Namespace

As you observe from the visual above, this time we are selecting the option to stream events to an event hub. Once you have selected the corresponding event hub, on the diagnostic settings also select the log options – Requests, and for metrics select All Metrics. Hit save on the top navigation pane (this is enabled once you select what to capture i.e., logs and metrics).

Here's the overview page where both the diagnostic settings (log analytics & event hub) are listed:



Subscription: Data Migration Jumpstart POD1
Resource group: adlsalerts_rg
Resource type: Data Lake Storage Gen1
Resource: adlsalertsgen1

Data Migration Jumpstart POD1 > adlsalerts_rg > adlsalertsgen1

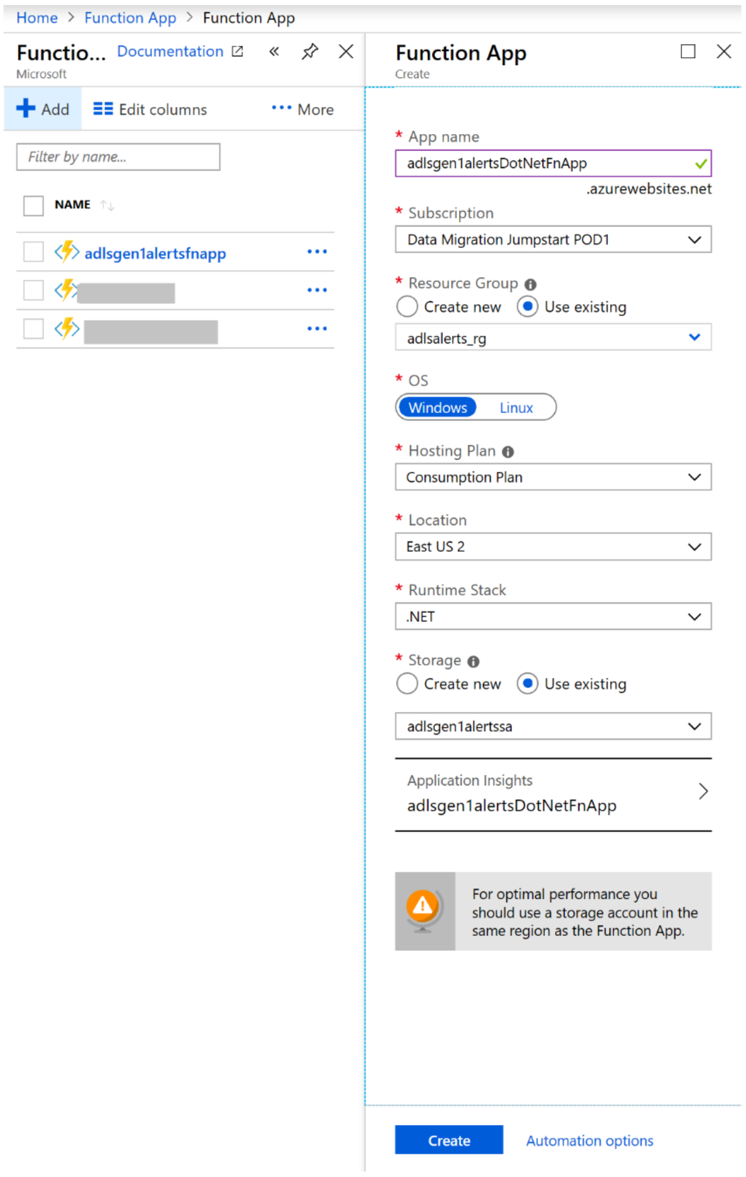
NAME	STORAGE ACCOUNT	EVENT HUB	LOG ANALYTIC	EDIT SETTING
adlsgen1diagnostics2eventhub	-	adlsgen1alerts2eventhub	-	Edit setting
adlsgen1diagnostics2sa	adlsgen1alertssa	-	adlsgen1alertslaws	Edit setting

Figure 22: Diagnostic Settings Overview

4.3 Define Azure Function to filter ADLS events

Once we have configured diagnostics to log events to the event hub, we can proceed to defining consumer applications to filter and raise alerts using Microsoft's Event Hub SDK (custom application). The other option here is to define Logic Apps. In this section, we will walk through the necessary steps to define an Azure Function as a trigger that is applied whenever an event is logged into the Event Hub.

