Contents

Workshop Summary

The first workshop, "Workshop 1", is intended to produce a data pipeline that ingests a publicly available COVID-19 dataset, via ADF, to ADLS. This dataset is then loaded into ADX by fully replacing the existing data during each iteration.

Azure Services

The following Azure services are introduced during this workshop to the architecture:

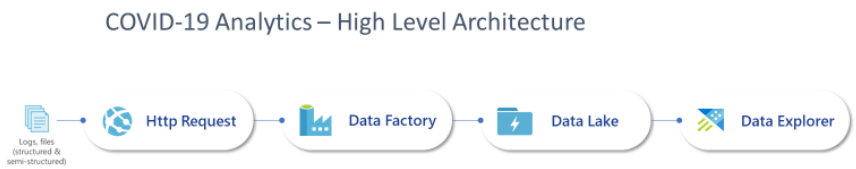
Azure Data Factory (ADF) [link]

Azure Data Lake (ADLS) [link]

Azure Data Explorer (ADX) [link]

Architecture

The diagram “COVID-19 HLD Architecture.pdf” shows the architecture that will be built. A copy of this can be found in the /downloads f



Data Sources

The COVID-19 datasets used in this workshop are publicly available. They are located here:

https://azure.microsoft.com/en-in/services/open-datasets/catalog/?q=covid

For this workshop, the “COVID-19 Data Lake” datasets were chosen:

https://azure.microsoft.com/en-in/services/open-datasets/catalog/covid-19-data-lake/

and then the “Bing COVID-19 Data”:

https://azure.microsoft.com/en-in/services/open-datasets/catalog/bing-covid-19-data/

We will be using the daily JSON file, in CSV format:

https://pandemicdatalake.blob.core.windows.net/public/curated/covid-19/bing_covid-19_data/latest/bing_covid-19_data.json

Activity 1 – Azure Data Lake (ADLS)

As per the diagram "Azure data Factory-Security.pdf" we are creating an Azure Data Lake (Gen-2) account.

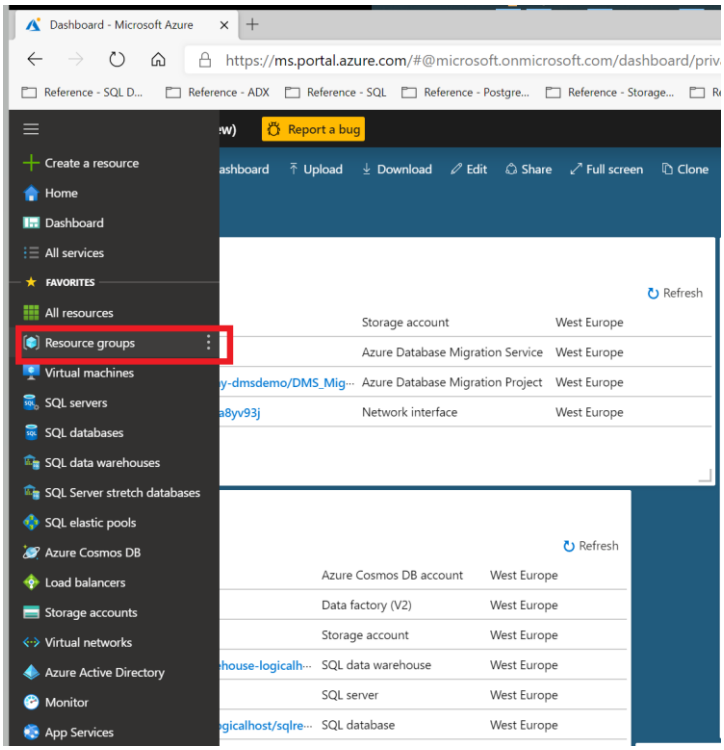
As part of this activity it maybe necessary to also refer to 'Activity 4 – Security & Access'.

Create the Resource Group

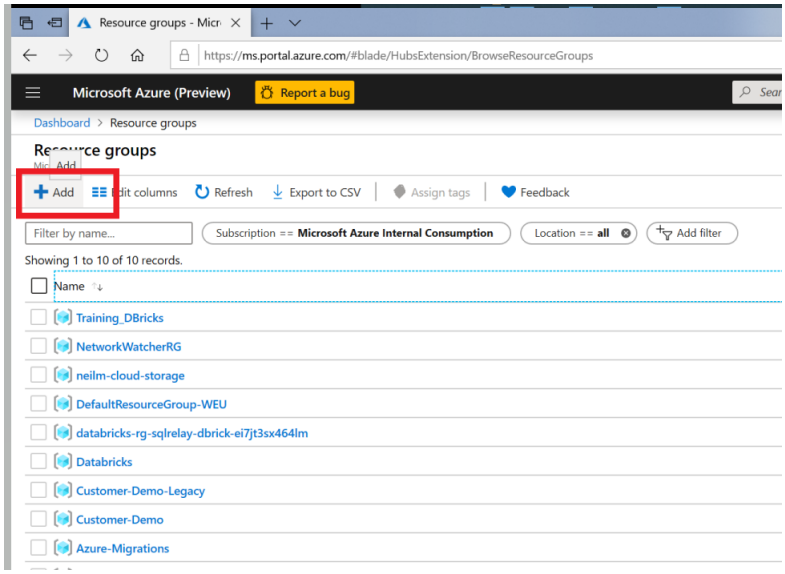
Define the Azure Resource Group that the Azure services will be created in.

NOTE: *Creating the Resource Group can also be achieved via PowerShell here.*

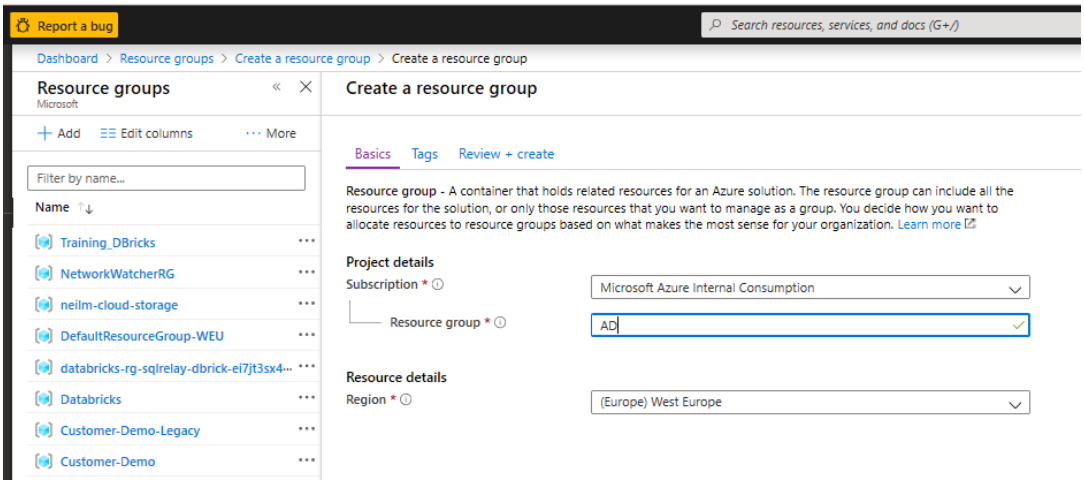
1. Sign in to the Azure portal.
2. In the Azure portal, select **Resource Groups**:



3. When the blade loads, select **Add:**



4. Enter the Resource Group Details:



Property	Description	Required
Subscription	The value must be set to the subscription the ADX resource group is to be created in.	Yes
Resource Group	The name of the workshop resource group. Enter a name of your choice.	Yes
Region	Enter the Azure region your resource group is to be located in. This should be a region that supports the Azure services required in this workshop and located as close as possible to your geographic location.	Yes

5. Select **Create.**

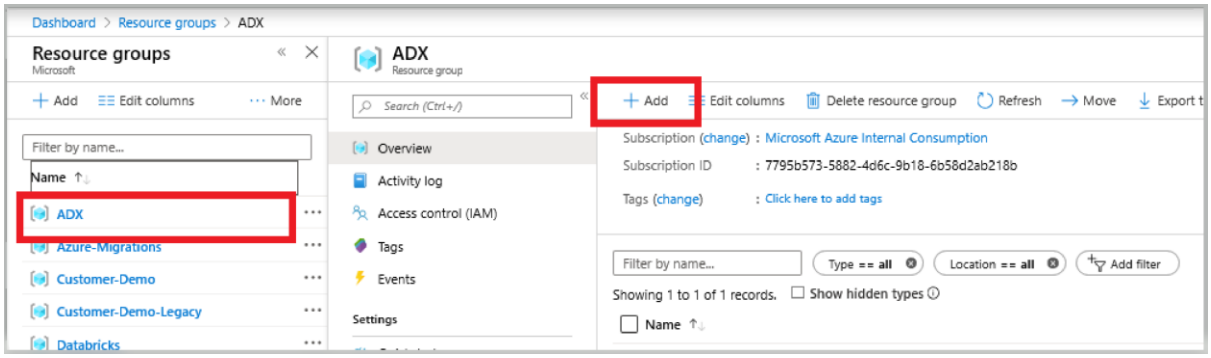
After a few minutes you will have created the resource group into which the Azure services can be deployed to.

Create the Storage Account

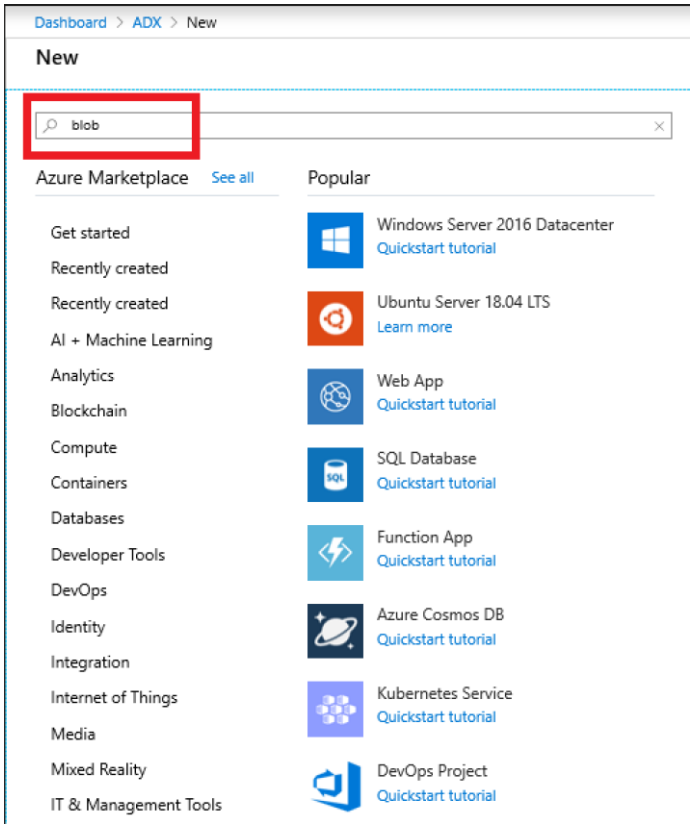
The data lake storage account will serve as the *data lake* for both this and all the other workshops.

An explanation of Azure Data Lake (ADLS) resource is here [link].

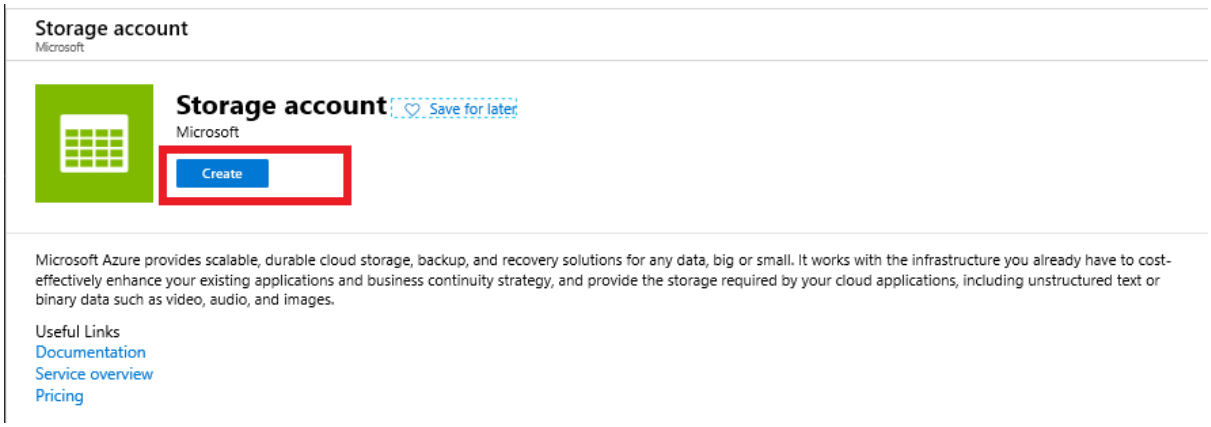
1. Within the resource Group you just created, select '**+ Add**' or '**Create Resource**'



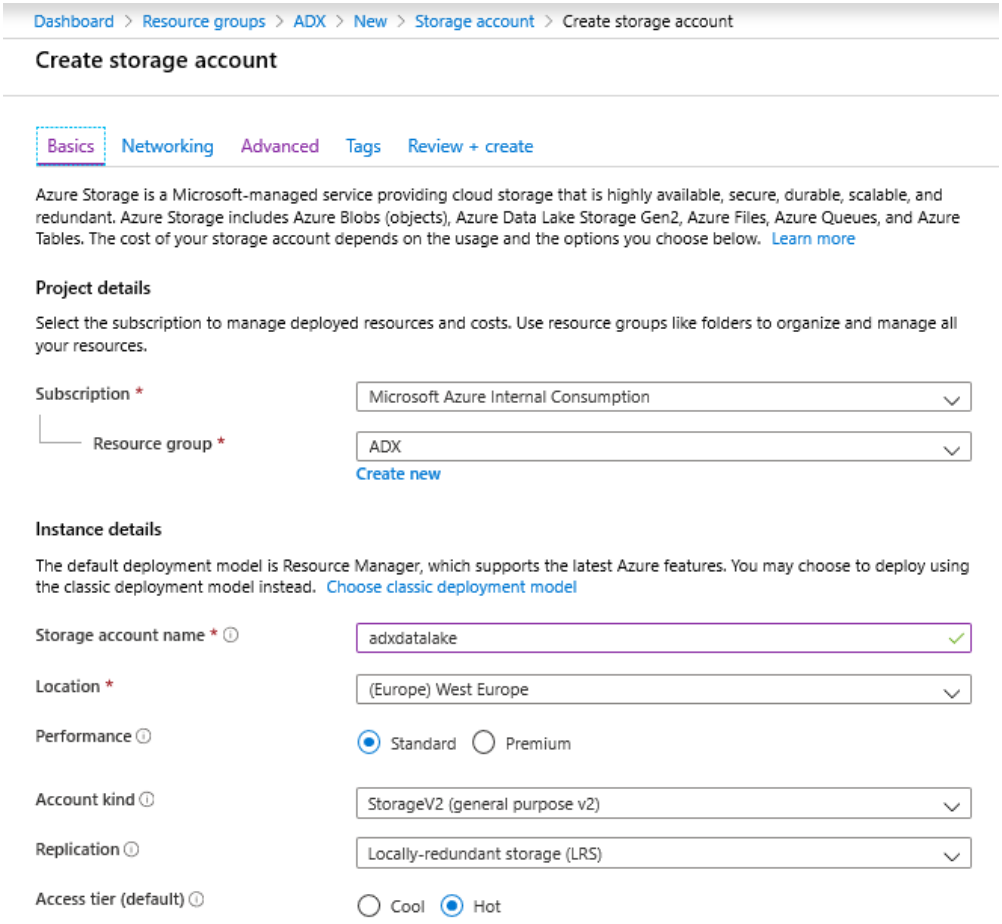
2. Type **blob** or **storage** in the search window (see below) and then select **Storage Account** from the results displayed:



3. On the Storage Account blade, select **Create**:



4. On the **Create storage account** blade, complete the Basics blade (see below)

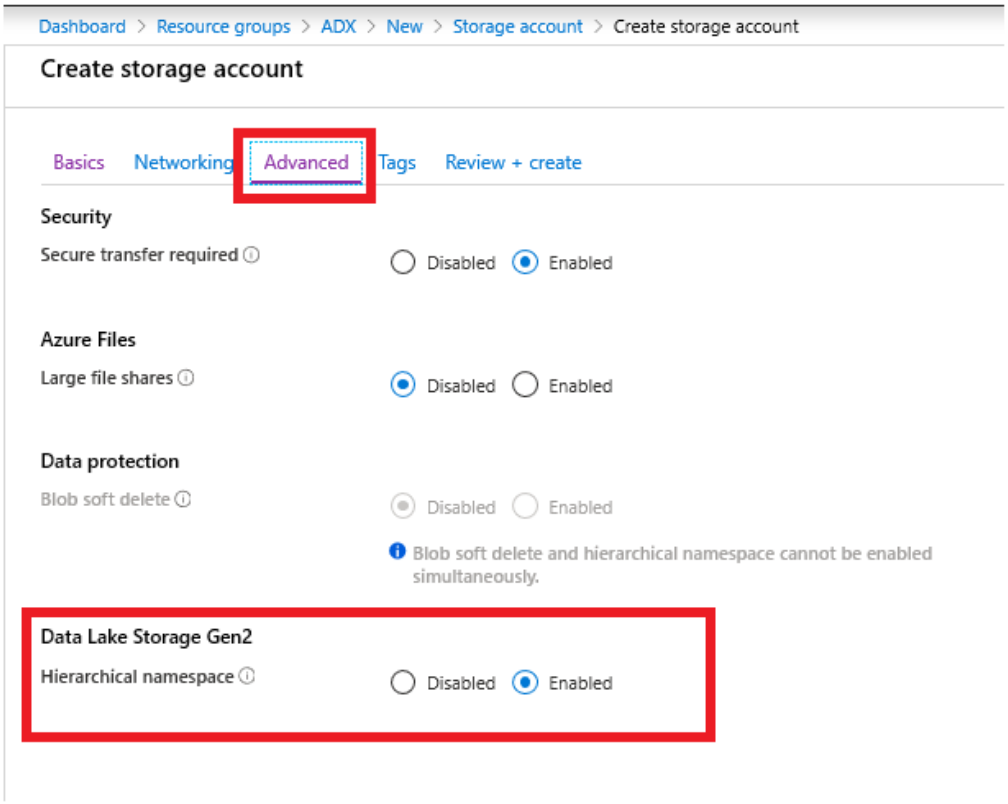


Complete the blade by entering the details as below:

Property	Description	Required
Subscription	The value must be set to the subscription the ADX resource group is created in.	Yes
Resource Group	The name of the workshop resource group you created in 'Activity 1.1'.	Yes

Property	Description	Required
Storage account name	Enter <for example> **demodatalakeadls**	Yes
Location	Select your chosen region to match the resource group *ADX* you have created.	Yes
Performance	Enter **Standard**	Yes
Account Kind	Select **Storage v2 (general purpose v2)**	Yes
Replication	Select **Locally-redundant storage (LRS)**	Yes

5. Select **Advanced**, this is where the blob storage account had the ADLS data lake attributes enabled:



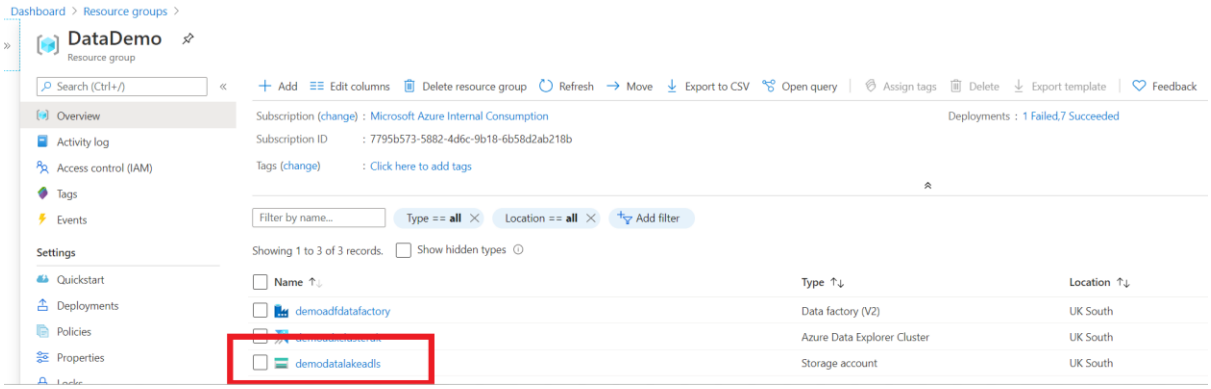
Complete the blade by entering the details as below:

Property	Description	Required
Security	Select **Enabled**	Yes

Property	Description	Required
Azure Files	Select **Disabled.**	Yes
Hierarchical Namespace	Select **Enabled**.	Yes

6. Select **Review + Create**. The blade below will appear, select **Create**:

After a short wait you will be notified that the ADLS storage has been created. You can go to the resource to review its setup or select **Resource Groups**, then select your resource group, to see the ***demodatalakeadls*** resource you have just created:



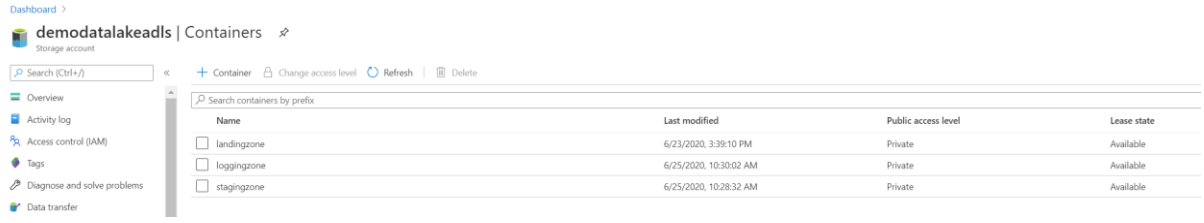
Notice that ***demodatalakeadls*** has the type: Storage Account.

Configure the ADLS storage

A root container is required for files to be landed to in a hierarchical structure. Files will be landed to ADLS via an Azure Data Factory Pipeline activity.

1. Select the resource group in the Azure portal
2. Select ***demodatalakeadls*** storage account
3. Select **Containers** under **Data Lake Storage**
4. Select **+ Container** to add a new container, as below:
5. Enter ***landingzone*** and select **OK;**
6. Repeat and create folders ***stagingzone*** and ***loggingzone***

The container structure should match the image below once completed. Also note the **Public Access Level** and **Lease State** settings:



The JSON document 'demodatalakeadls.json' has the configuration settings for ***demodatalakeadls*** is located in the /downloads folder.