The very first step is to define a Function App. This can be accomplished by choosing the Function App tab under the left navigation pane – favorites tab, in your Azure Portal.



* App name: adlsgen1alertsDotNetFnApp .azurewebsites.net

* Subscription: Data Migration Jumpstart POD1

* Resource Group: Create new [✓ Use existing]

adlsalerts_rg

* OS: [✓ Windows] Linux

* Hosting Plan: Consumption Plan

* Location: East US 2

* Runtime Stack: .NET

* Storage: Create new [✓ Use existing]

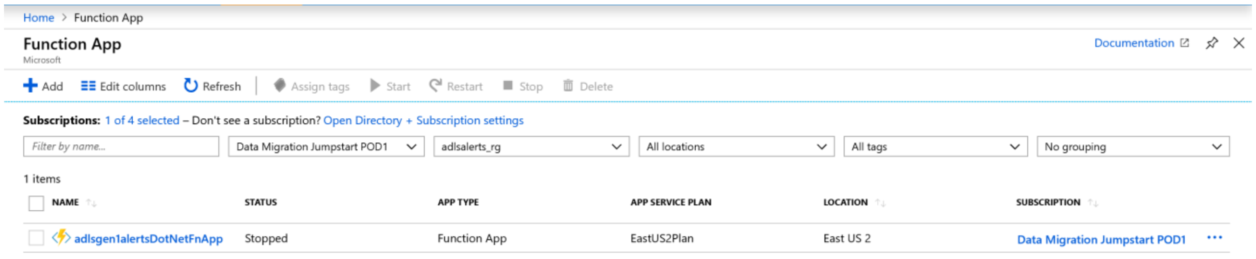
adlsgen1alertssa

Application Insights: adlsgen1alertsDotNetFnApp

For optimal performance you should use a storage account in the same region as the Function App.

Figure 23: Create Function App

You can verify the created resource – function app, from the function apps overview blade (view):



Subscriptions: 1 of 4 selected – Don't see a subscription? Open Directory + Subscription settings

Filter by name... | Data Migration Jumpstart POD1 | adlsalerts_rg | All locations | All tags | No grouping

1 items

NAME	STATUS	APP TYPE	APP SERVICE PLAN	LOCATION	SUBSCRIPTION
adlsgen1alertsDotNetFnApp	Stopped	Function App	EastUS2Plan	East US 2	Data Migration Jumpstart POD1

Figure 24: Existing Function Apps Overview

Subsequently will define an in-portal function using one of the available templates as shown in the following visual:

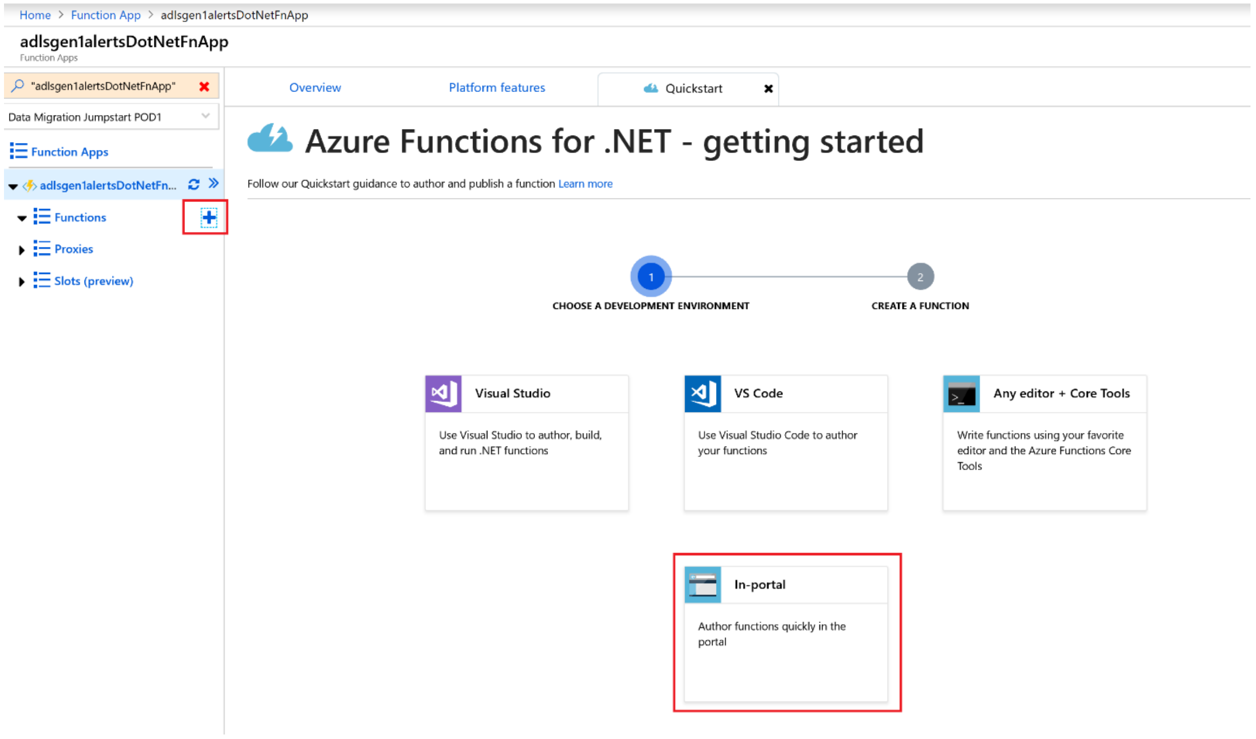

Home > Function App > adlsgen1alertsDotNetFnApp
adlsgen1alertsDotNetFnApp
Function Apps
"adlsgen1alertsDotNetFnApp"
Data Migration Jumpstart POD1
Function Apps
adlsgen1alertsDotNetFn...
Functions
Proxies
Slots (preview)
Overview
Platform features
Quickstart
Azure Functions for .NET - getting started
Follow our Quickstart guidance to author and publish a function Learn more
1
CHOOSE A DEVELOPMENT ENVIRONMENT
2
CREATE A FUNCTION
Visual Studio
Use Visual Studio to author, build, and run .NET functions
VS Code
Use Visual Studio Code to author your functions
Any editor + Core Tools
Write functions using your favorite editor and the Azure Functions Core Tools
In-portal
Author functions quickly in the portal

Figure 25: Define Azure Function (In-portal template)