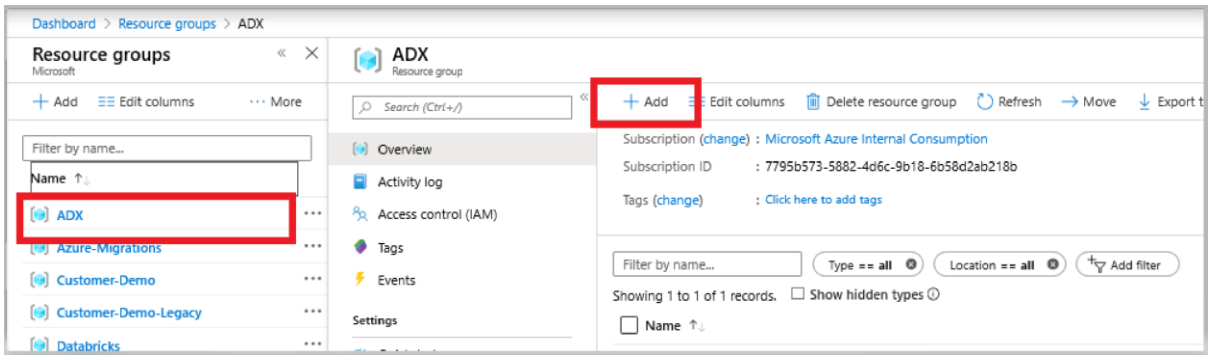
Activity 2 – Azure Data Explorer (ADX)

Create ADX Cluster

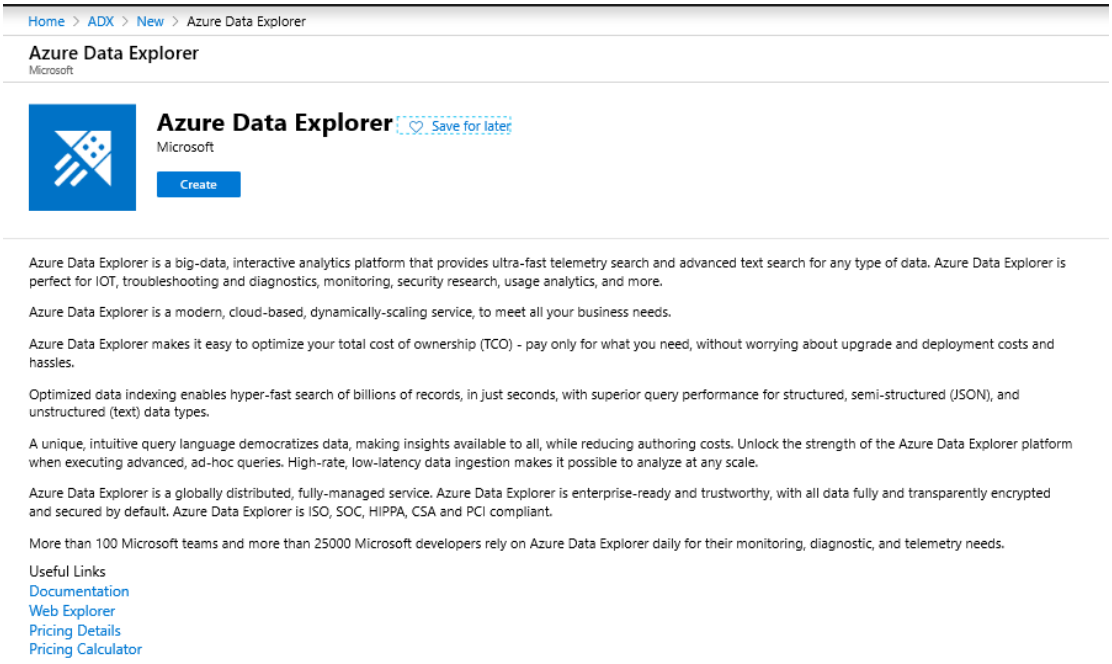
As per the diagram "Lab1 - Activity 4 – Azure Data Explorer.pdf" we are creating the ADX cluster.

As part of this activity it may be necessary to also refer to 'Activity 4 – Security & Access'.

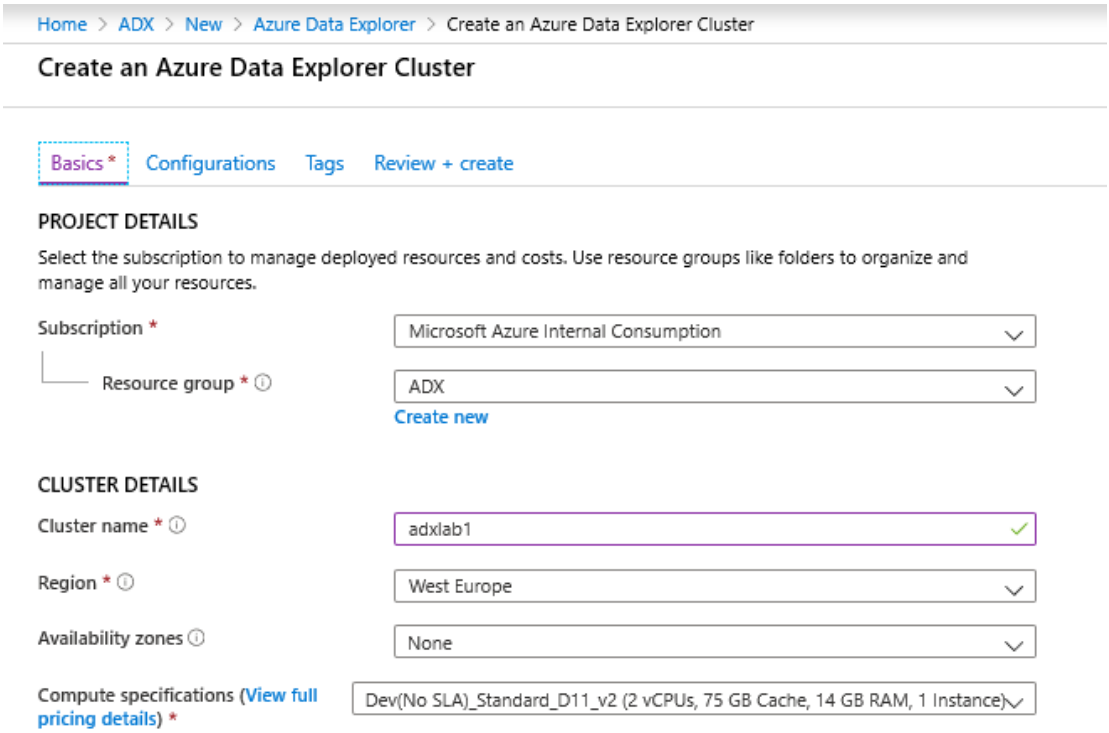
1. Within the ADX resource Group, select **+ Add** or **Create Resource**



2. Type **data explorer** in the search window (see below) and then select **Azure Data Explorer** from the results displayed. The following blade will appear:



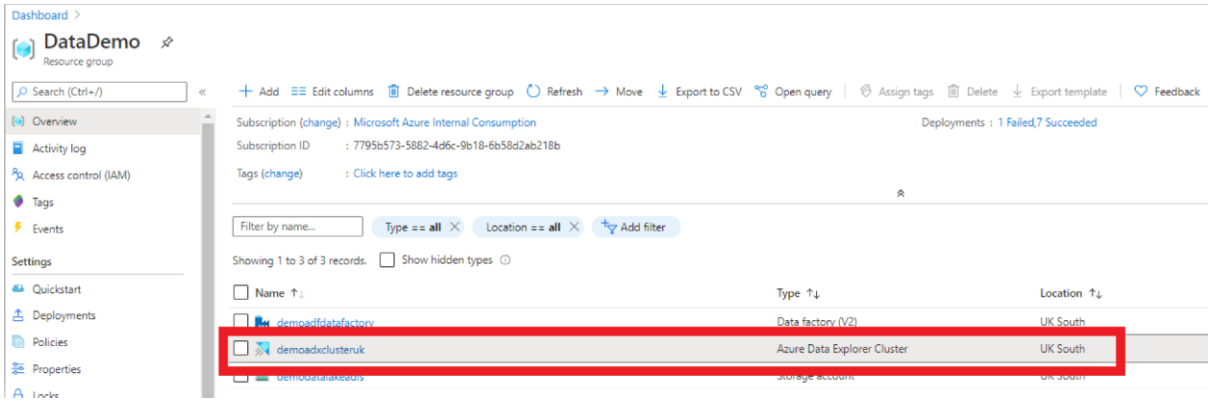
3. Select **Create** and the following blade will appear:



Complete the **Basics** blade by entering the details as below:

Property	Description	Required
Subscription	The value must be set to the subscription the ADX resource group is created in.	Yes
Resource Group	The name of the workshop resource group.	Yes
Name	Enter **demoadxclusteruk**	Yes
Region	Select your chosen region to match the resource group *ADX* you have created.	Yes
Availability zones	Select **None**	Yes
Compute specifications	The options available in this listbox will depend upon you **Region** selection. As this is a workshop, select a low spec configuration to work with. For a PoC, MVP or production installations you will need a higher specification. Select '**Dev(No SLA)**' or the smallest specification in your listbox.	Yes

4. Select **Review + Create**, then select **Create** to build the ADX cluster. After a short wait you will be notified that the ADX cluster has been created. You can go to the resource to review its setup or select **Resource Groups**, then select **your resource group**, to see the ***demoadxclusteruk*** resource you have just created:



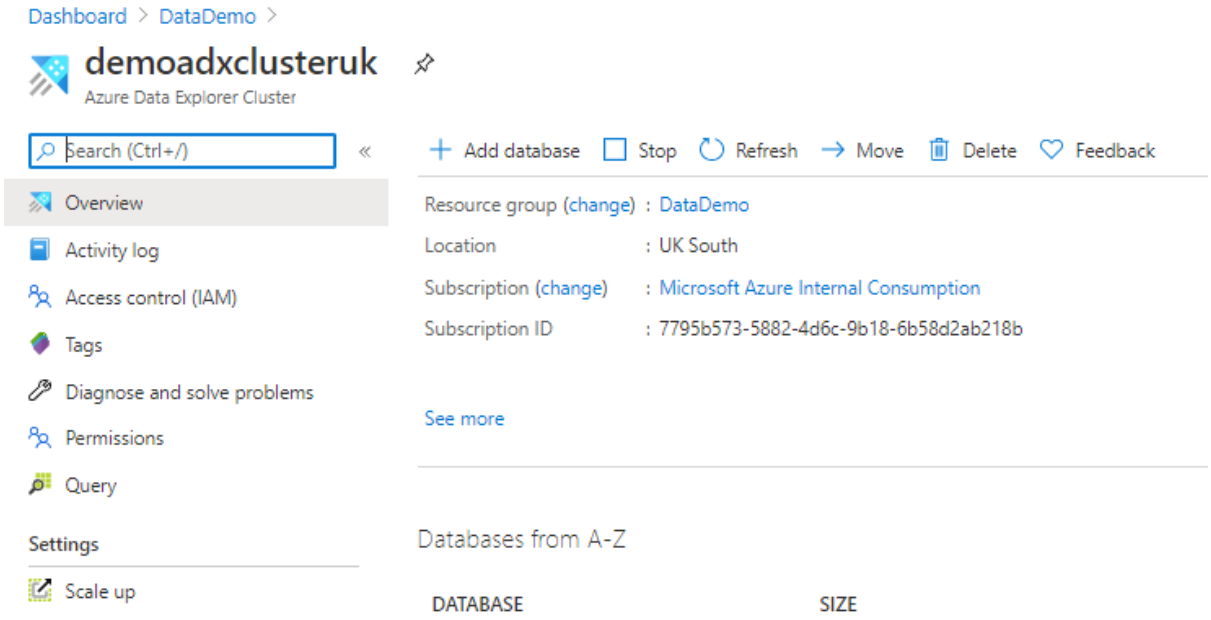
5. Select **Properties** to see the ADX cluster URI. This will be needed later in this workshop.

Create ADX Database

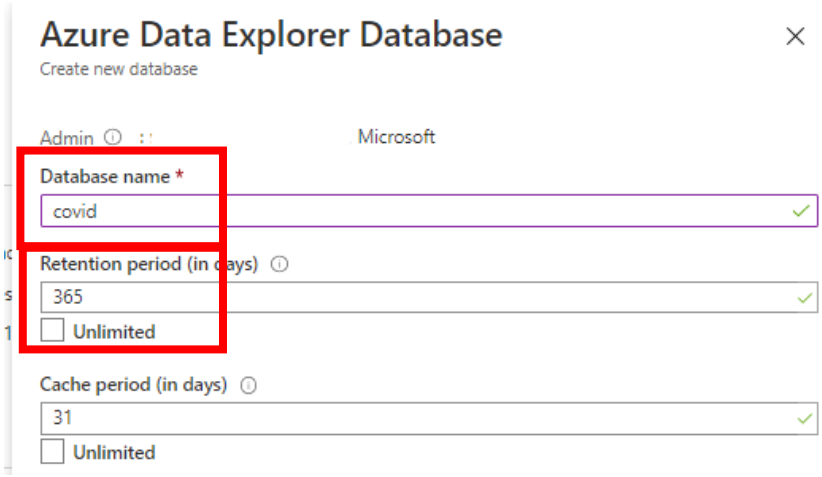
We will now create the ADX database for the COVID-19 data to be ingested and loaded into.

The KQL script to for the DDL [KQL_covidstaging.kql] is located in the /donloads folder.

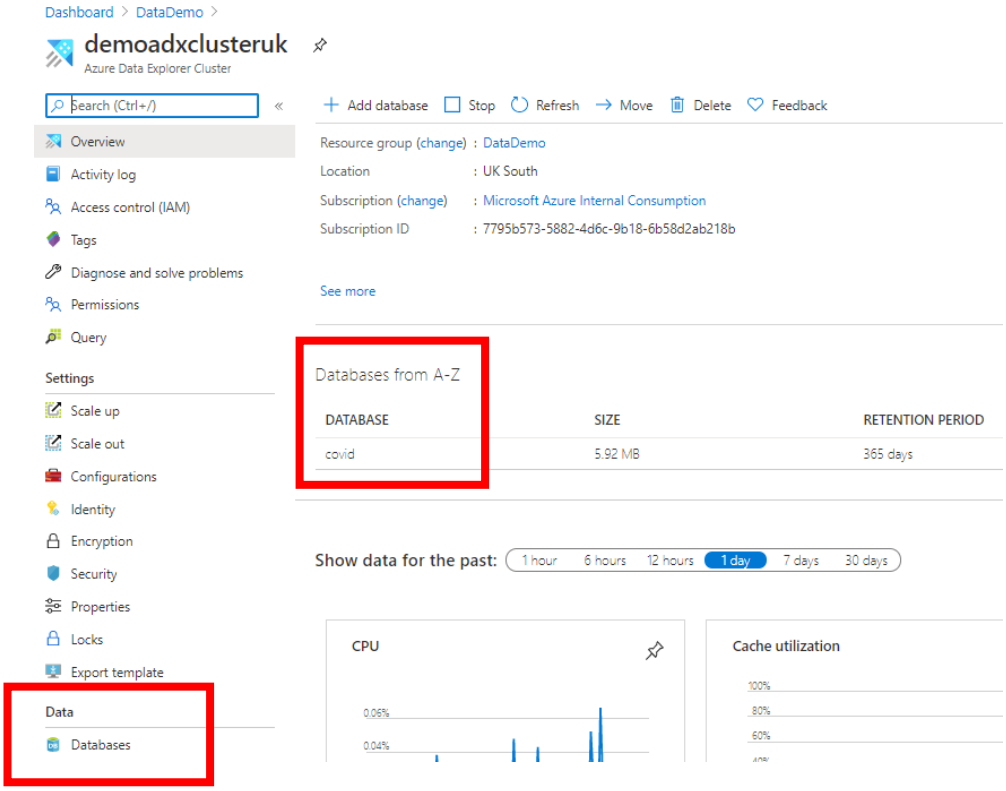
1. In the workshop resource group, select the ADX cluster resource you created [**demoadxclusteruk**]
2. When the ADX resource blade appears, select '**+ Add database**':



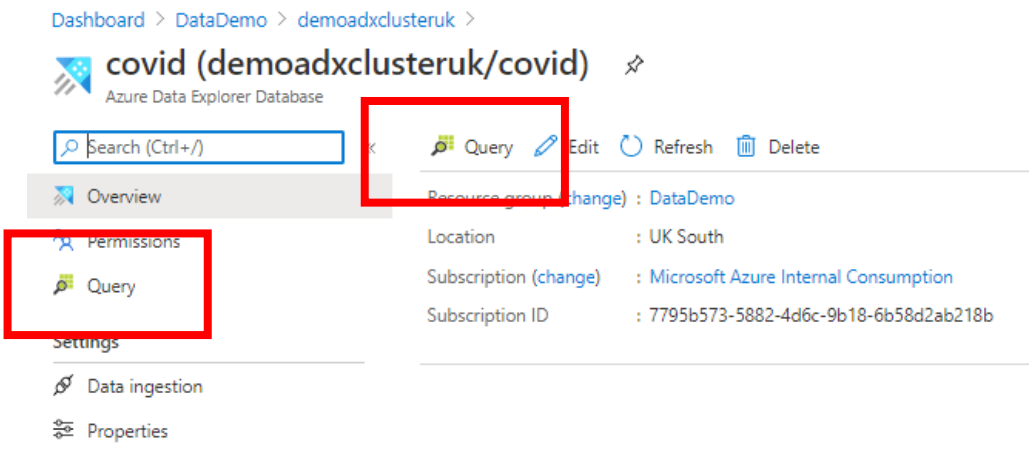
3. Enter '**covid**' for the database name and complete the tab as below:



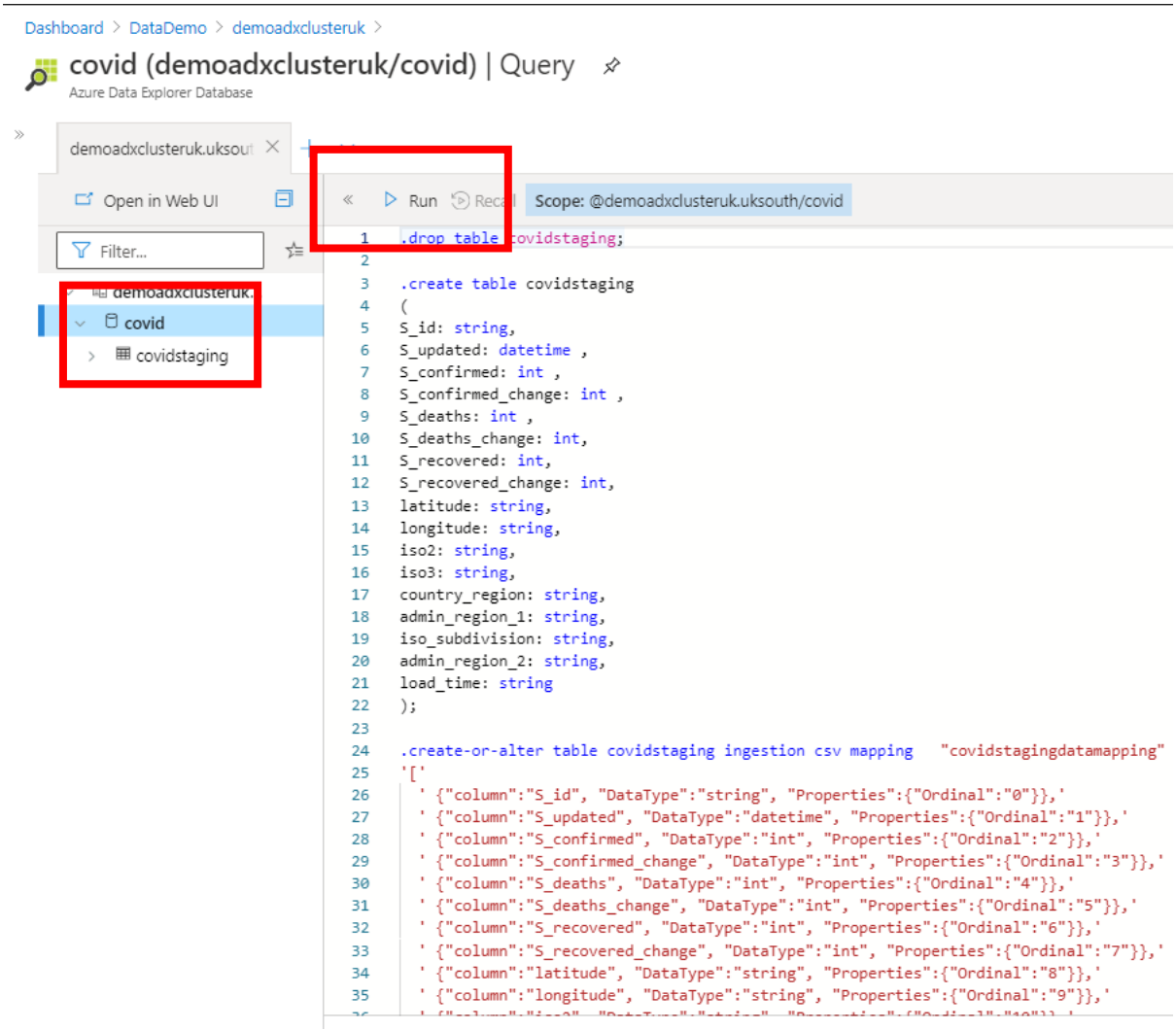
4. Select **Create** to complete the database creation.
5. Once the database has been created, select **covid** under **Databases from A-Z** or **Data**:



6. When the **covid** database blade appears, select **Query**:



7. When the query window appears, open/paste the [KQL_covidstaging.kql] KQL into the query window and then select **Run** to create the table. The newly created table will appear under the **covid** database outline:



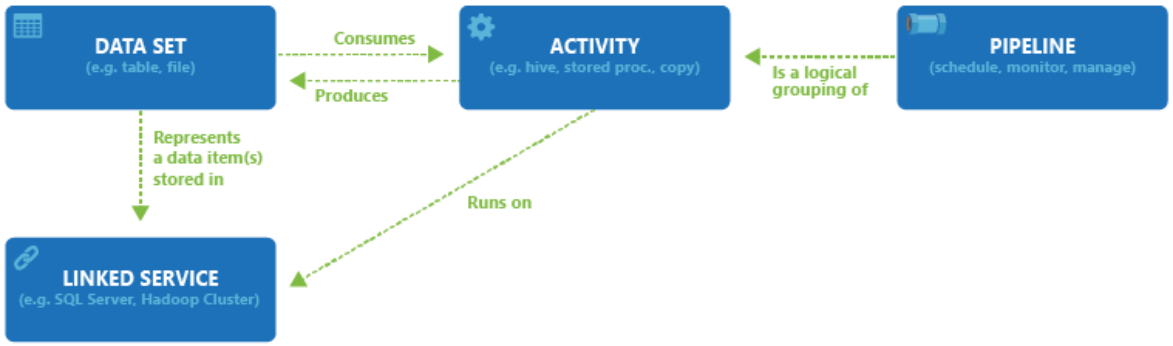
Activity 3 – Data Factory (ADF)

Overview

Data Factory provides a single hybrid data integration service for all skill levels. It is a cloud-based data integration service that allows you to create data-driven workflows in the cloud that orchestrate and automate data movement and data transformation.