Then, select more templates…

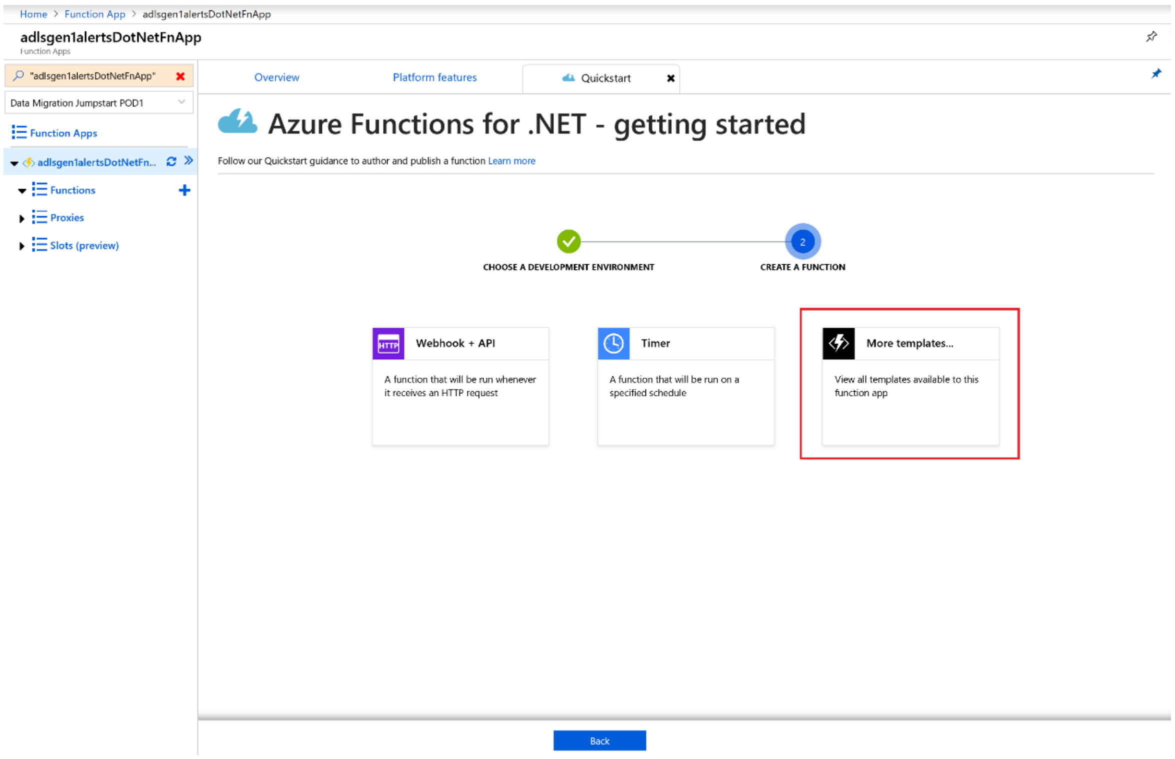

Home > Function App > adlsgen1alertsDotNetFnApp
adlsgen1alertsDotNetFnApp
Function Apps
"adlsgen1alertsDotNetFnApp"
Data Migration Jumpstart POD1
Function Apps
adlsgen1alertsDotNetFn...
Functions
Proxies
Slots (preview)
Overview
Platform features
Quickstart
Azure Functions for .NET - getting started
Follow our Quickstart guidance to author and publish a function Learn more
CHOOSE A DEVELOPMENT ENVIRONMENT
CREATE A FUNCTION
Webhook + API
A function that will be run whenever it receives an HTTP request
Timer
A function that will be run on a specified schedule
More templates...
View all templates available to this function app
Back

Figure 26: Azure Function - More Templates...

The next step is to search for Azure Event Hub Trigger template as shown below:

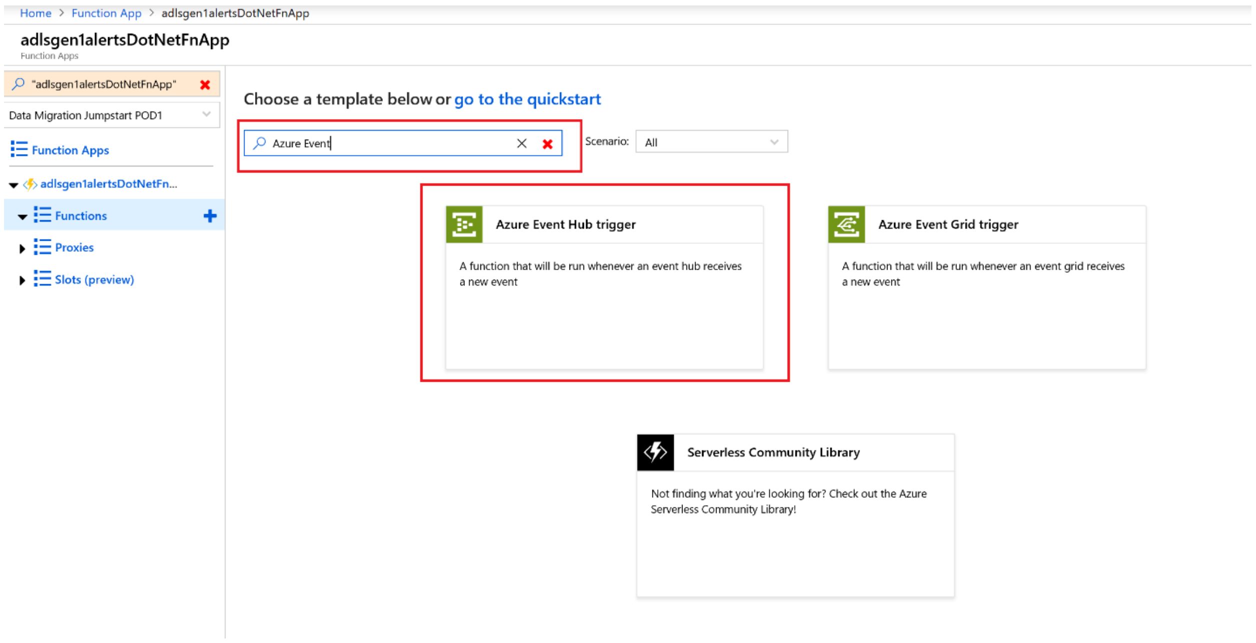
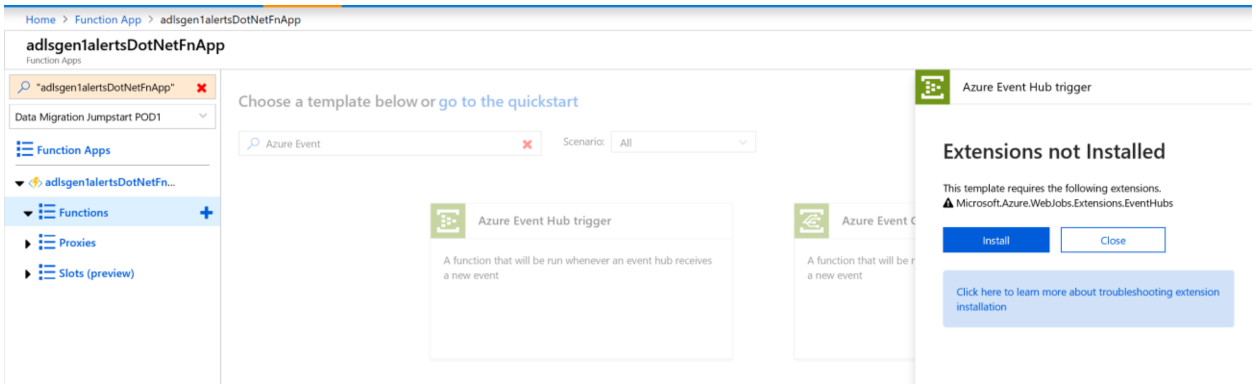

Home > Function App > adlsgen1alertsDotNetFnApp
adlsgen1alertsDotNetFnApp
Function Apps
"adlsgen1alertsDotNetFnApp"
Data Migration Jumpstart POD1
Function Apps
adlsgen1alertsDotNetFn...
Functions
Proxies
Slots (preview)
Choose a template below or go to the quickstart
Azure Event
Scenario:
All
Azure Event Hub trigger
A function that will be run whenever an event hub receives a new event
Azure Event Grid trigger
A function that will be run whenever an event grid receives a new event
Serverless Community Library
Not finding what you're looking for? Check out the Azure Serverless Community Library!