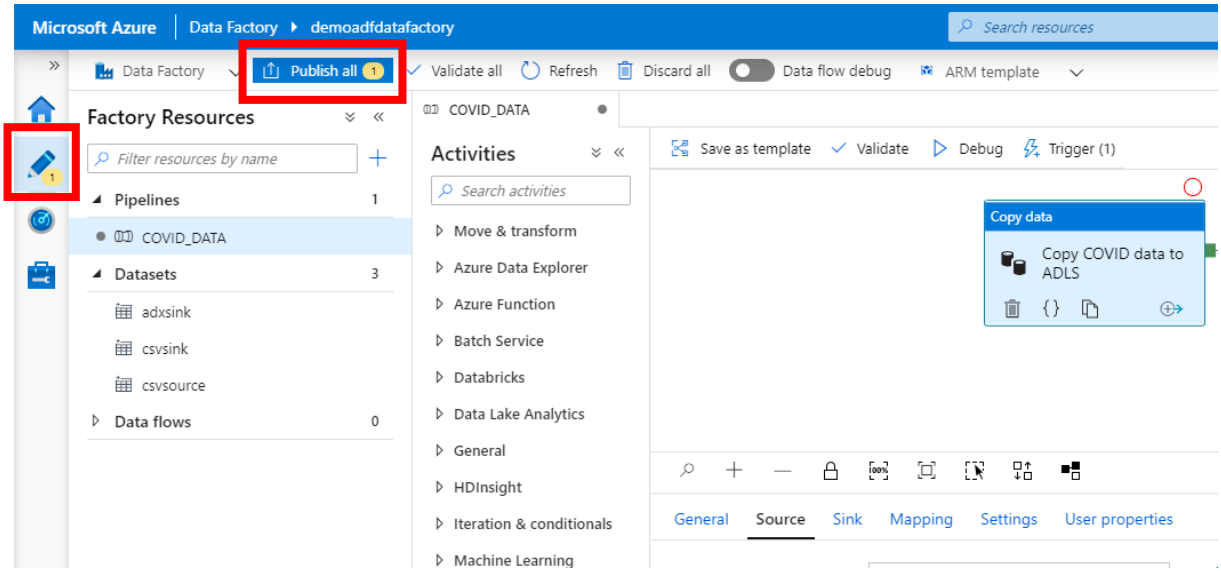
An explanation of Azure Data Factory (ADF) is here [link].

For an explanation of Azure Data Factory (ADF) terminology used within the next sections [Activities, Datasets, Pipeline, Linked Service), please see this link: https://docs.microsoft.com/en-us/azure/data-factory/concepts-datasets-linked-services



Publishing in ADF

ADF runs as a service. When working with Pipelines, Datasets, Activities and all objects within ADF, you need to 'Publish' (or save) your changes. You can see when ADF requires that changes need to be published(saved) as a yellow count of unsaved changes is displayed on the ADF resources blade, as shown in the image below:



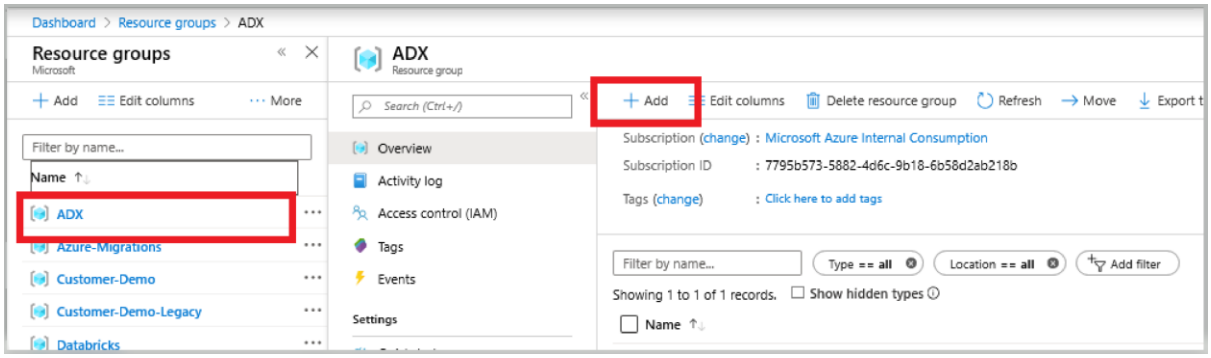
Through the course of the creating the ADF pipeline, remember to 'Publish' all changes.

As part of this activity it may be necessary to also refer to 'Activity 4 – Security & Access'.

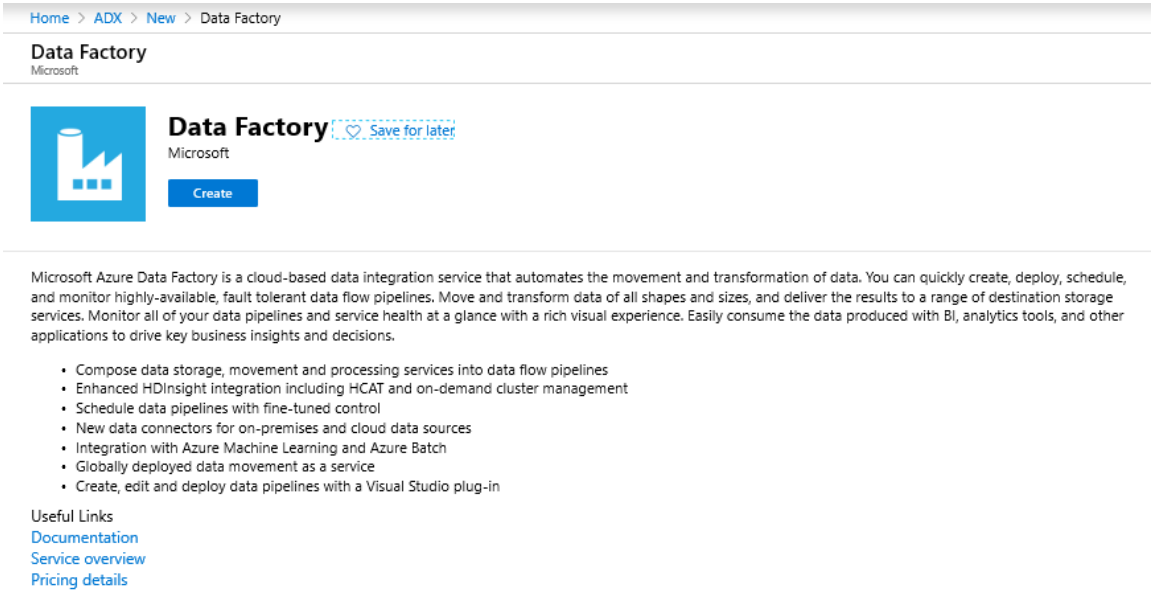
Create the Data Factory

As per the diagram "Azure data Factory-Security.pdf" we are creating an Azure Data Factory (ADF). The ADF activity will be triggered by a schedule. The ADF pipeline will move the COVID-19 data file to the ***demodatalakeadls*** and deposit it in the **/landingzone/..** hierarchical structure based on the Year/Month/Day of the file pull.

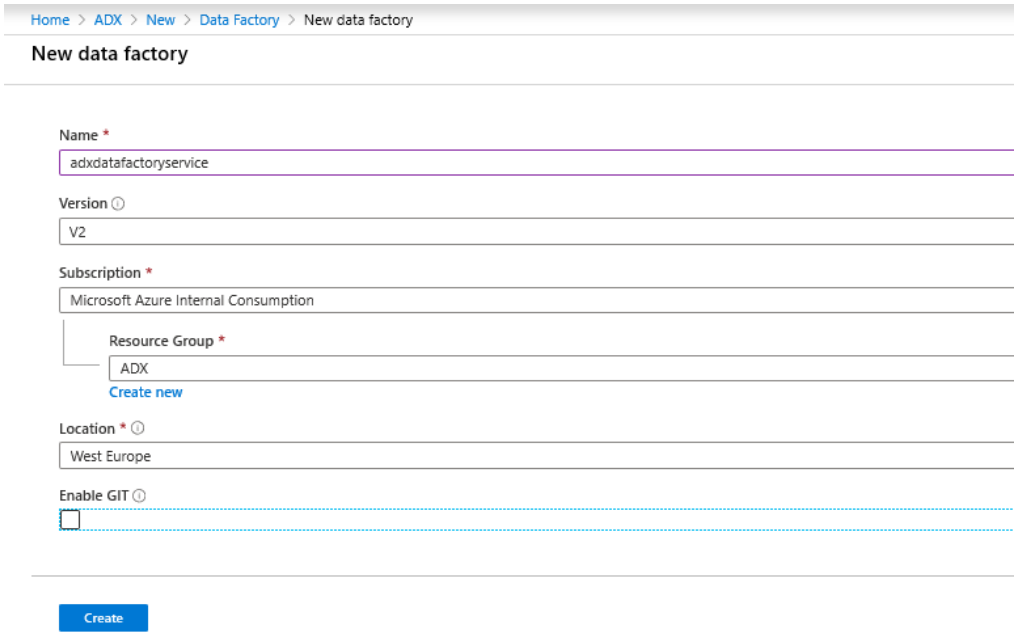
1. Within the ADX resource Group, select **+ Add** or **Create Resource**



2. Type **data factory** in the search window (see below) and then select **Data Factory** from the results displayed. The following blade should be displayed:



3. Select **Create**, the **New data factory** blade will appear:

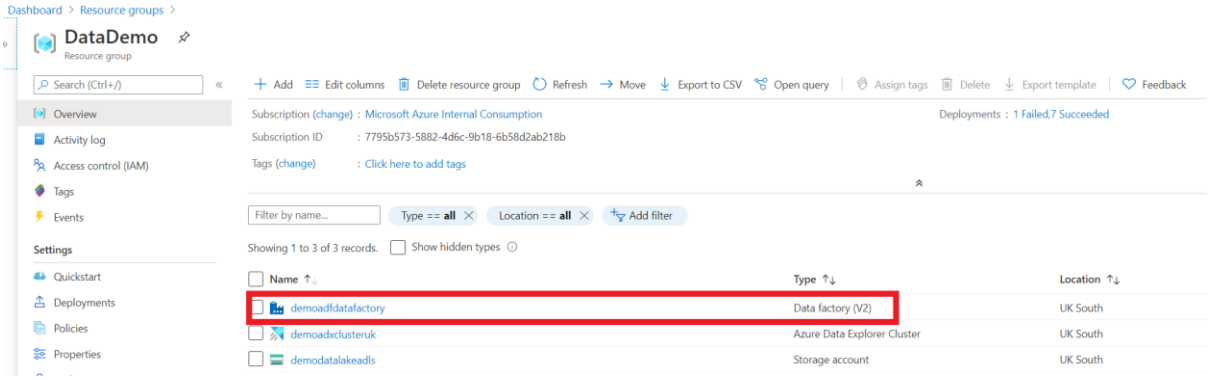


Complete the blade by entering the details as below:

Property	Description	Required
Name	Enter **demoadfdatafactory**	Yes
Version	Enter **V2**	Yes
Subscription	The value must be set to the subscription the ADX resource group is created in.	Yes
Resource Group	The name of the workshop resource group you have chosen.	Yes
Region	Select your chosen region to match the resource group *ADX* you have created.	Yes
Enable GIT	Uncheck this box. GIT integration allows you to utilise in a DevOps CI/CD environment. This integration is beyond the scope of this workshop, so will not be configured. Further details can be found here if this is of interest. https://docs.microsoft.com/en-us/azure/data-factory/continuous-integration-deployment	Yes

NOTE: *GIT integration allows you to utilise in a DevOps CI/CD environment. This integration is beyond the scope of this workshop, so will not be configured. Further details can be found here - https://docs.microsoft.com/en-us/azure/data-factory/continuous-integration-deployment.*

4. Select **Create** to build the ADF. After a short wait you will be notified that the Azure Data Factory has been created. You can go to the resource to review its setup or select **Resource Groups**, then select **your resource group**, to see the ***demoadfdatafactory*** resource you have just created:

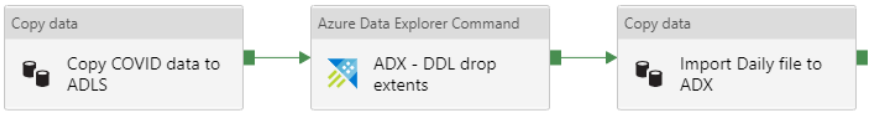


Create ADF Pipeline

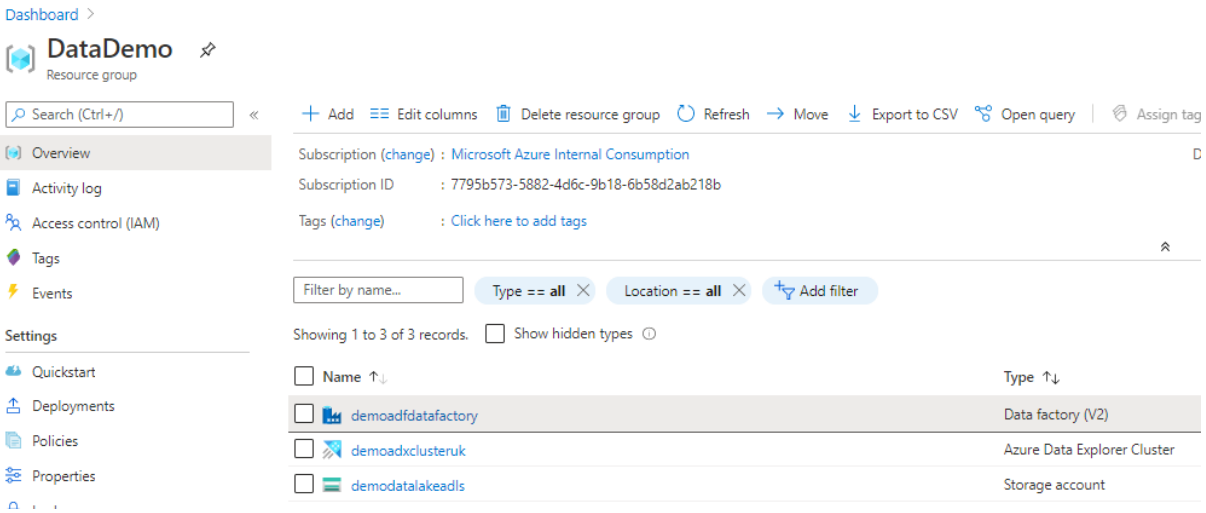
We will now create the ADF pipeline. The pipeline performs the following activities:

a) Connects to the public COVID-19 server and ingests(pulls) the daily (csv) dataset to Azure
b) Saves the COVD-19 data to a parameterised data (yyyy/MM/DD) driven hierarchy
c) Loads the daily file into ADX

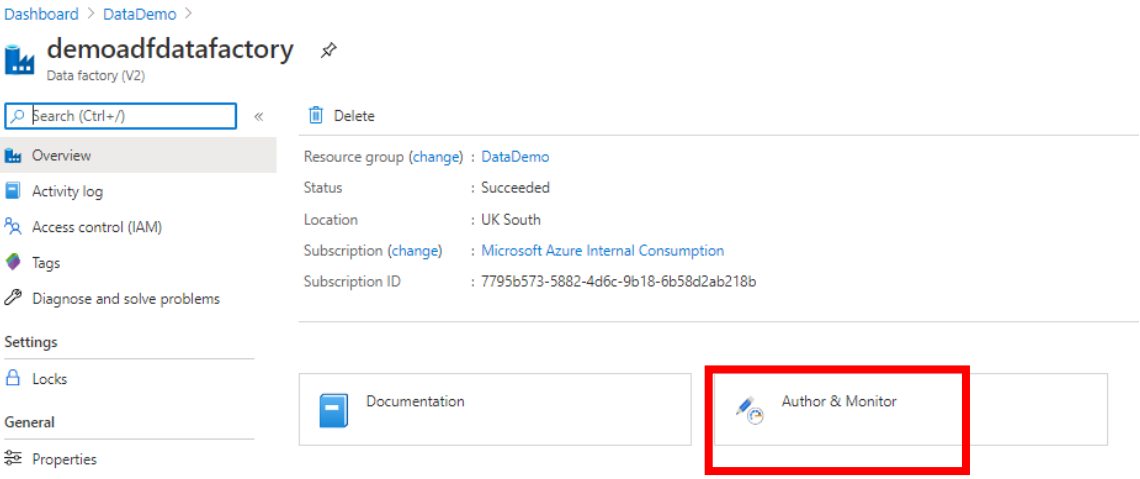
The steps above correspond to the image below:



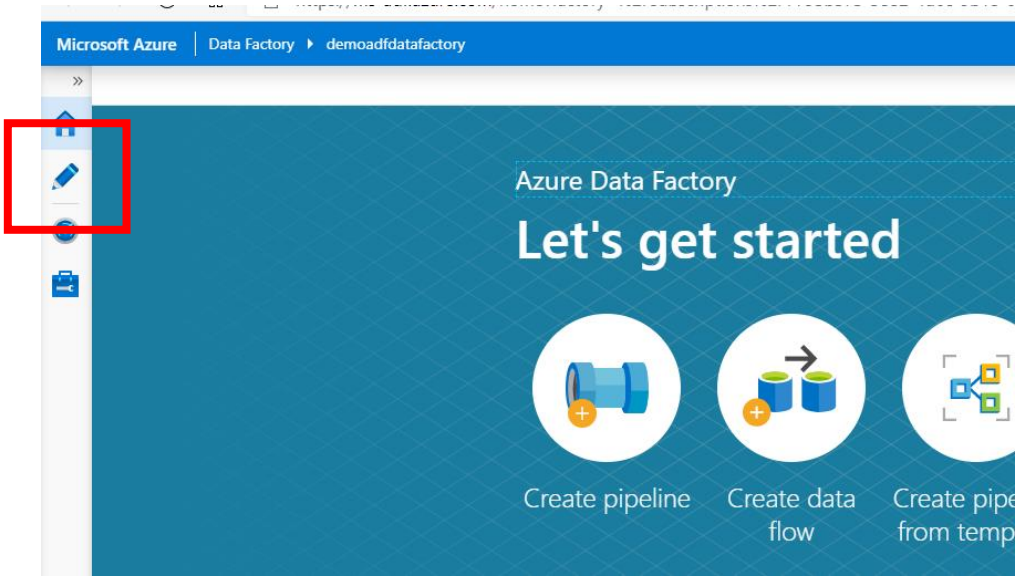
1. Select the ADF created earlier:



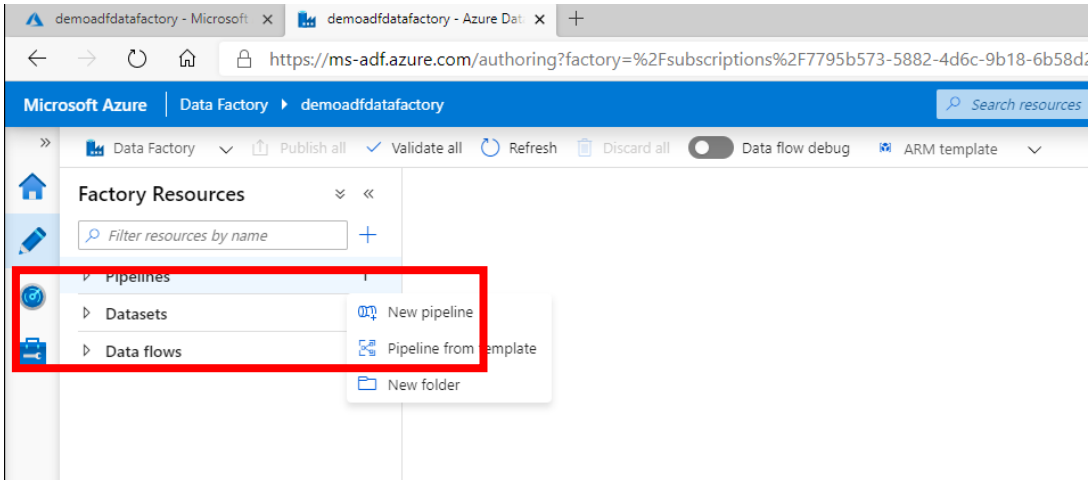
2. Select **Author & Monitor:**



3. Select **Author** icon:



4. Select **Pipelines,** then '**…**', then '**New Pipeline**':


demoadfdatafactory - Microsoft
demoadfdatafactory - Azure Dat
https://ms-adf.azure.com/authoring?factory=%2Fsubscriptions%2F7795b573-5882-4d6c-9b18-6b58d
Microsoft Azure
Data Factory
demoadfdatafactory
Search resources
Data Factory
Publish all
Validate all
Refresh
Discard all
Data flow debug
ARM template
Factory Resources
Filter resources by name
Pipelines
Datasets
Data flows
New pipeline
Pipeline from template
New folder

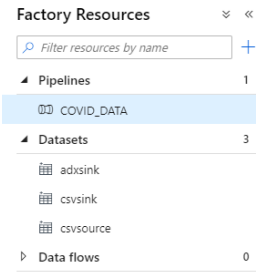
Create Pipeline Linked Services

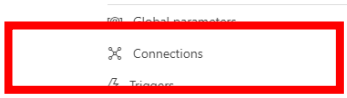
ADF 'Httplinkedservice' linked service

This is the ADF connection to the COVID-19 public [http hosted] datasets.

The JSON file for this ADF activity is in the /download folder "Httplinkedservice.json".

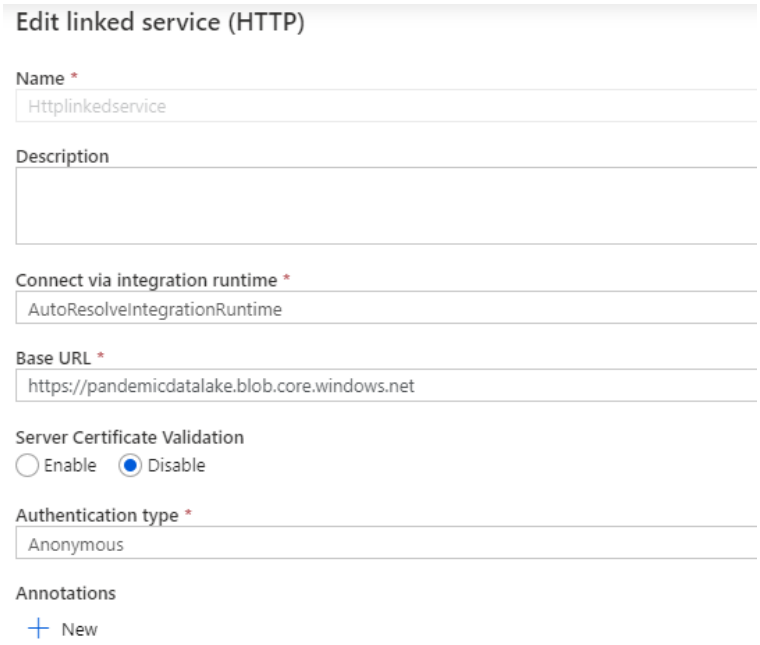
1. Select **Connections:**





2. Select **'+ New'**

   Complete the blade as per the screenshot below. Select '**Test Connection**' to confirm the Linked Service is working correctly. Select '**Apply**'.



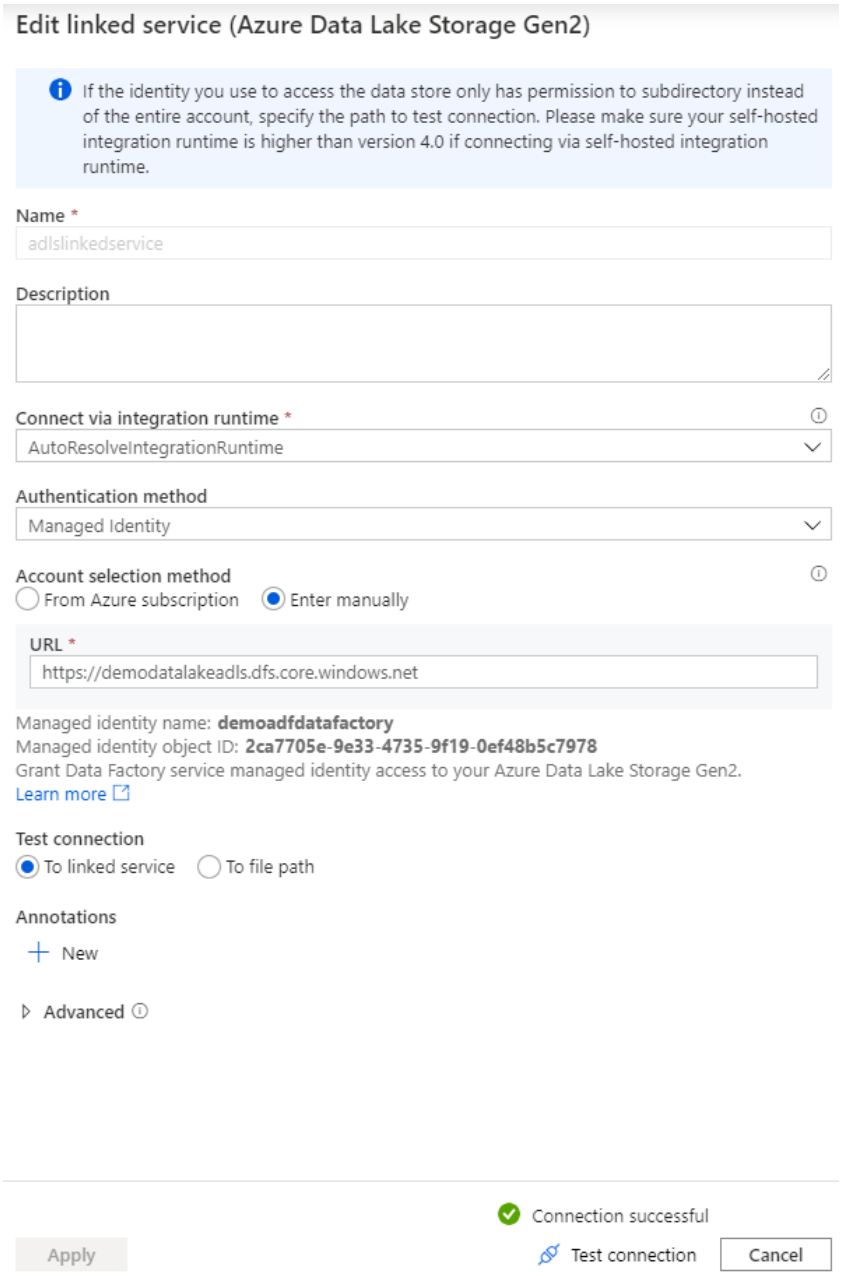
ADF 'adlslinkedservice' linked service

This is the ADF connection to ADLS. Note, on this blade, the 'Managed Identity details (name & object ID) are shown. These should be noted, as they will be required in the section '**Security & Access**'.

The JSON file for this is in the /download folder "adlslinkedservice.json".

1. Select **'+ New**'

   Complete the blade as per the screenshot below. Select '**Test Connection**' to confirm the Linked Service is working correctly. Select '**Apply**':


Edit linked service (Azure Data Lake Storage Gen2)
If the identity you use to access the data store only has permission to subdirectory instead of the entire account, specify the path to test connection. Please make sure your self-hosted integration runtime is higher than version 4.0 if connecting via self-hosted integration runtime.
Name *
adlslinkedservice
Description
Connect via integration runtime *
AutoResolveIntegrationRuntime
Authentication method
Managed Identity
Account selection method
From Azure subscription
Enter manually
URL *
https://demodatalakeadls.dfs.core.windows.net
Managed identity name: demoadfdatafactory
Managed identity object ID: 2ca7705e-9e33-4735-9f19-0ef48b5c7978
Grant Data Factory service managed identity access to your Azure Data Lake Storage Gen2.
Learn more
Test connection
To linked service
To file path
Annotations
New
Advanced
Connection successful
Apply
Test connection
Cancel

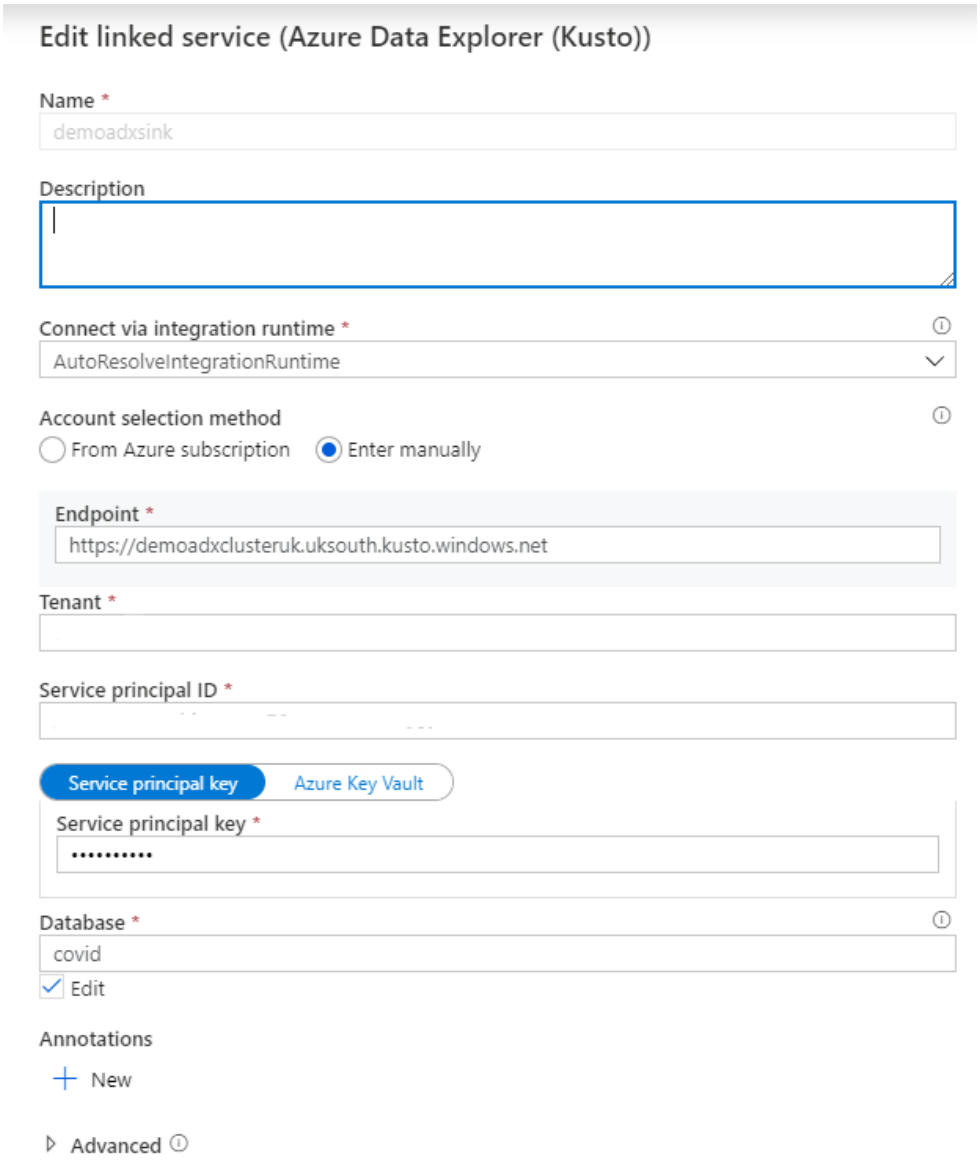
ADF 'demoadxsink' linked service

This is the ADF connection to the ADX cluster. Note, the Service Principal details are required from the 'Security & Access' section.

The JSON file for this is in the /download folder "demoadxsink.json".

1. Select **'+ New'**

   Complete the blade as per the screenshot below. Select '**Test Connection**' to confirm the Linked Service is working correctly. Select '**Apply**':



Note: the 'Tenant' and 'Service Principal ID' fields have been blanked out. You need to substitute your Service Principal values here.

Create Pipeline Datasets

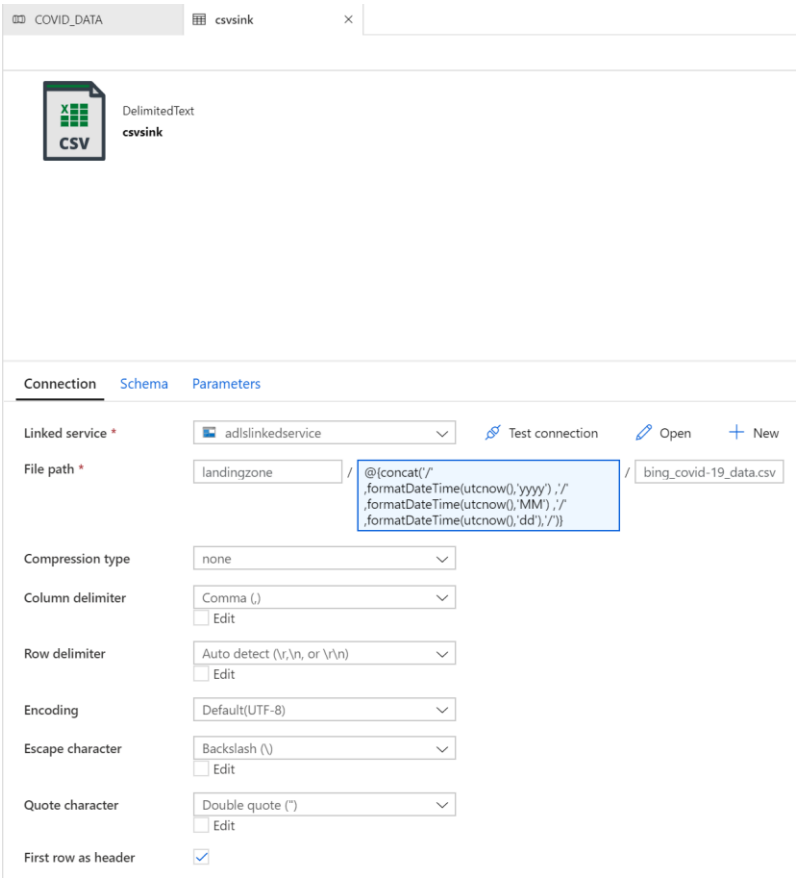
ADF 'csvsink' dataset

This dataset uses the '**adlslinkedservice'** to write the COVID-19 dataset into a date parameterised folder structure: landingzone/YYYY/MM/DD/.

The JSON file for this is in the /download folder "csvsink.json".

1. Complete the **Connection** / **Schema** tabs as per the following images.
2. **Connection** tab:

   Select the **adlslinkedservice** Linked Service and then '**Test Connection**' to verify.



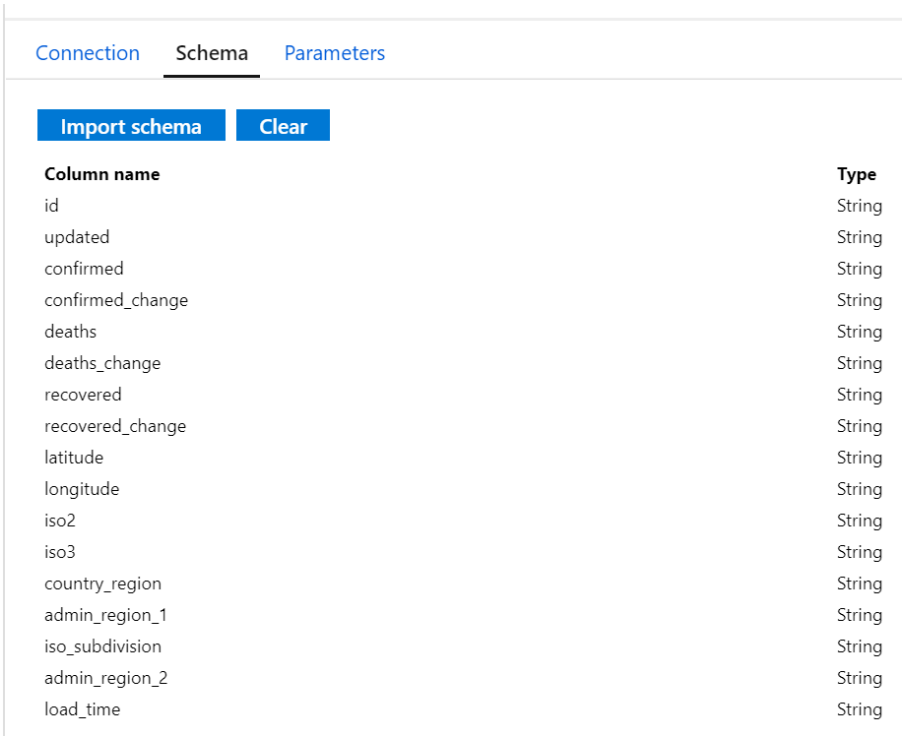
   Note: for clarity, field values are:

   - **File Path** is:

     @{concat('/' ,formatDateTime(utcnow(),'yyyy') ,'/' ,formatDateTime(utcnow(),'MM') ,'/' ,formatDateTime(utcnow(),'dd'),'/')}

3. Select '**Preview Data**' to check the dataset is working correctly.
4. **Schema** tab:

   Select '**Import Schema**' to verify the datafile/dataset (+linked service)


Connection
Schema
Parameters
Import schema
Clear
Column name
Type
id
String
updated
String
confirmed
String
confirmed_change
String
deaths
String
deaths_change
String
recovered
String
recovered_change
String
latitude
String
longitude
String
iso2
String
iso3
String
country_region
String
admin_region_1
String
iso_subdivision
String
admin_region_2
String
load_time
String

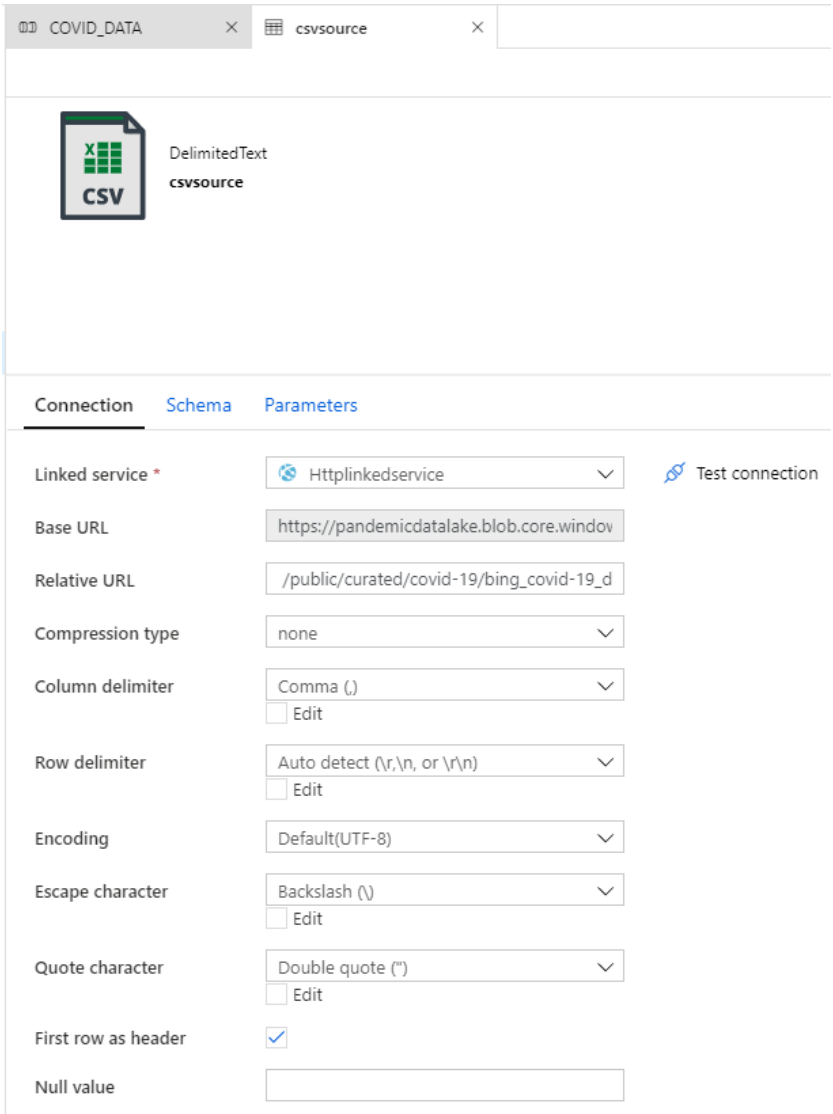
ADF 'csvsource' dataset

This dataset uses the '**Httplinkedservice'** to read the COVID-19 dataset from the public [https://pandemicdatalake.blob.core.windows.net] host server.

The JSON file for this is in the /download folder "csvsource.json".

1. Complete the **Connection** / **Schema** tabs as per the following images.
2. **Connection** tab:

   Select the **Httplinkedservice** Linked Service and then '**Test Connection**' to verify.

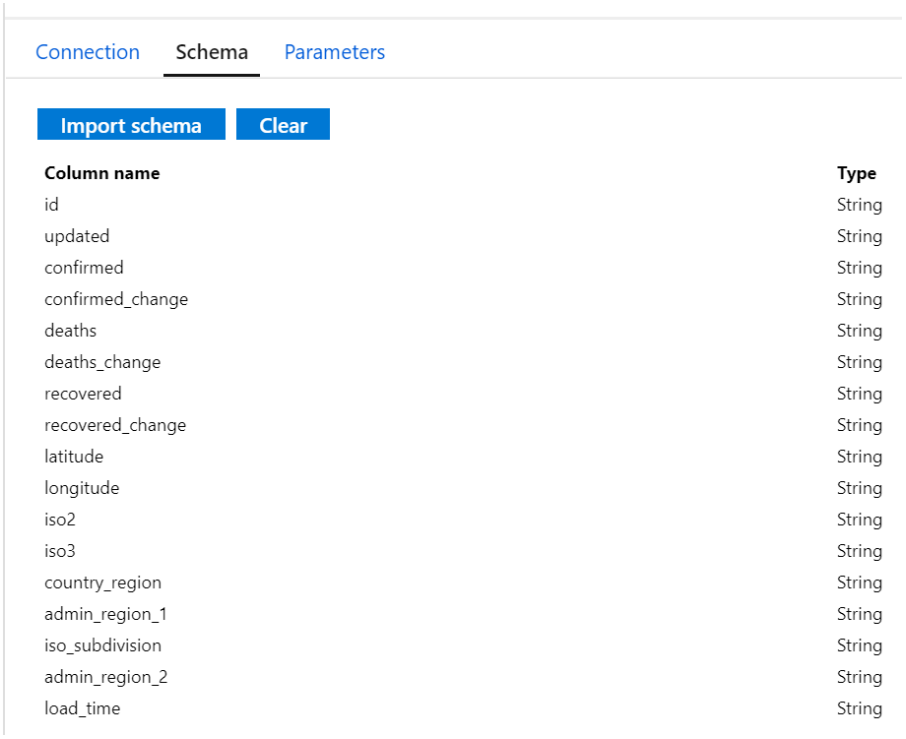


   Note: for clarity, field values are:

   - **Base URL**: https://pandemicdatalake.blob.core.windows.net
   - **Relative URL**: /public/curated/covid-19/bing_covid-19_data/latest/bing_covid-19_data.csv

3. **Schema** tab:

   Select '**Import Schema**' to verify the datafile/dataset (+linked service)


Connection
Schema
Parameters
Import schema
Clear
Column name
Type
id
String
updated
String
confirmed
String
confirmed_change
String
deaths
String
deaths_change
String
recovered
String
recovered_change
String
latitude
String
longitude
String
iso2
String
iso3
String
country_region
String
admin_region_1
String
iso_subdivision
String
admin_region_2
String
load_time
String

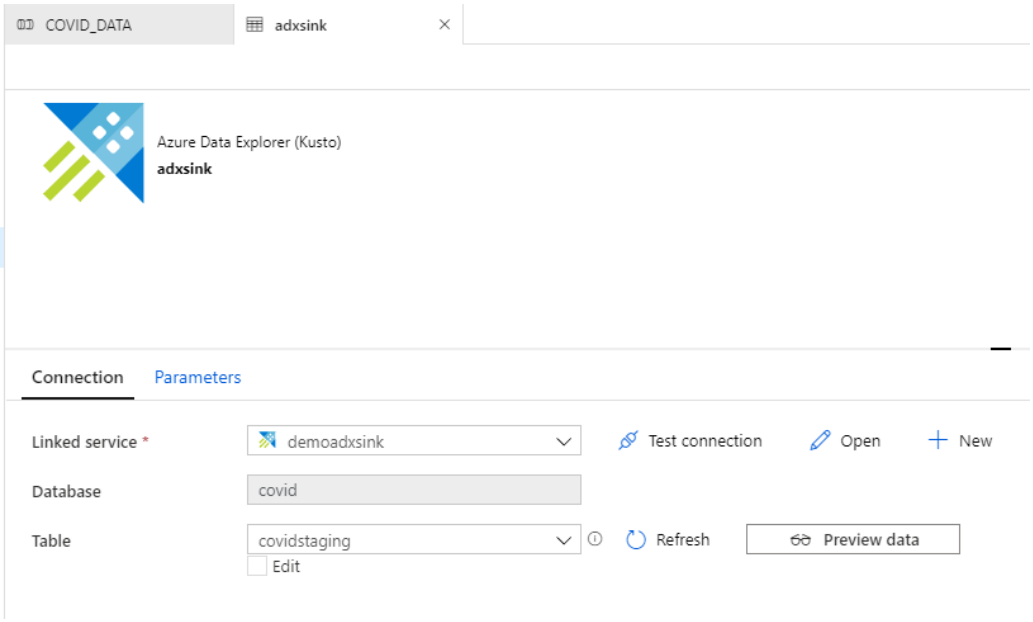
ADF 'adxsink' dataset

This dataset uses the '**demoadxsink'** to connect to the ADX cluster you have created.

The JSON file for this is in the /download folder "demoadxsink.json".

1. Complete the tab as per the following images.
2. **Connection** tab:

   Select the **demoadxsink'** Linked Service and then '**Test Connection**' to verify.