Figure 27: Azure Functions - Azure Event Hub Trigger

Install the missing plugin if prompted,

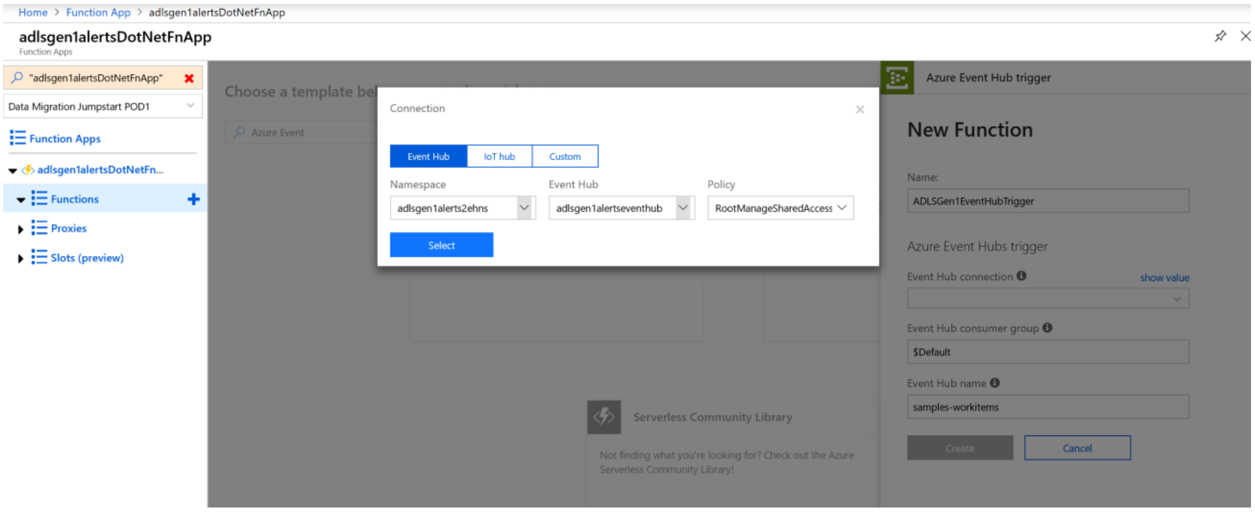

Extensions not Installed

This template requires the following extensions.
⚠ Microsoft.Azure.WebJobs.Extensions.EventHubs

Home > Function App > adlsgen1alertsDotNetFnApp
adlsgen1alertsDotNetFnApp
Function Apps
Choose a template below or go to the quickstart
Azure Event Hub trigger
A function that will be run whenever an event hub receives a new event
Install
Close
Click here to learn more about troubleshooting extension installation

Figure 28: MicrosoftAzure.WebJobs.Extensions.EventHubs Extension

Select the available event hub namespace, event hub and access policy from your subscription.


Connection

Namespace: adlsgen1alerts2ehns
Event Hub: adlsgen1alertseventhub
Policy: RootManageSharedAccess

New Function

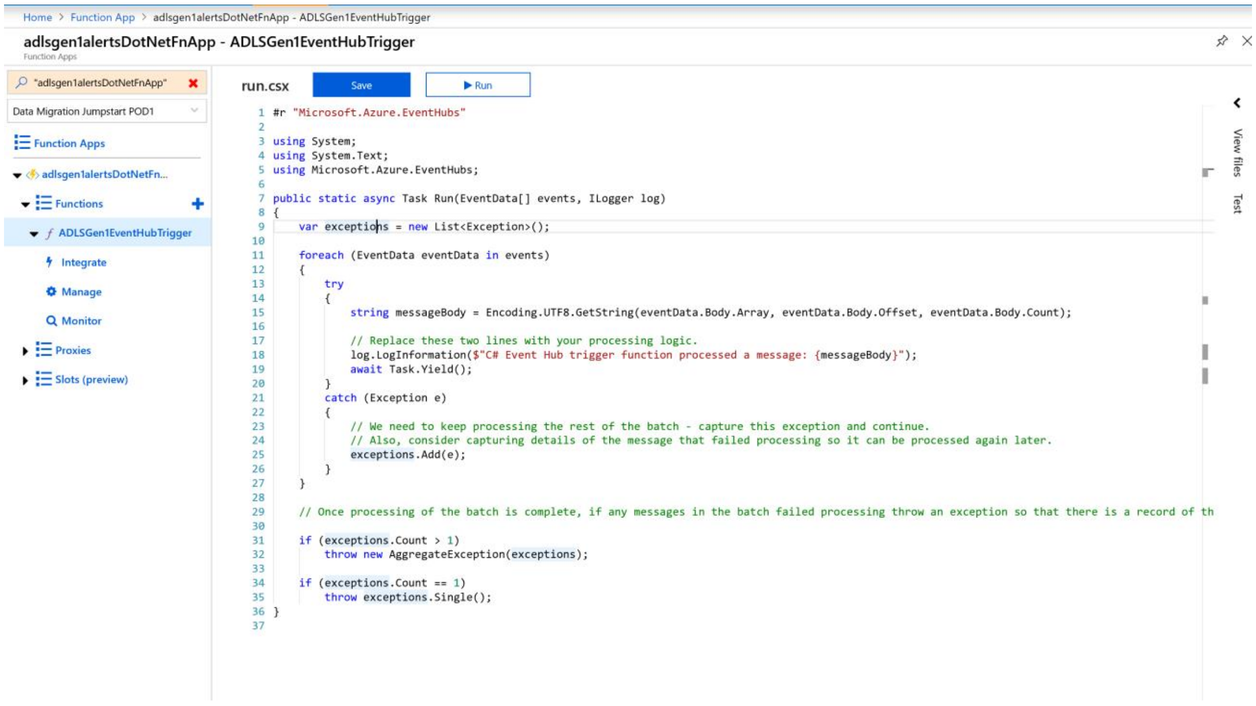
Name: ADLSGen1EventHubTrigger

Azure Event Hubs trigger

Event Hub consumer group: $Default

Event Hub name: samples-workitems

Click create and a sample code will be generated by the designer as shown below.


Home > Function App > adlsgen1alertsDotNetFnApp - ADLSGen1EventHubTrigger
adlsgen1alertsDotNetFnApp - ADLSGen1EventHubTrigger
Function Apps
"adlsgen1alertsDotNetFnApp"
Data Migration Jumpstart POD1
Function Apps
adlsgen1alertsDotNetFn...
Functions
ADLSGen1EventHubTrigger
Integrate
Manage
Monitor
Proxies
Slots (preview)
run.csx
Save
Run
View files
Test

```
#r "Microsoft.Azure.EventHubs"

using System;
using System.Text;
using Microsoft.Azure.EventHubs;

public static async Task Run(EventData[] events, ILogger log)
{
    var exceptions = new List<Exception>();

    foreach (EventData eventData in events)
    {
        try
        {
            string messageBody = Encoding.UTF8.GetString(eventData.Body.Array, eventData.Body.Offset, eventData.Body.Count);

            // Replace these two lines with your processing logic.
            log.LogInformation($"C# Event Hub trigger function processed a message: {messageBody}");
            await Task.Yield();
        }
        catch (Exception e)
        {
            // We need to keep processing the rest of the batch - capture this exception and continue.
            // Also, consider capturing details of the message that failed processing so it can be processed again later.
            exceptions.Add(e);
        }
    }

    // Once processing of the batch is complete, if any messages in the batch failed processing throw an exception so that there is a record of th

    if (exceptions.Count > 1)
        throw new AggregateException(exceptions);

    if (exceptions.Count == 1)
        throw exceptions.Single();
}