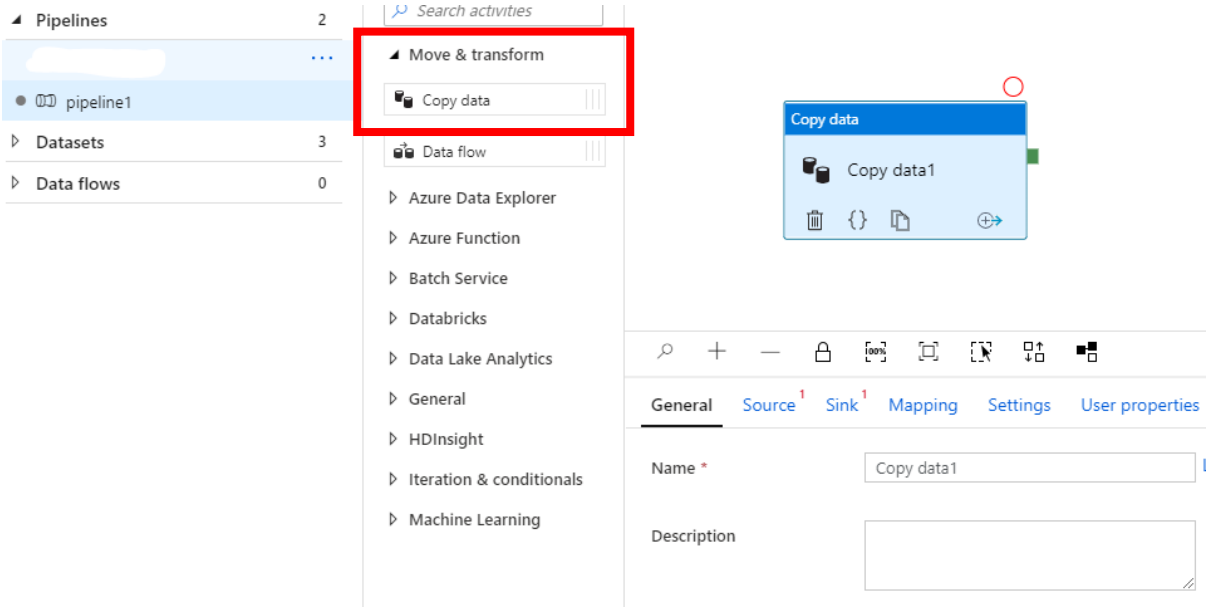
Create Pipeline Activities

ADF 'Copy COVID data to ADLS' activity

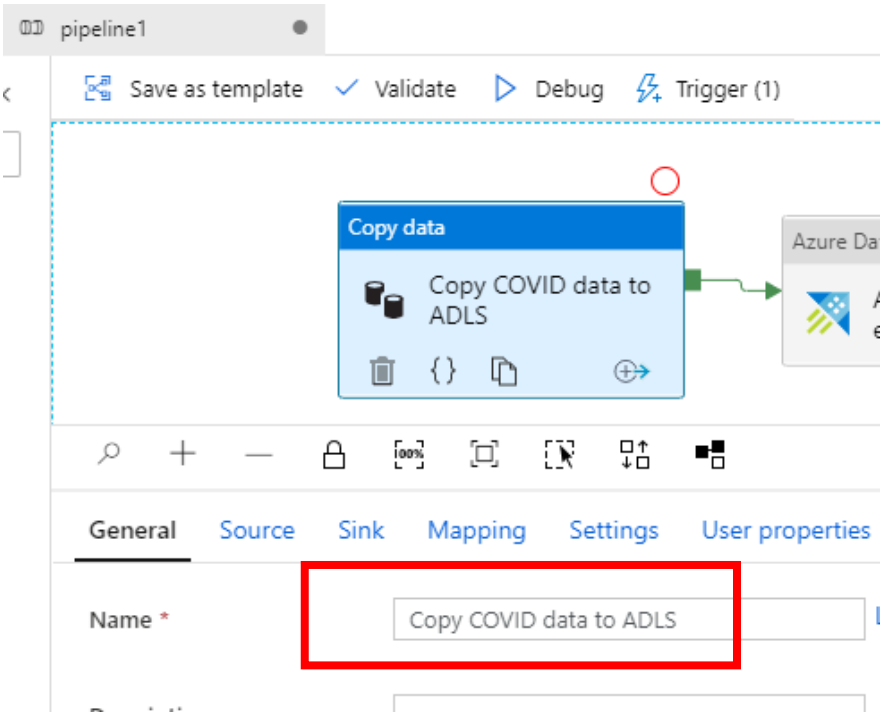
This activity connects to the COVID-19 public server and ingests the daily dataset to Azure ADLS.

The JSON file for this ADF activity is in the /download folder "ADF - Copy COVID data to ADLS.json".

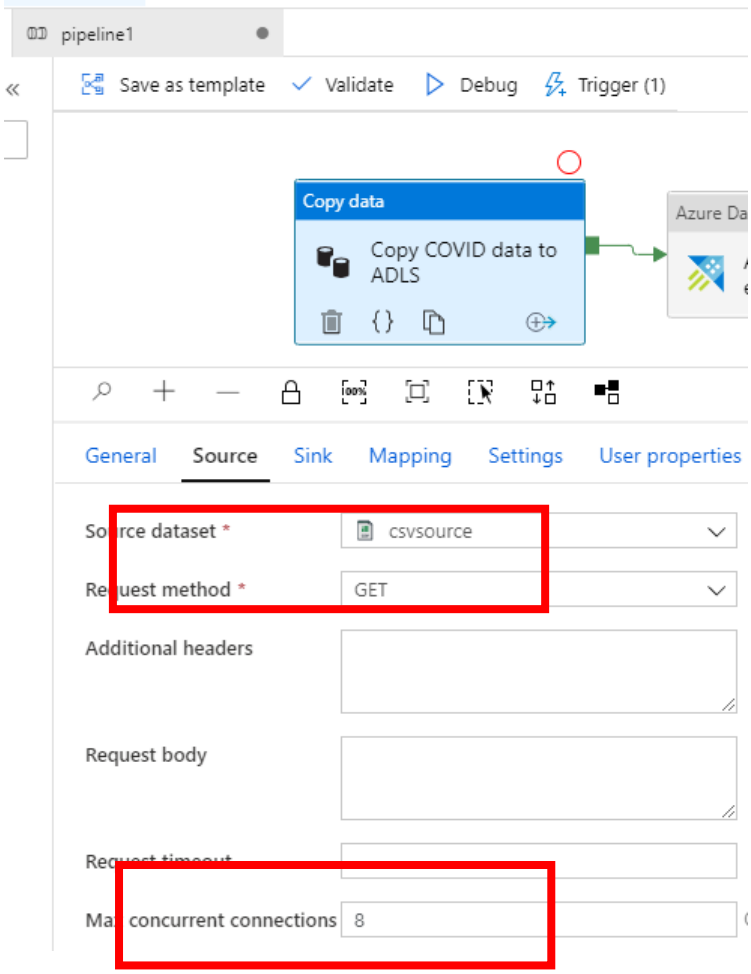
1. Drag **Copy data** activity, from the **Move & transform** collection, and drag to the pipeline canvas:



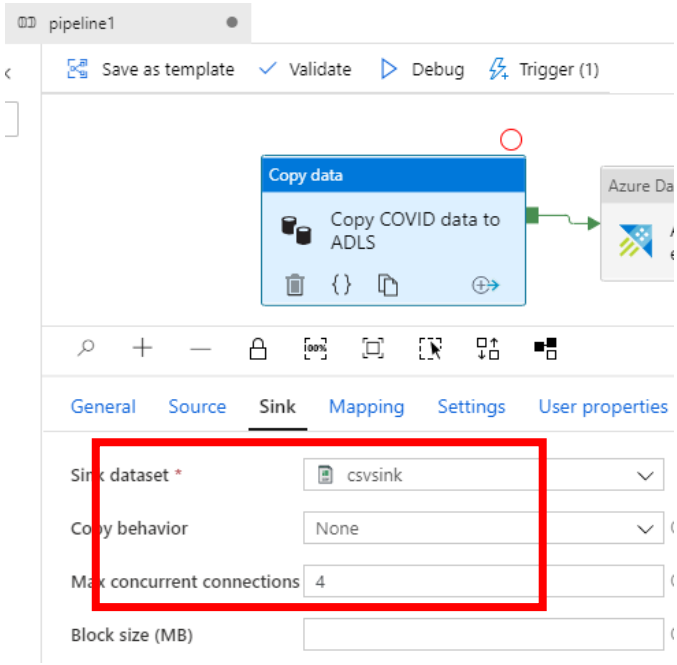
2. Complete the **General** / **Source** / **Sink / Mapping** tabs as per the following images.
3. **General** tab:



4. **Source** tab:

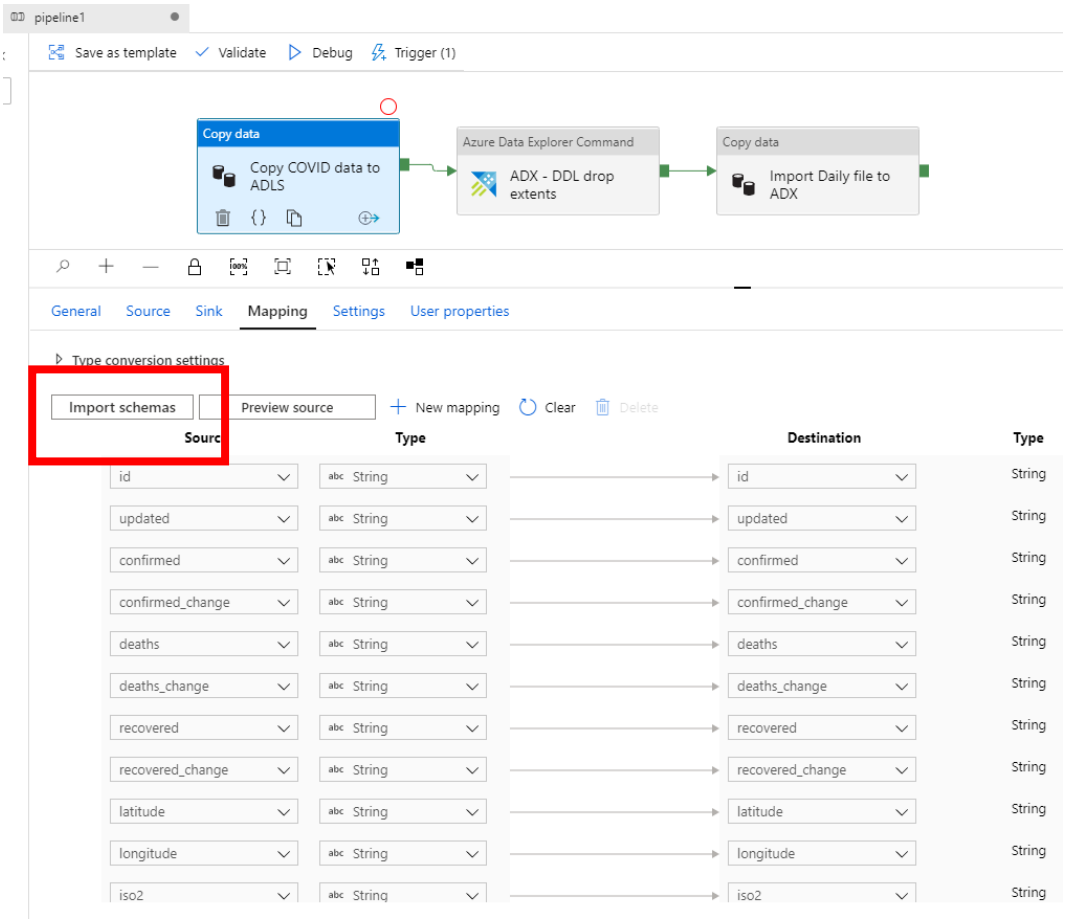


5. **Sink** tab:



6. **Mapping** tab

Select '**Import Schemas**' and then ensure all of the column definitions match:

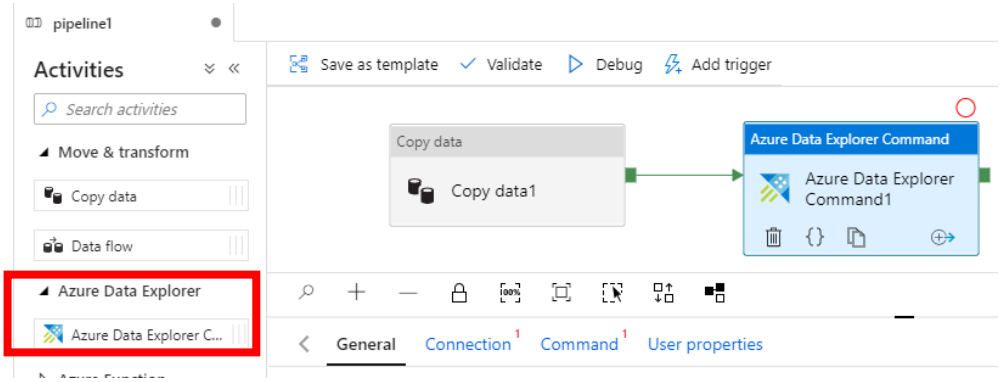


ADF 'ADX – DDL drop extents' activity

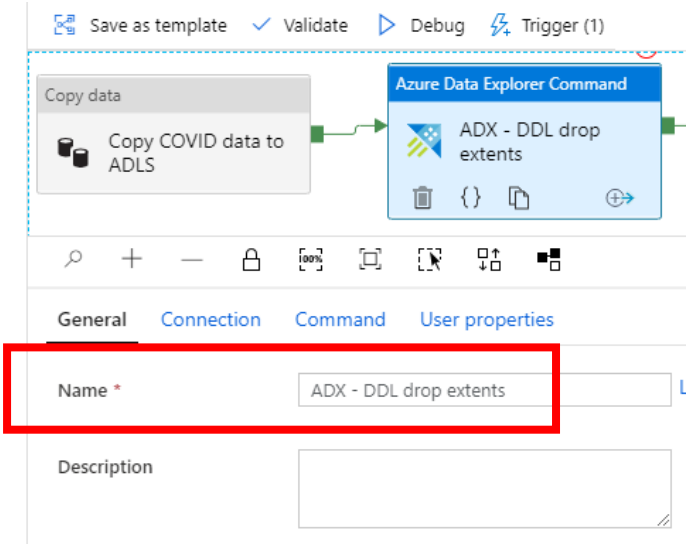
This activity runs native commands on the ADX cluster. For Workshop 1 it simply clears down the **covidstaging** table ready for the next data load using a KQL command.

The JSON file for this ADF activity is in the /download folder "ADF - ADX - DDL drop extents.json".

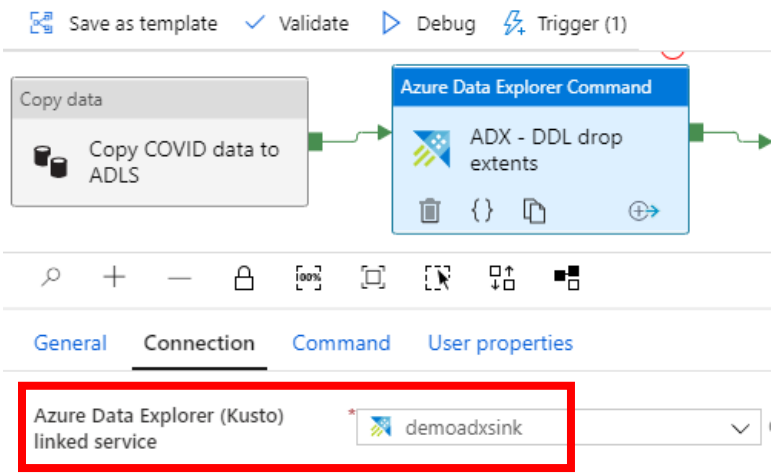
1. Drag **Azure Data Explorer Command**, from the **Azure Data Explorer** collection, to the pipeline canvas. Connect the two activities as shown:



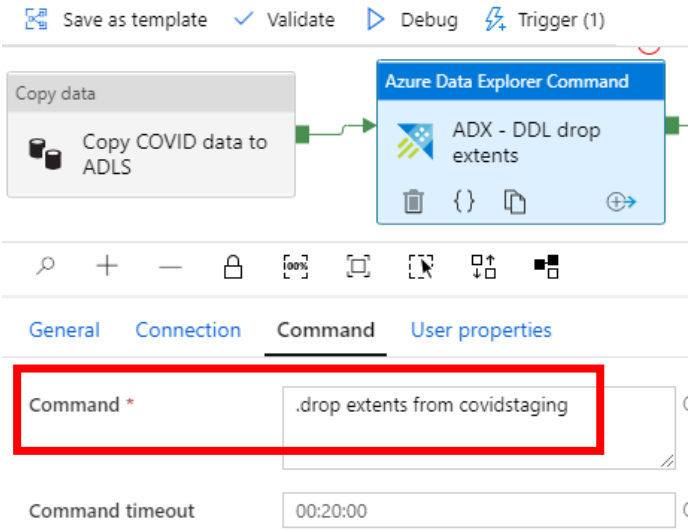
2. Complete the **General** / **Connection** / **Command** tabs as per the following images.
3. **General** tab:



4. **Connection** tab:



5. **Command** tab:



The KQL for the Command is:

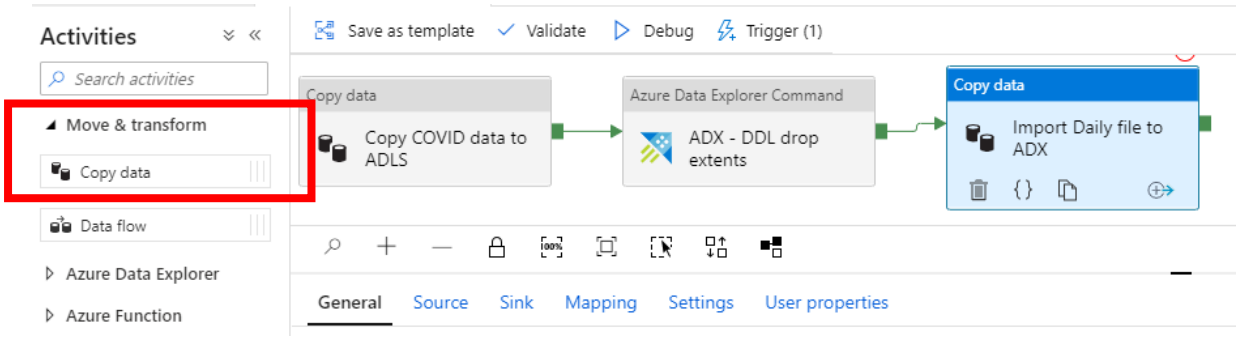
```
.drop extents from covidstaging
```

ADF 'Import Daily File to ADX' activity

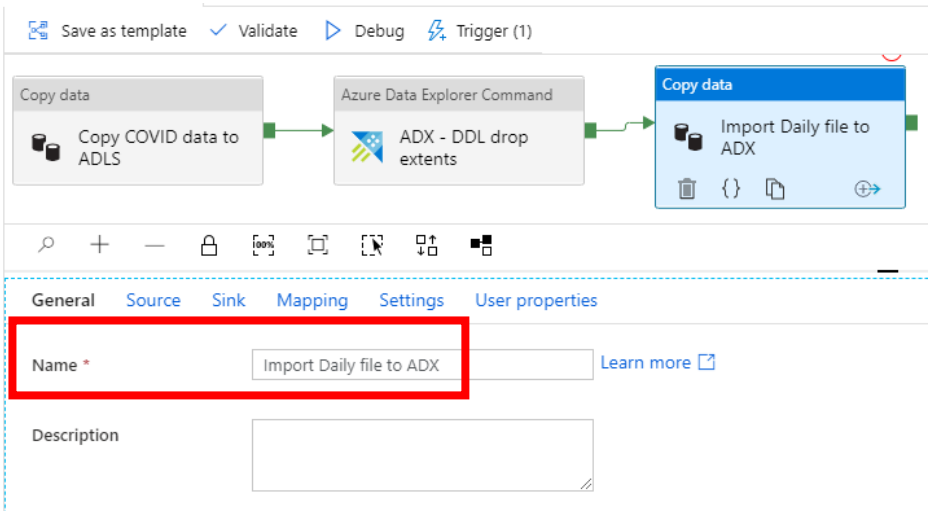
This activity loads the daily COVID-19 datafile into ADX.

The JSON file for this ADF activity is in the /download folder "ADF - Import Daily file to ADX.json".

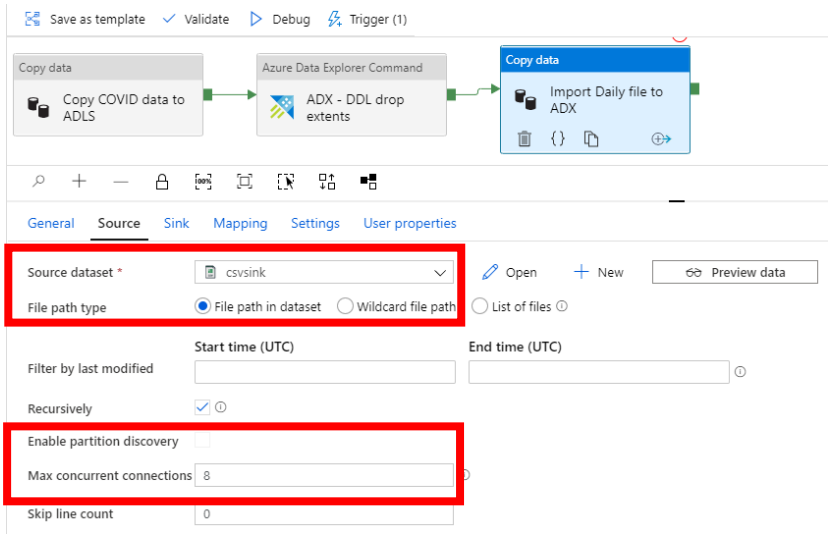
1. Drag **Copy data** activity, from the **Move & transform** collection, and drag to the pipeline canvas. Connect the activities as shown:



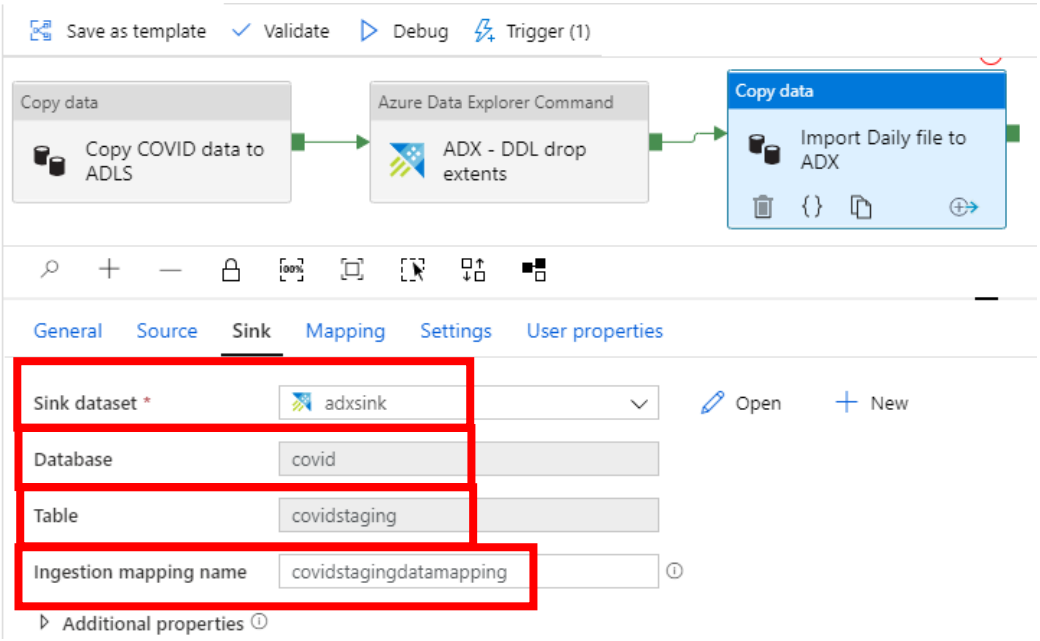
2. Complete the **General** / **Source** / **Sink** / **Mapping** tabs as per the following images.
3. **General** tab:



4. **Source** tab:


Save as template
Validate
Debug
Trigger (1)
Copy data
Copy COVID data to ADLS
Azure Data Explorer Command
ADX - DDL drop extents
Copy data
Import Daily file to ADX
General
Source
Sink
Mapping
Settings
User properties
Source dataset *
csvsink
Open
New
Preview data
File path type
File path in dataset
Wildcard file path
List of files
Filter by last modified
Start time (UTC)
End time (UTC)
Recursively
Enable partition discovery
Max concurrent connections
8
Skip line count
0

5. **Sink** tab:


Save as template
Validate
Debug
Trigger (1)
Copy data
Copy COVID data to ADLS
Azure Data Explorer Command
ADX - DDL drop extents
Copy data
Import Daily file to ADX
General
Source
Sink
Mapping
Settings
User properties
Sink dataset *
adxsink
Open
New
Database
covid
Table
covidstaging
Ingestion mapping name
covidstagingdatamapping
Additional properties

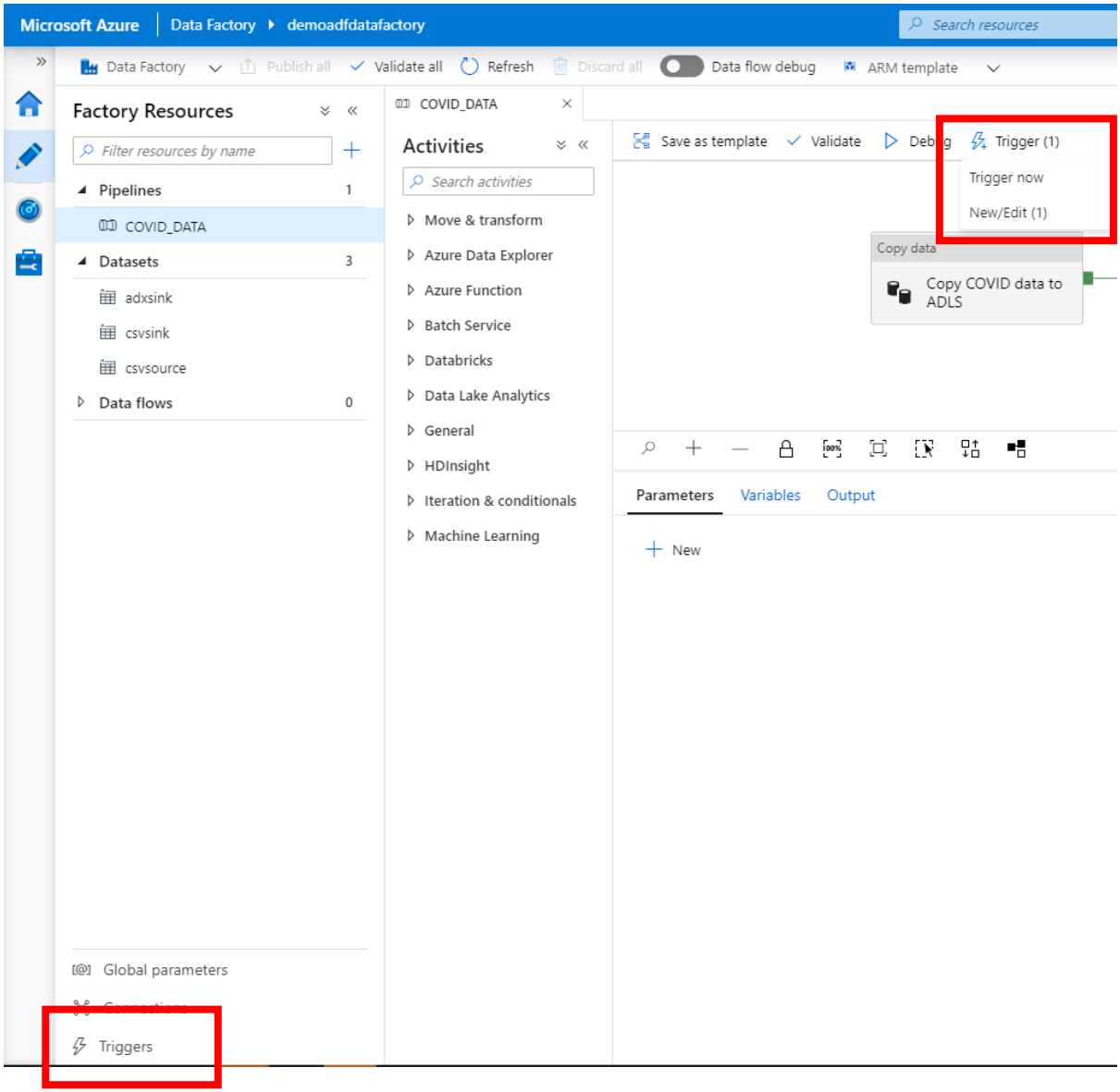
Create Pipeline Trigger

Triggers are how ADF Pipelines are executed. Triggers come in three forms:

- Event-based
- Scheduled
- Tumbling Window

In this workshop we will be using a scheduled trigger: https://docs.microsoft.com/en-us/azure/data-factory/how-to-create-schedule-trigger

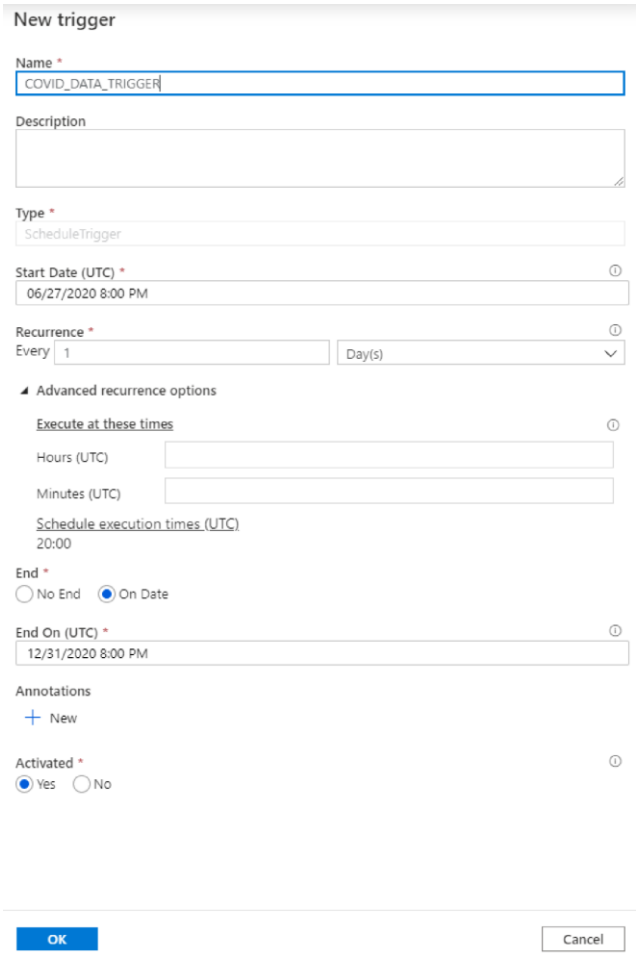
In the ADF UI a pipeline can be triggered manually, or a created from the selections as seen in the image below:



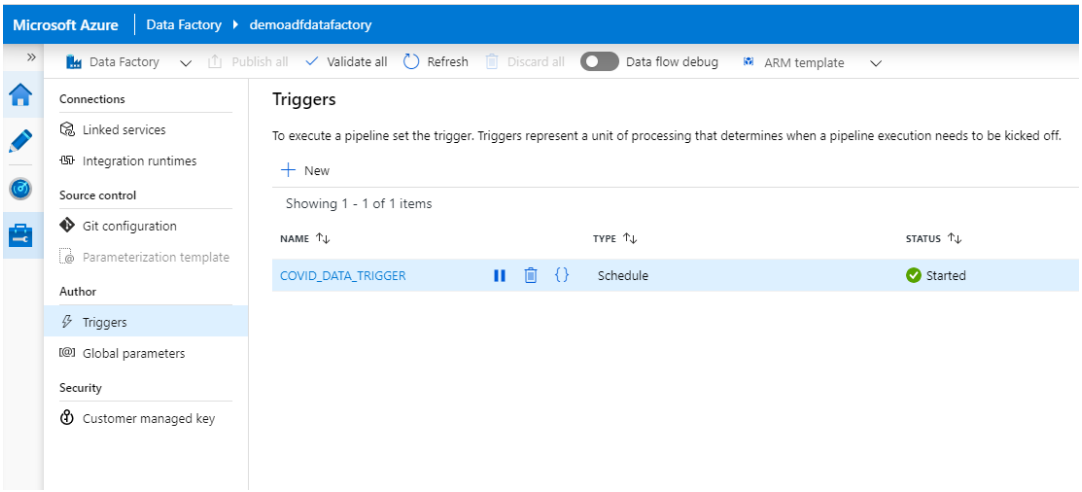
We will be creating a scheduled trigger 'COVID-DATA_TRIGGER' to run on a daily basis to execute the 'COVID_DATA' pipeline.

The JSON file for this ADF trigger is in the /download folder "COVID-DATA_TRIGGER.json".

1. From the COVID_DATA Pipeline canvas, select '**Trigger**', then '**New/Edit'** or **'+ New**', depending upon the access path. Complete the blade as shown in the image below:



2. Select '**OK**'
3. Triggers can then be seen in the '**Triggers**' blade:



Activity 4 – Security & Access

During the previous activities, and to complete the ADF pipeline build, we need to grant access to Azure services (ADX, ADF, ADLS). This is achieved via Azure Active Directory objects (App Registration, Service Principal, Managed Identity).

Terminology Explained

To assist with understanding the security principals within this workshop, *the document "Azure data Factory-Security.pdf" can be found in the /downloads folder.*

Application Registration

To delegate Identity and Access Management functions to Azure AD, an application must be registered with an Azure AD tenant. Refer to "Azure data Factory-Security.pdf" in /downloads.

> https://docs.microsoft.com/en-us/azure/active-directory/develop/howto-create-service-principal-portal

Managed Identities

> https://docs.microsoft.com/en-us/azure/active-directory/managed-identities-azure-resources/overview

Service Principal

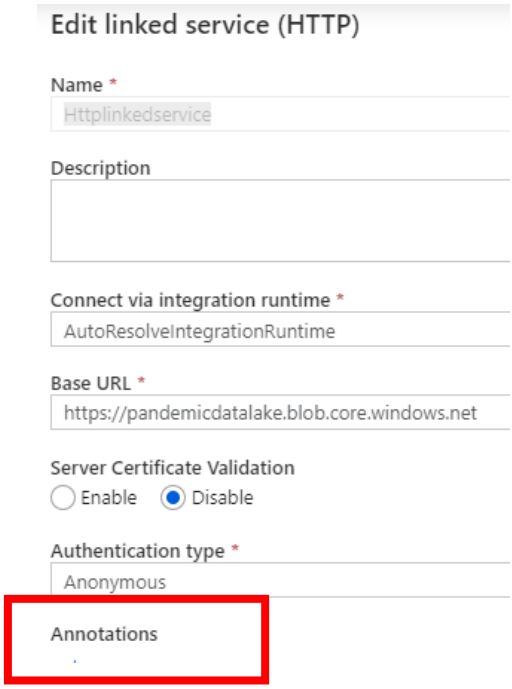
To access resources that are secured by an Azure AD tenant, the entity that requires access must be represented by a security principal. This is true for both users (user principal) and applications (service principal).

> https://docs.microsoft.com/en-us/azure/active-directory/develop/app-objects-and-service-principals#:~:text=A%20service%20principal%20must%20be%20created%20in%20each,Web%20application%2FAPI%20also%20has%20a%20service%20principal%20

ADF – Security Configuration (Linked Service)

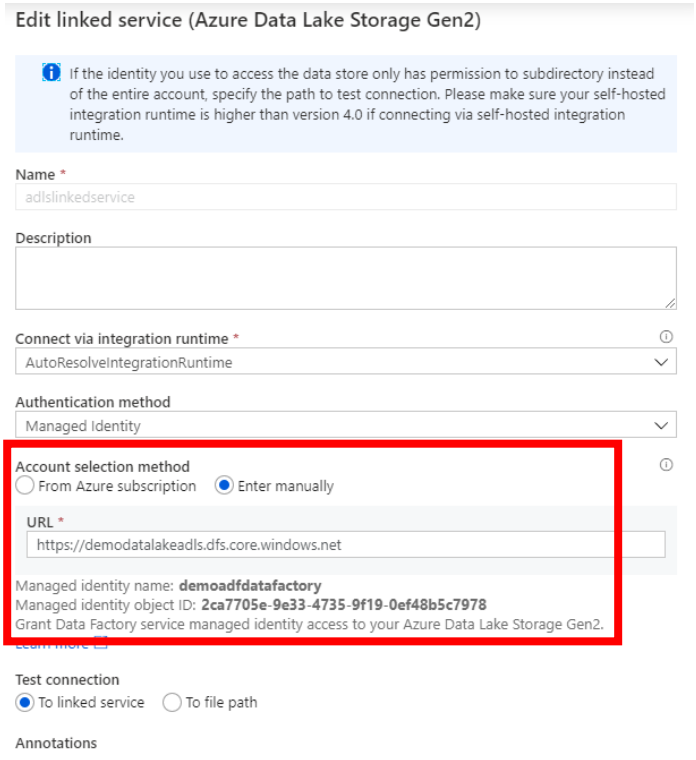
Httplinkedservice

This Linked Service, being a public hosted server, uses anonymous authentication. The configuration can be seen here:



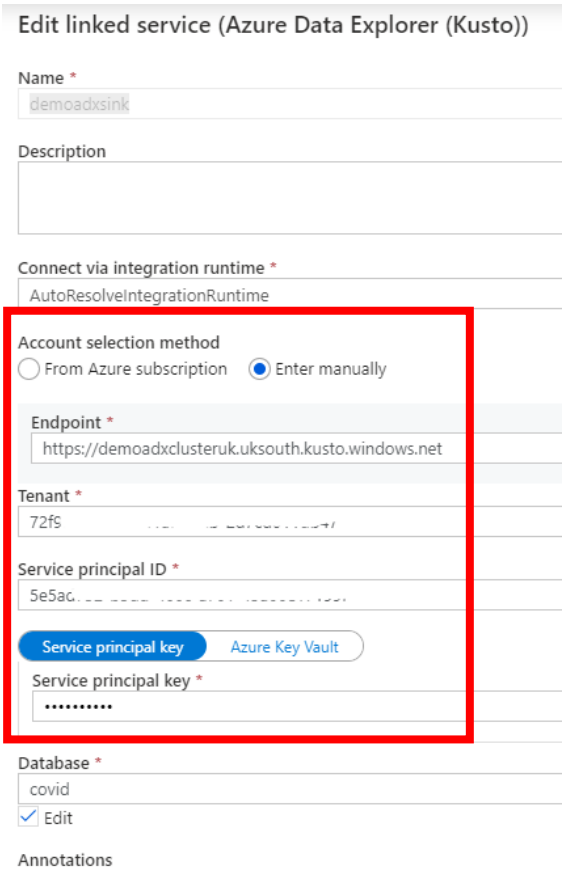
Adlslinkedservice

This Linked Service uses the ADF generated Managed Identity to authenticate to ADLS. Configuration is shown here:



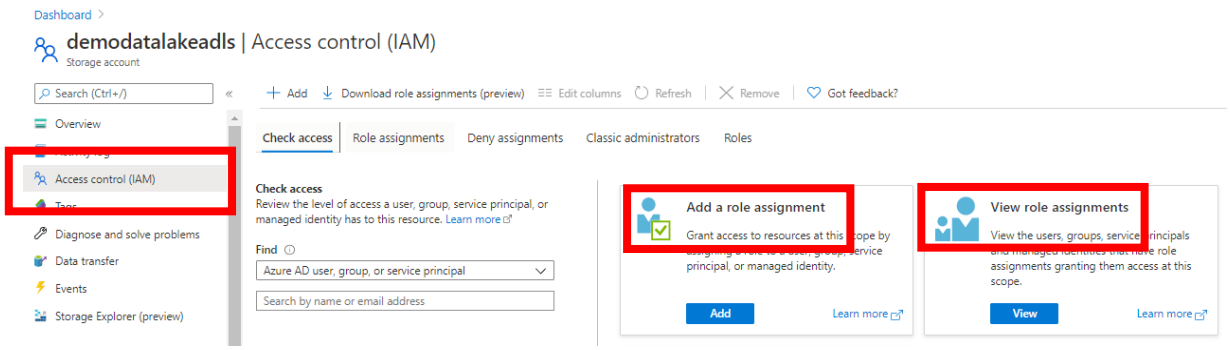
demoadxsink

This Service uses the Service Principal ***datademosecurity*** to authenticate, as shown here:

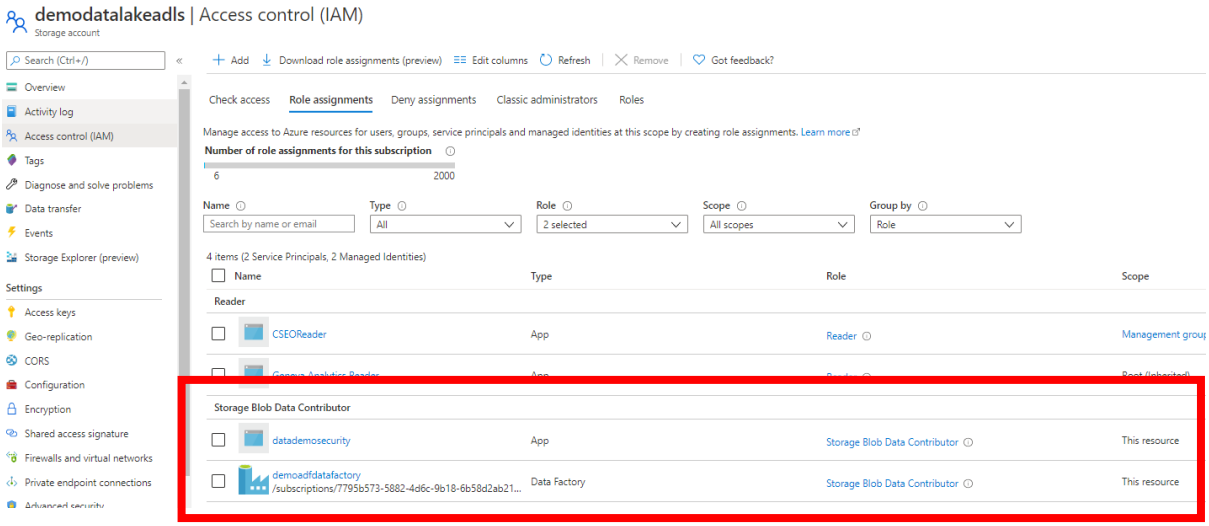


ADLS – Security Configuration

We will grant access to ADF (via ***demoadfdatafactory***) and ADX (via ***datademosecurity***). Security is managed via the 'Access control (IAM)' option off **thedemodatalakeadls** blade as shown below:

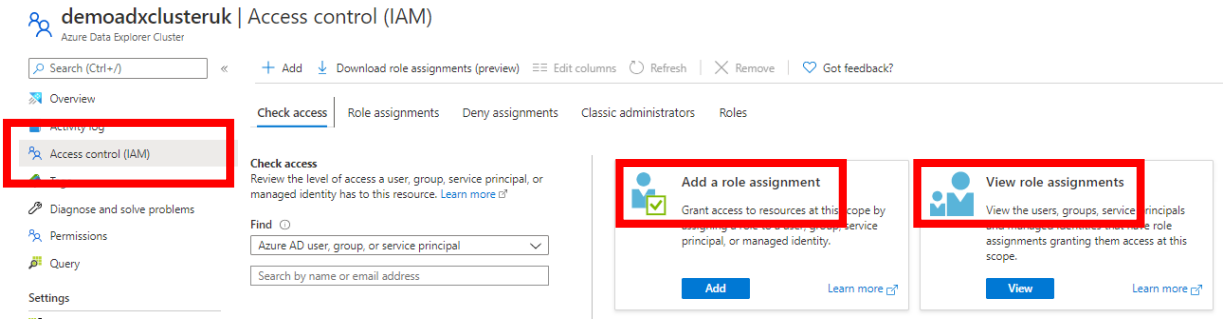


Permissions [that need to be] granted are shown here:

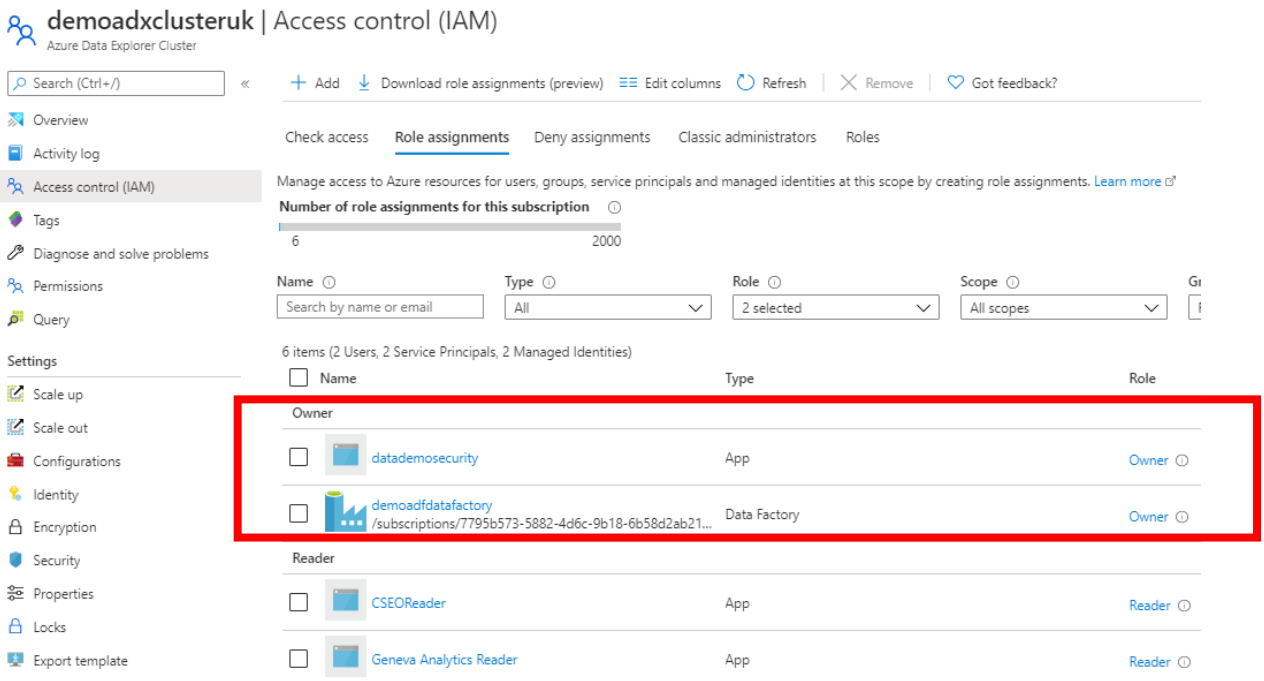


ADX – Security Configuration (cluster)

We will grant access to ADF (via ***demoadfdatafactory***) and ADX (via ***datademosecurity***). Security is managed via the 'Access control (IAM)' option off the **demoadxclusteruk** blade as shown below:



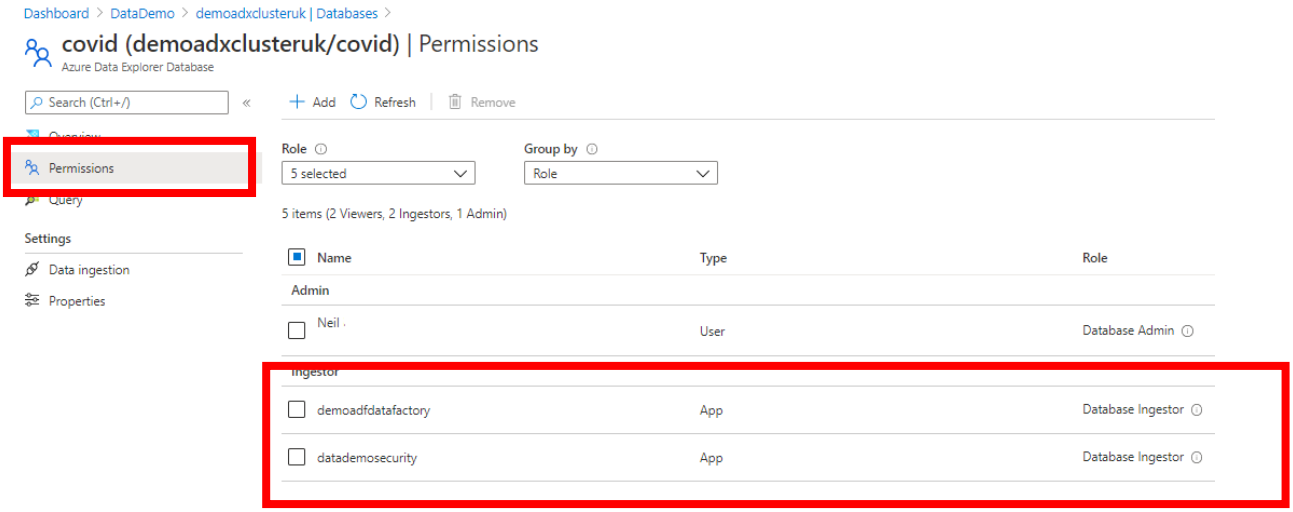
Permissions granted ('Owner is used as an example, 'Contributor' could also be selected) at the ADX cluster level are shown here:



Note AD access may be required for Azure Data Explorer Dashboard access. This should be granted 'Reader' role.