```

Figure 29: Sample Code generated by Designer

Optionally , we can also enhance the sample code, for example to send the events to downstream applications to consume the events.

```
private static void SendEventDownStream(string message) {
    var httpWebRequest =
          (HttpWebRequest)WebRequest
            .Create("https://webhook.site/a42e189a-0854-40a4-b629-dc838e750088");
    httpWebRequest.ContentType = "application/json";
    httpWebRequest.Method = "POST";

    using (var streamWriter = new StreamWriter(httpWebRequest.GetRequestStream())){
        streamWriter.Write(message);
        streamWriter.Flush();
        streamWriter.Close();
    }

    var httpResponse = (HttpWebResponse)httpWebRequest.GetResponse();
    using (var streamReader = new StreamReader(httpResponse.GetResponseStream())){
        var result = streamReader.ReadToEnd();
    }
}
```

Figure 30: Generated Code customization sample

5 ADLS Event Notification Alerts using Event Grid

The third and last approach discussed in this white paper is the approach where Event Grid is used to setup event notification alerts. This is more of a futuristic feature, where events from Data lake can be directly ingested into Event Grid. Here is the empirical view of the event flows:

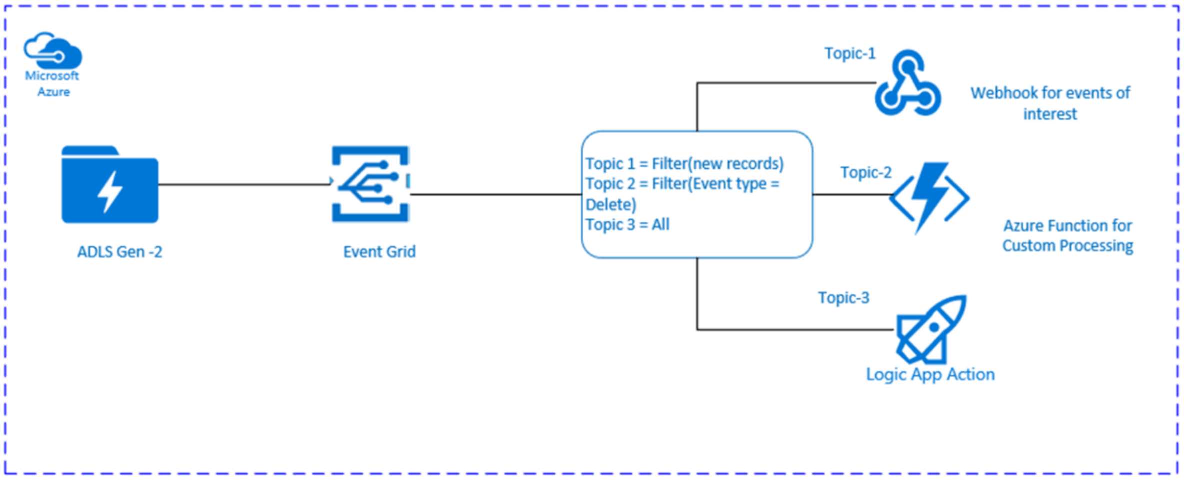

Microsoft Azure
ADLS Gen -2
Event Grid
Topic 1 = Filter(new records)
Topic 2 = Filter(Event type = Delete)
Topic 3 = All
Topic-1
Webhook for events of interest
Topic-2
Azure Function for Custom Processing
Topic-3
Logic App Action

Figure 31: Event Notification Alerts using Event Grid

5 Appendices

5.1 Appendix - External References

- Struggling to get insights for your Azure Data Lake Store? Azure Log Analytics can help!
- Azure Log Analytics
- Azure Monitor
- Azure Event Hubs
- Azure Logic Apps

6 Feedback and suggestions

If you have feedback or suggestions for improving this data migration asset, please contact the Data SQL Ninja Team (askdmjfordmtools@microsoft.com). Thanks for your support!

Note: For additional information about migrating various source databases to Azure, see the Azure Database Migration Guide.