ADX – Security Configuration (database)

Permissions granted at the ADX database level are shown here:



Note AD access may be required for Azure Data Explorer Dashboard access. This should be granted 'Database Viewer' role.

Further Reading

Azure documentation around the security aspects of the services used within this workshop can be found in the following links:

ADF – Security Considerations

https://docs.microsoft.com/en-us/azure/data-factory/data-movement-security-considerations#:~:text=Data%20Factory%20management%20resources%20are%20built%20on%20Azure,grouping%20of%20activities%20that%20together%20perform%20a%20task.

ADLS – Access Control

https://docs.microsoft.com/en-us/azure/storage/blobs/data-lake-storage-access-control

ADX - Roles

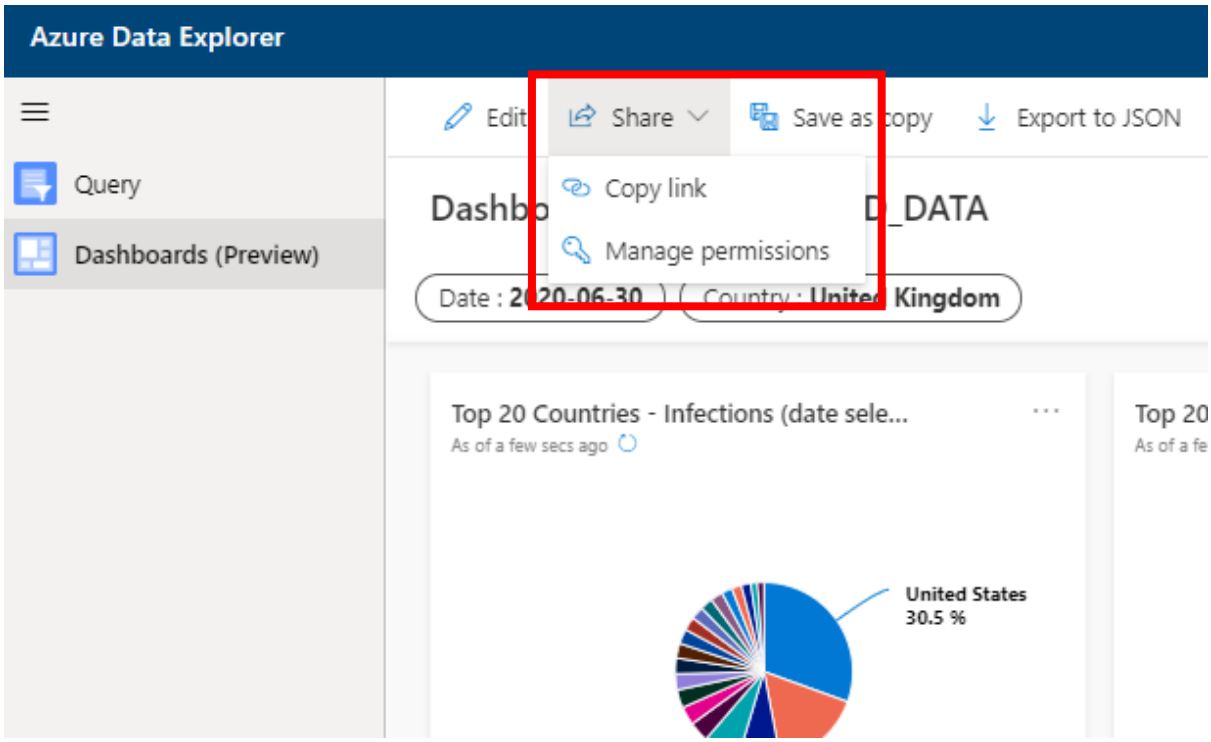
https://docs.microsoft.com/en-us/azure/data-explorer/kusto/management/security-roles

ADX – Managed Identity

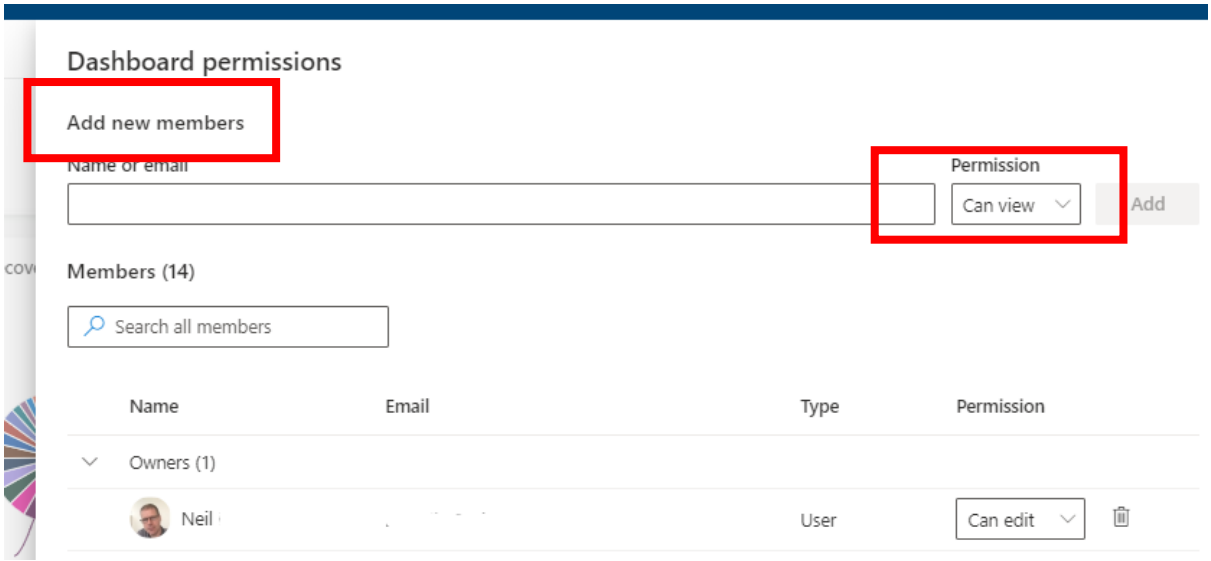
https://docs.microsoft.com/en-us/azure/data-explorer/managed-identities?tabs=portal

ADX Dashboard – Sharing

It is possible to share your ADX Dashboard. This is achieved via the '**Share**':



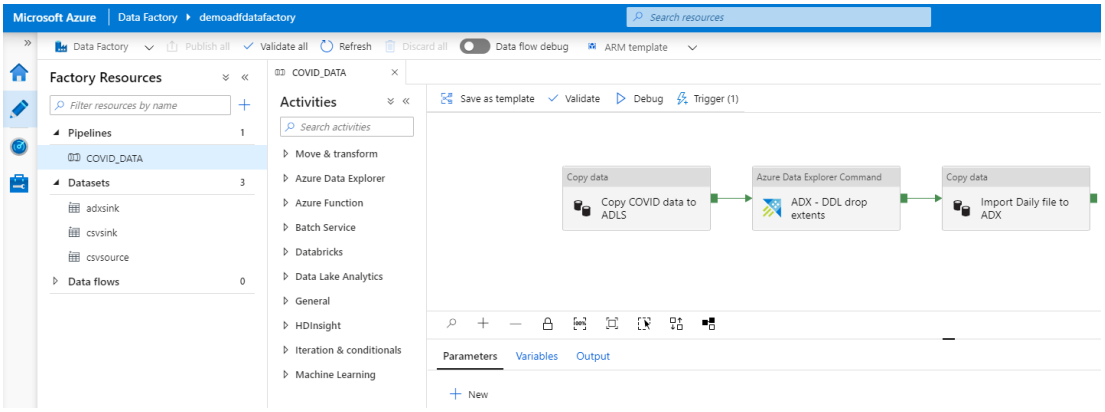
Selecting '**Manage Permissions**' allows edit or read access to be granted to a given Active Directory resource. Use '**Add new member**' and '**Permission**' to grant dashboard access, as shown below:



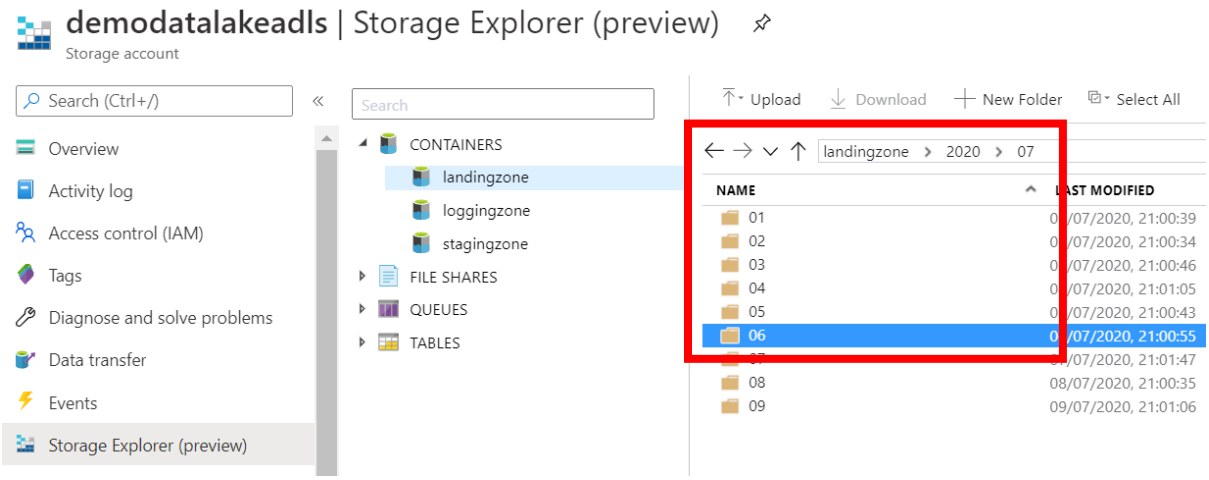
Remember ADX database [Reader permission will also need to be granted. See 'ADX – Security Configuration (database)' above.

Activity 5 – ADF Pipeline

We now have a completed ADF Pipeline '**COVID_DATA**':



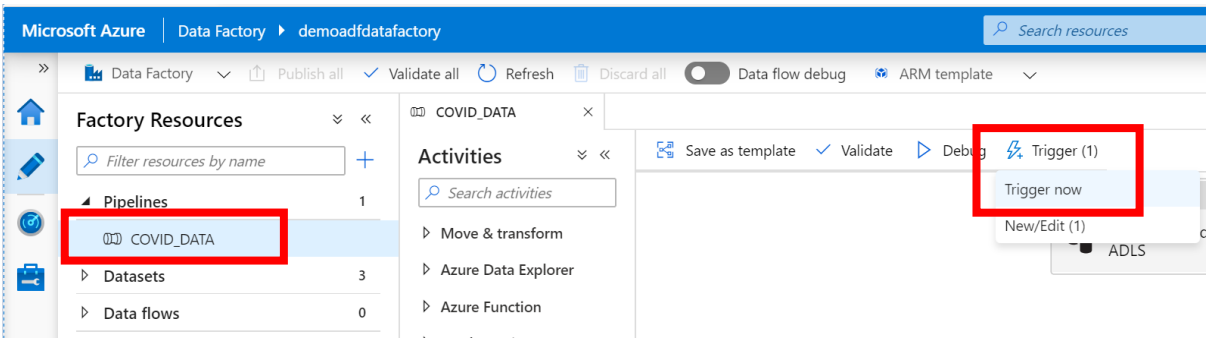
The scheduled trigger '**COVID_DATA_TRIGGER**' will run the ADF **'COVID_DATA'** landing COVID-19 files to ADLS in a date orientated hierarchy (landingzone/YYYY/MM/DD/). Repeat executions of the trigger(pipeline) will [on consecutive days] produce the same results but into landingzone/YYYY/MM/**DD + 1** where + 1 is the next/consecutive date, as shown in the image below [date shown = 6th July 2020):



Start/activate the ADF trigger

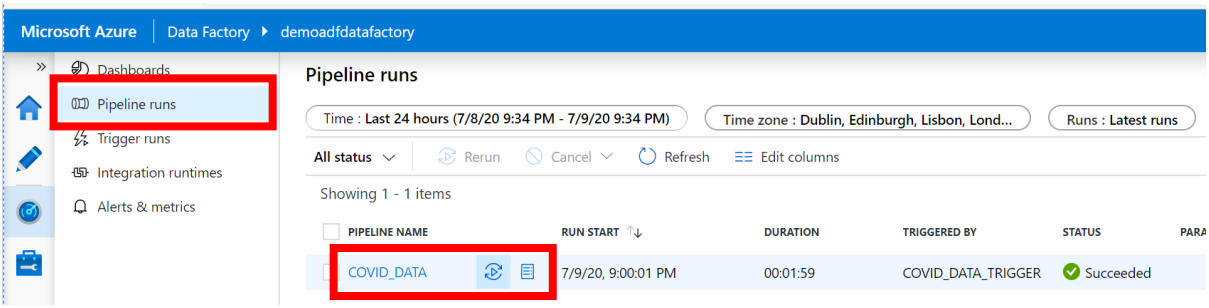
To test the ADF pipeline we need to either (a) manually trigger it or (b) run the pipeline through the ADF monitor.

For (a):



1. Select the **COVID_DATA** pipeline
2. Select '**Trigger**' / '**Trigger now**' option

For (b):

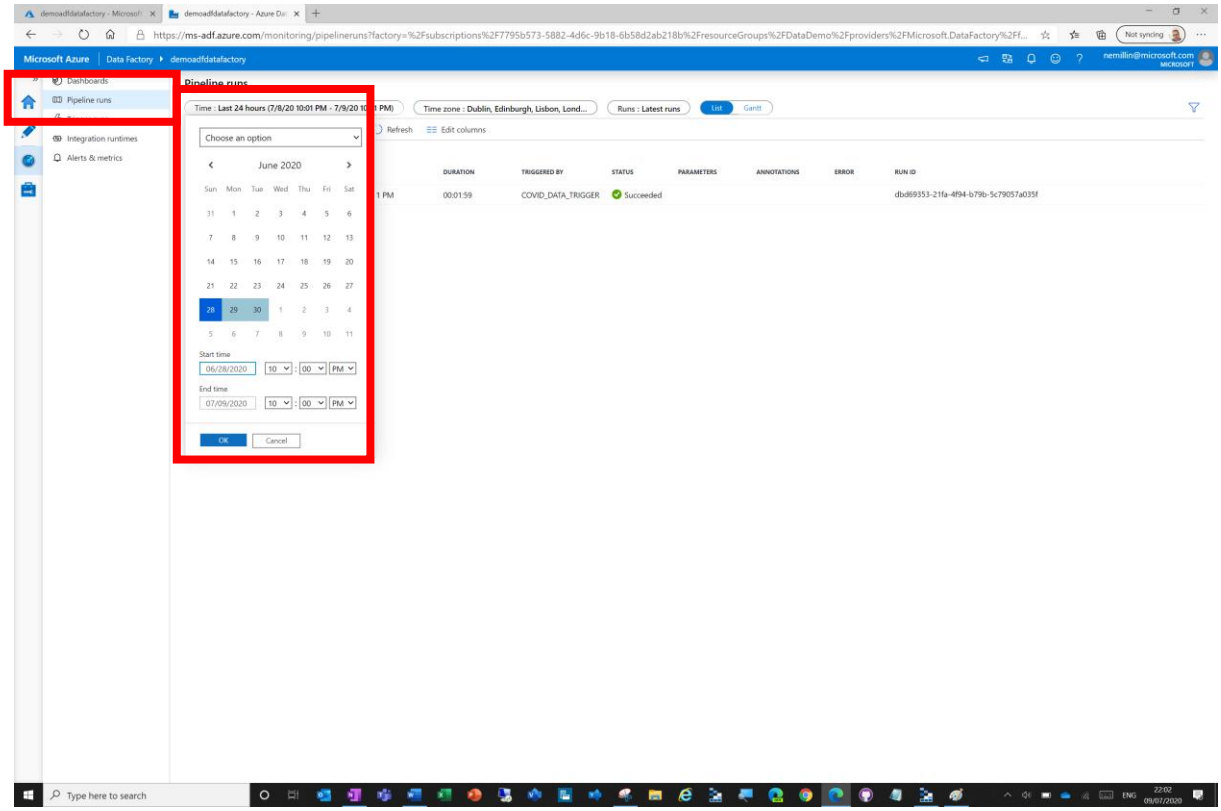


3. Select the '**Pipeline runs**'
4. Select the **COVID_DATA** pipeline
5. The '**Rerun**' icon will appear, select this to execute the pipeline

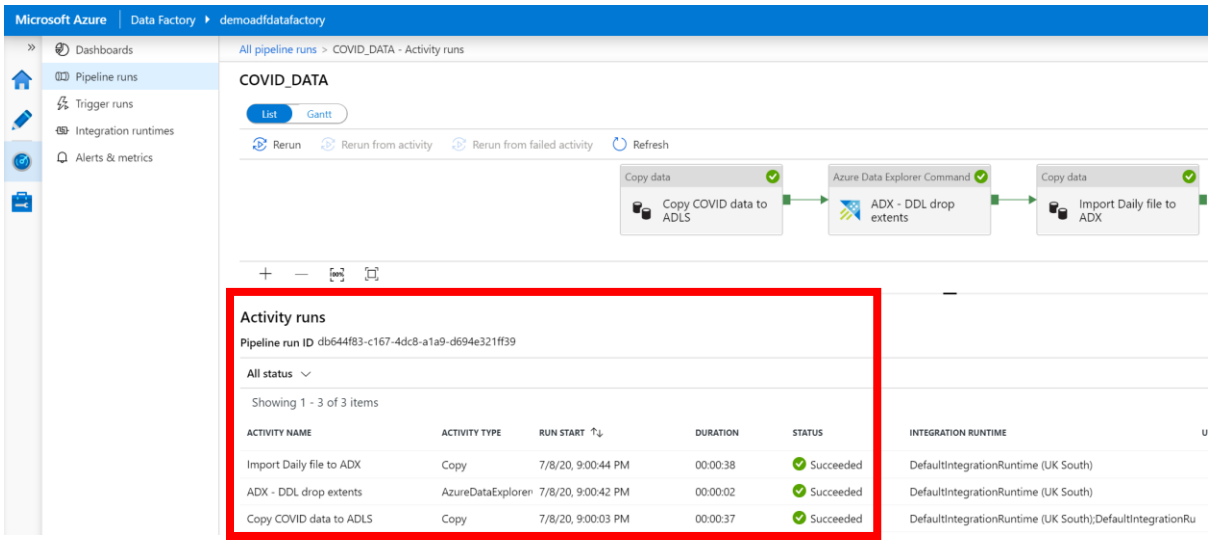
Monitor ADF Pipeline execution

We can monitor the progress (and historical executions) and status of the ADF pipeline:

1. Select **Pipeline runs**
2. Select the date range you want to monitor:



3. ADF will now show the historical pipeline runs. Selecting the required pipeline run, or the currently executing pipeline, will open a view displaying the individual pipeline activities:



Familiarise yourself with monitoring, and reviewing, ADF pipeline activity:
https://docs.microsoft.com/en-us/azure/data-factory/monitor-visually

Activity 6 – Analytics via KQL

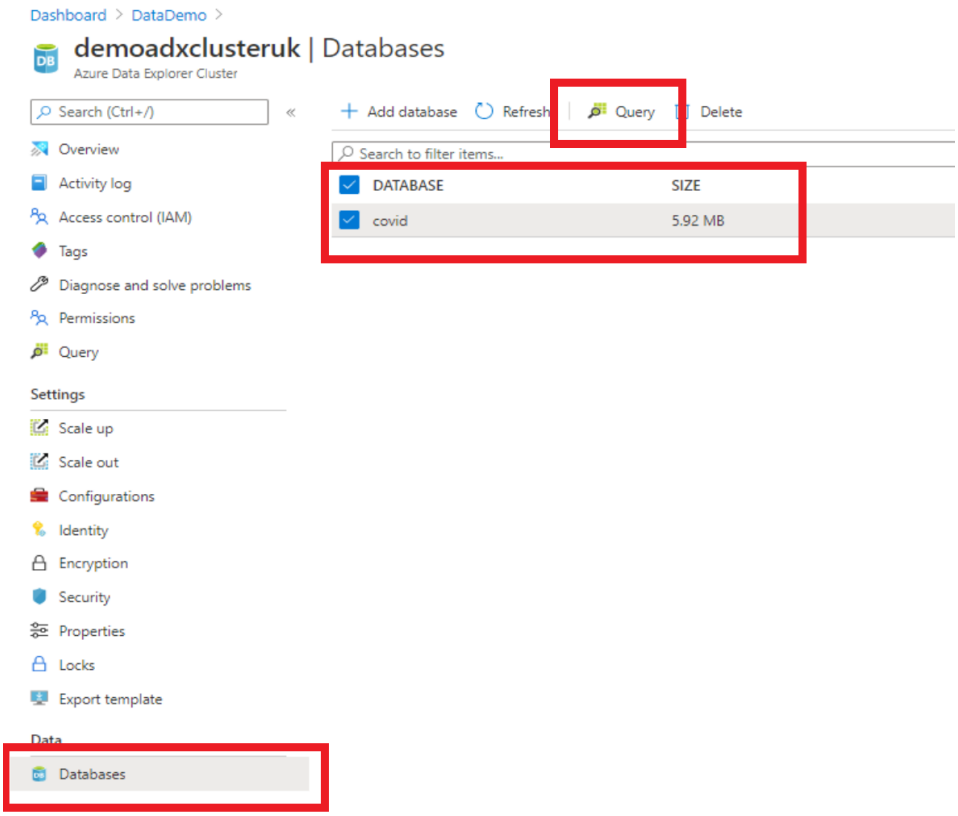
We can now use KQL to query data ingested [via ADF] to ADX**.**

ADX KQL queries

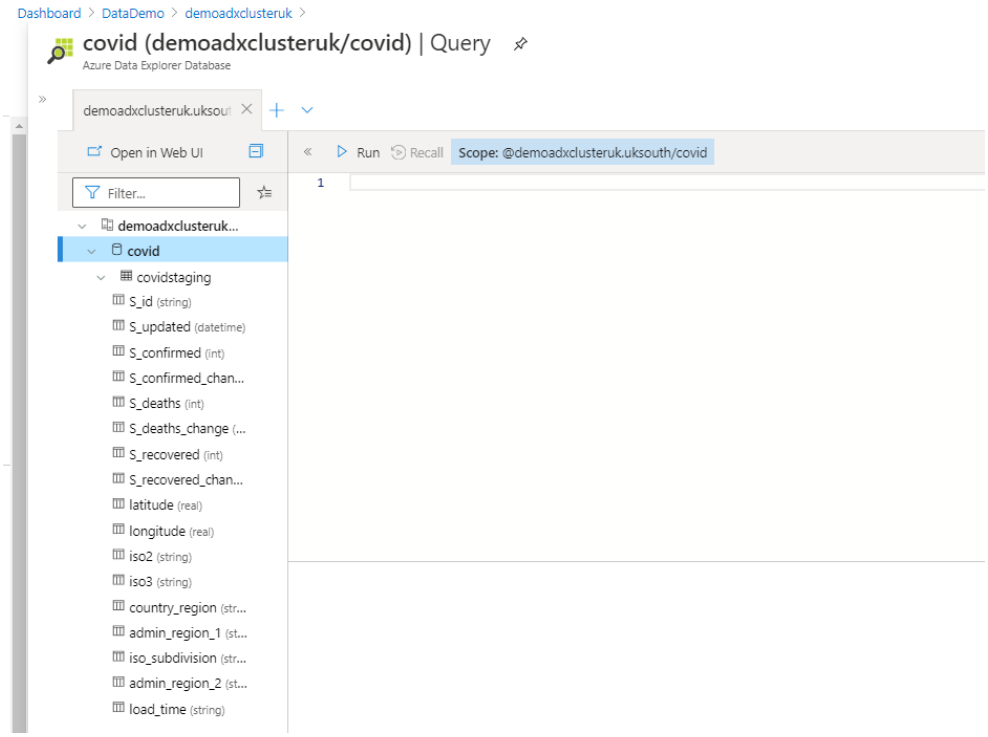
We can use the Azure Portal ADX Query for this activity or the URL: https://dataexplorer.azure.com/

For the Azure Portal:

1. Select your resource group
2. Select the ADX cluster
3. Select **Databases**, select the **covid** database:



This will then open an ADX KQL query window:



Basic Analytics

So, lets run some basic analytics. To facilitate with this section the following links will help understand KQL:

- Pluralsight: https://www.pluralsight.com/courses/kusto-query-language-kql-from-scratch
- Pluralsight: https://www.pluralsight.com/courses/microsoft-azure-data-explorer-advanced-query-capabilities
- https://docs.microsoft.com/en-us/sharepoint/dev/general-development/keyword-query-language-kql-syntax-reference

All KQL used in this section can be found in the /downloads folder – "KQL – Query tool.kql"

With the database name highlighted (as per image above) type the following KQL and select **Run**:

```
covidstaging
| where country_region == 'Worldwide'
| where S_confirmed > 0
```

This will show you the data for 'Worldwide' geography, on a daily basis with a positive *S_confirmed* count. Adding in the *Project* will permit specific columns to be selected:

```
covidstaging
```

```
| where country_region == 'Worldwide'
| where S_confirmed > 0
| project S_updated, S_confirmed, S_confirmed_change
```

The following command will select all data for ‘New Zealand’ geography:

```
covidstaging
| where country_region == 'New Zealand'
```

Play with the data, selecting specific dates or geographies to become familiar with its schema and structure. The following command will show the **covidstaging** schema:

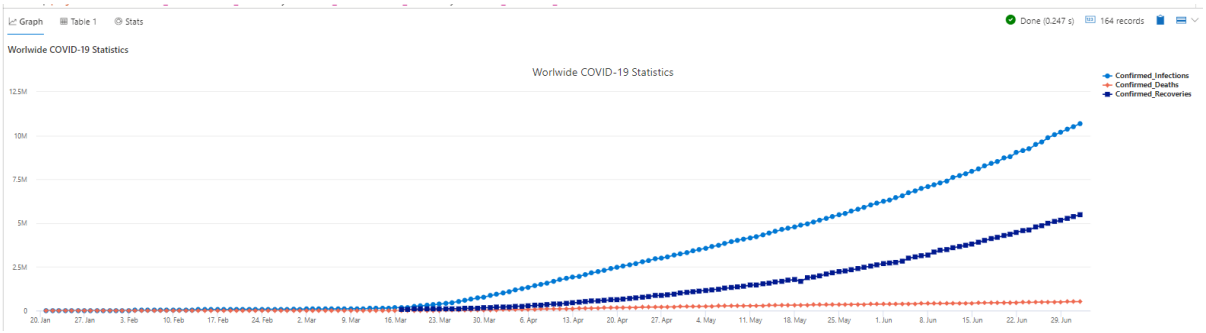
```
.show table covidstaging
```

We need KQL queries that are more sophisticated and and/or visual. The following takes the ‘Worldwide’ geography data and visualises it as a line chart. *Render* activates KQL query graphical visualisations:

(https://docs.microsoft.com/en-us/azure/data-explorer/kusto/query/renderoperator?pivots=azuredataexplorer)

```
// queries - Linechart - worldwide, all dates
covidstaging
| where country_region == 'Worldwide'
| project   S_updated, S_confirmed, S_deaths, S_recovered
| render linechart with (title = "Worlwide COVID-19 Statistics")
| project-rename Confirmed_Infections = S_confirmed, Confirmed_Recoveries =
S_recovered, Confirmed_Deaths = S_deaths
```

Notice how *project-rename* has been used to tidy the column names displayed:



Toggle the ‘Graph’, ‘Table’ and ‘Stats’ tab views to see the underlying data and ADX performance statistics:

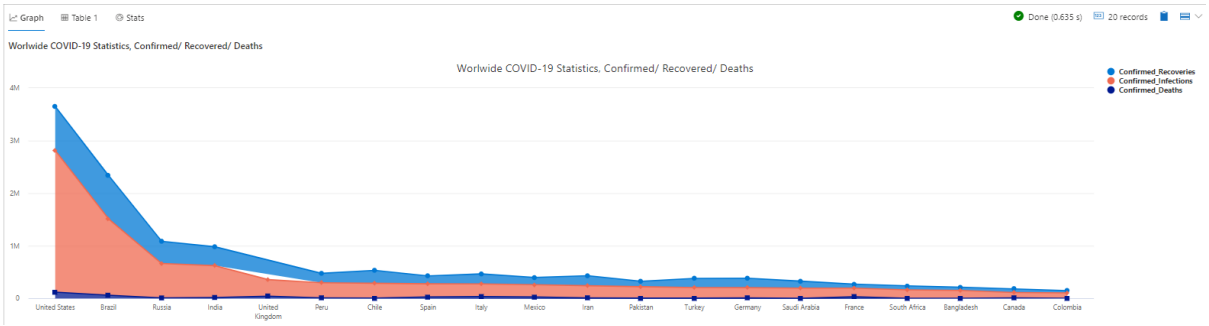
Increasing the KQL sophistication will give greater sophistication to the analytics. Introducing the *let* command allows a temporary table to be built to determine the maximum date:

```
let maxDateTable = covidstaging | where country_region == 'Worldwide' |  order by S_updated desc | take 1 | project S_updated;
```

If we join this temporary table to our main KQL, for *country_region* data, we can take the *top 20* confirmed cases count:

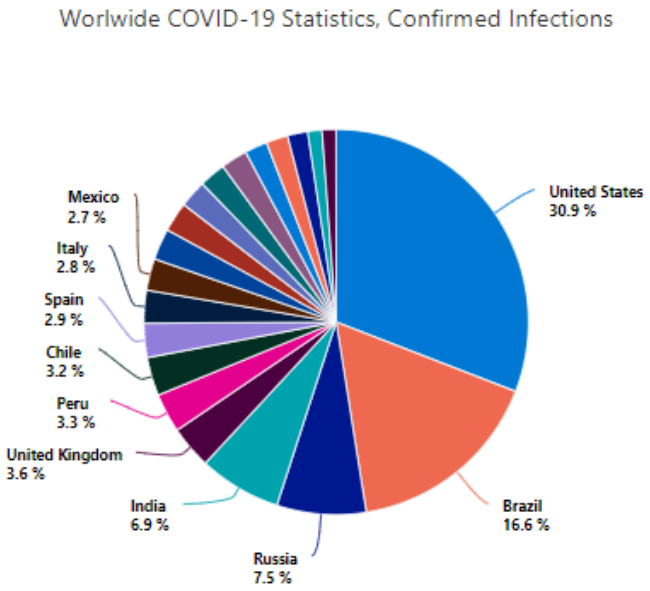
```
// queries - stacked area chart - country, latest date

let maxDateTable = covidstaging | where country_region == 'Worldwide' |  order by S_updated desc | take 1 | project S_updated;
covidstaging
| join kind= inner  (maxDateTable) on $left.S_updated == $right.S_updated
| where country_region != 'Worldwide'
| where admin_region_1 =='' 
| project country_region, S_recovered, S_confirmed, S_deaths
| order by S_confirmed desc
| render stackedareachart with (  title = "Worlwide COVID-19 Statistics, Confirmed/ Recovered/ Deaths" , xcolumn = country_region)
| project-rename Confirmed_Deaths = S_deaths, Confirmed_Infections = S_confirmed, Confirmed_Recoveries = S_recovered
| take 20
```



Using visualisations can have a large impact on the understanding of the data. For example, making some minor changes to the KQL above and utilising a pie chart gives a clearer picture of infections by *country_region*:

```
// queries - piechart - country, latest date, confirmed
let maxDateTable = covidstaging | where country_region == 'Worldwide' |  order by S_updated
desc | take 1 | project S_updated;
covidstaging
| join kind= inner  (maxDateTable) on $left.S_updated == $right.S_updated
| where country_region != 'Worldwide'
| where admin_region_1 =='
| project country_region,  S_confirmed
| order by S_confirmed desc
| render piechart  with (  title = "Worlwide COVID-19 Statistics, Confirmed Infections" ,
xcolumn = country_region, legend = hidden)
| project-rename Confirmed_Infections = S_confirmed
| take 20
```

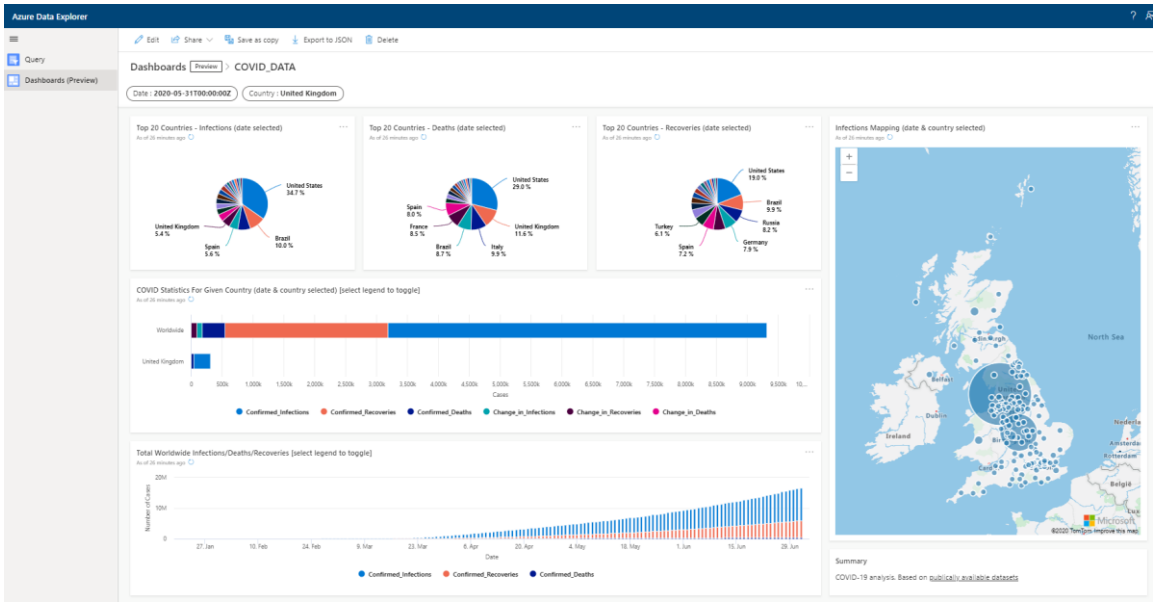


Activity 7 – Analytics via ADX Dashboards

Using the ADX query tool is a great way to performs "what if" analysis over your data, but it assumes a knowledge of KQL is required. If this is not the case, or you need to build a sophisticated dashboarding [with secure sharing capabilities] experience then the Azure Data Explorer Dashboards can be utilised.

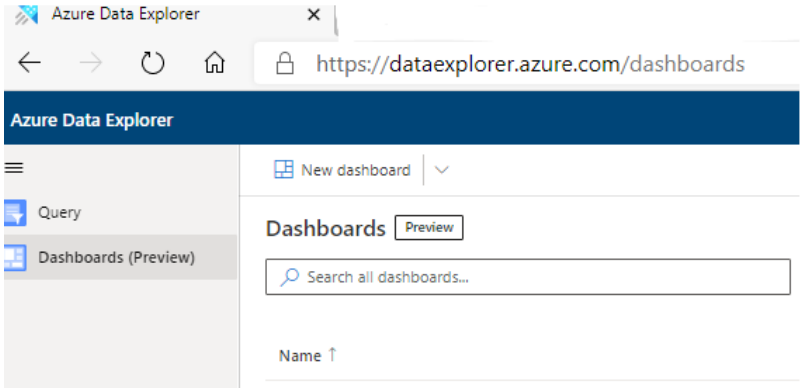
An explanation of (the new) Azure Data Explorer Dashboards can be found here: https://docs.microsoft.com/en-us/azure/data-explorer/azure-data-explorer-dashboards#:~:text=%20Visualize%20data%20with%20Azure%20Data%20Explorer%20dashboards,support%20a...%205%20Next%20Steps.%20%20More%20

We will now build the following Azure Data Explorer Dashboard:

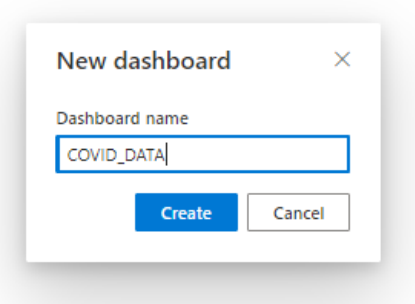


Create Dashboard

Type the ADX Dashboard URL: https://dataexplorer.azure.com/dashboards

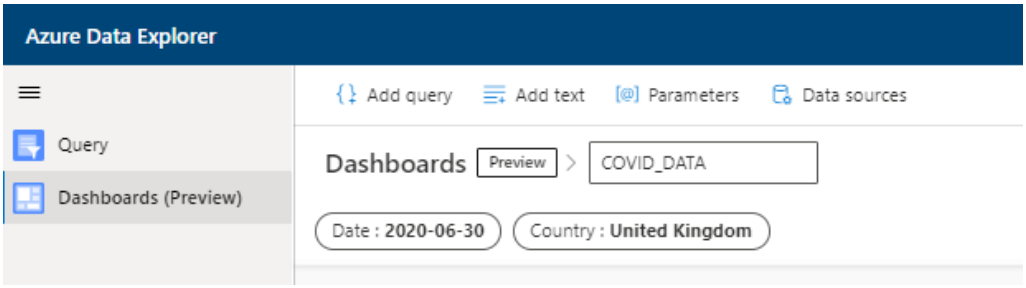


Select '**+ New**' and enter the Dashboard name '**COVID_DATA'**:



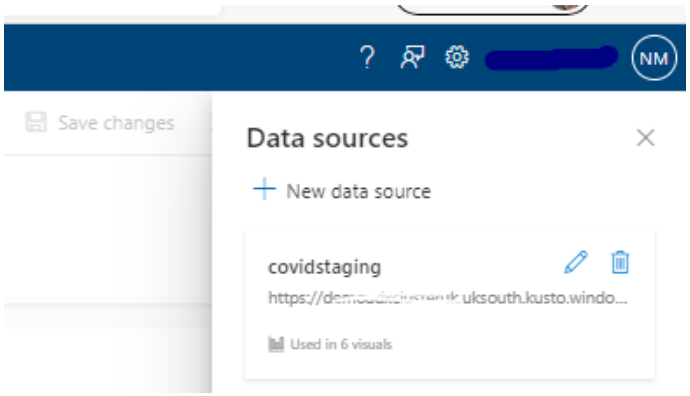
Data sources

- Select '**Data sources'**:



- Select '**+ New data source**'
- Enter the ADX Cluster URI (see section Activity 2)
- Select '**Connect'**
- Select '***covid'*** in the listbox
- Enter the Data source name: '***covidstaging***'
- Click '**Apply'**

This will create the data source:



Create Parameters

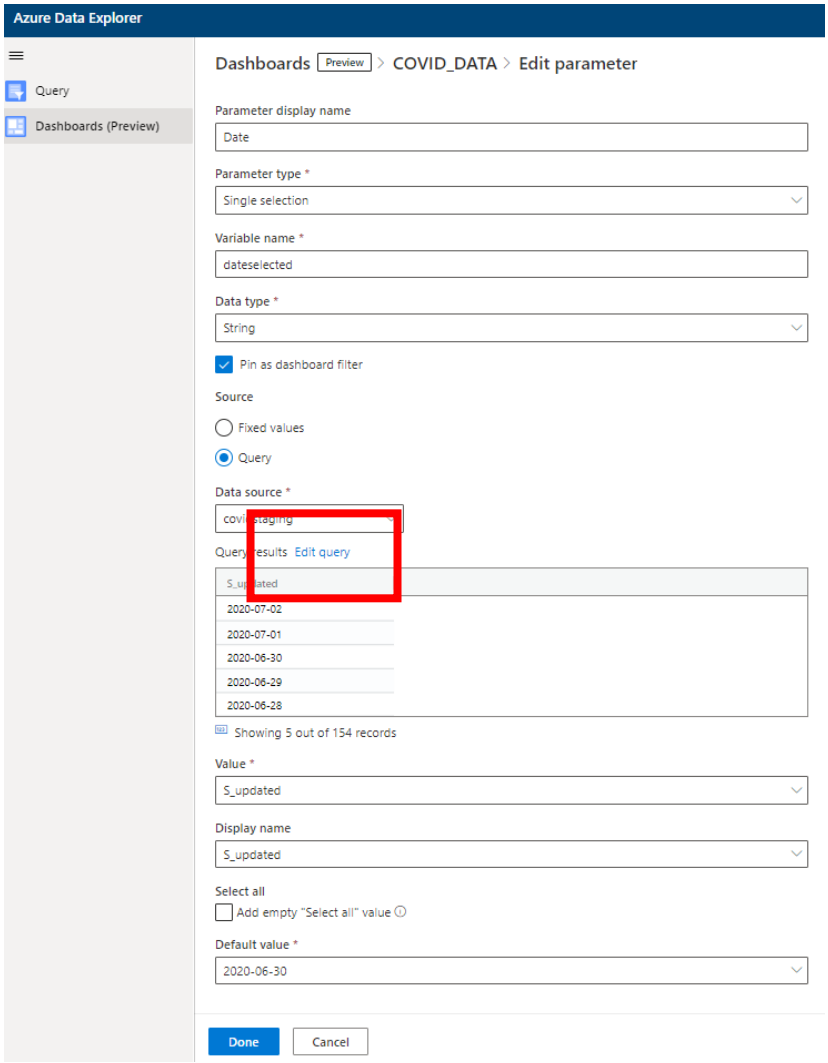
We will now create the Dashboard parameters that will drive the Dashboard visualisations:

An explanation of ADX Dashboard parameters is here: https://docs.microsoft.com/en-us/azure/data-explorer/dashboard-parameters

Date Parameter

The **Date** parameter allows the dashboard to be context sensitive to the selected date value.

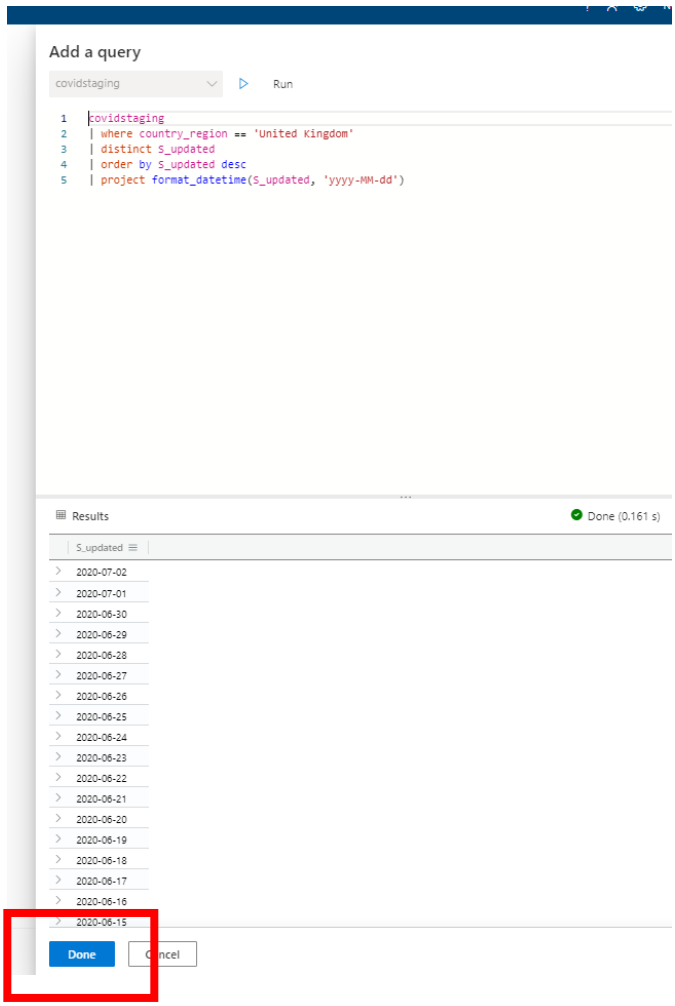
- Select '**Parameters**'
- Select '**+ New parameter**'
- Enter the fields as shown in the image below. The '**edit query**' KQL is shown beneath also:



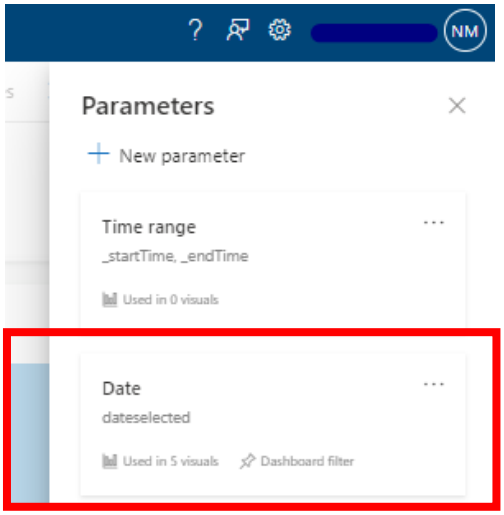
- Select '**edit query**' and enter the KQL below:

```
covidstaging
| where country_region == 'United Kingdom'
| distinct S_updated
| order by S_updated desc
| project format_datetime(S_updated, 'yyyy-MM-dd')
```

- Select 'Done' to save the KQL date query into the date parameter definition:



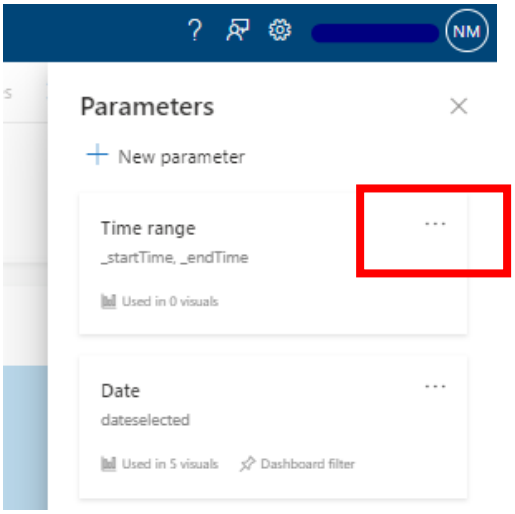
- Select '**Done**' to then save the **Date** parameter itself. You should now have created the **Date** parameter:



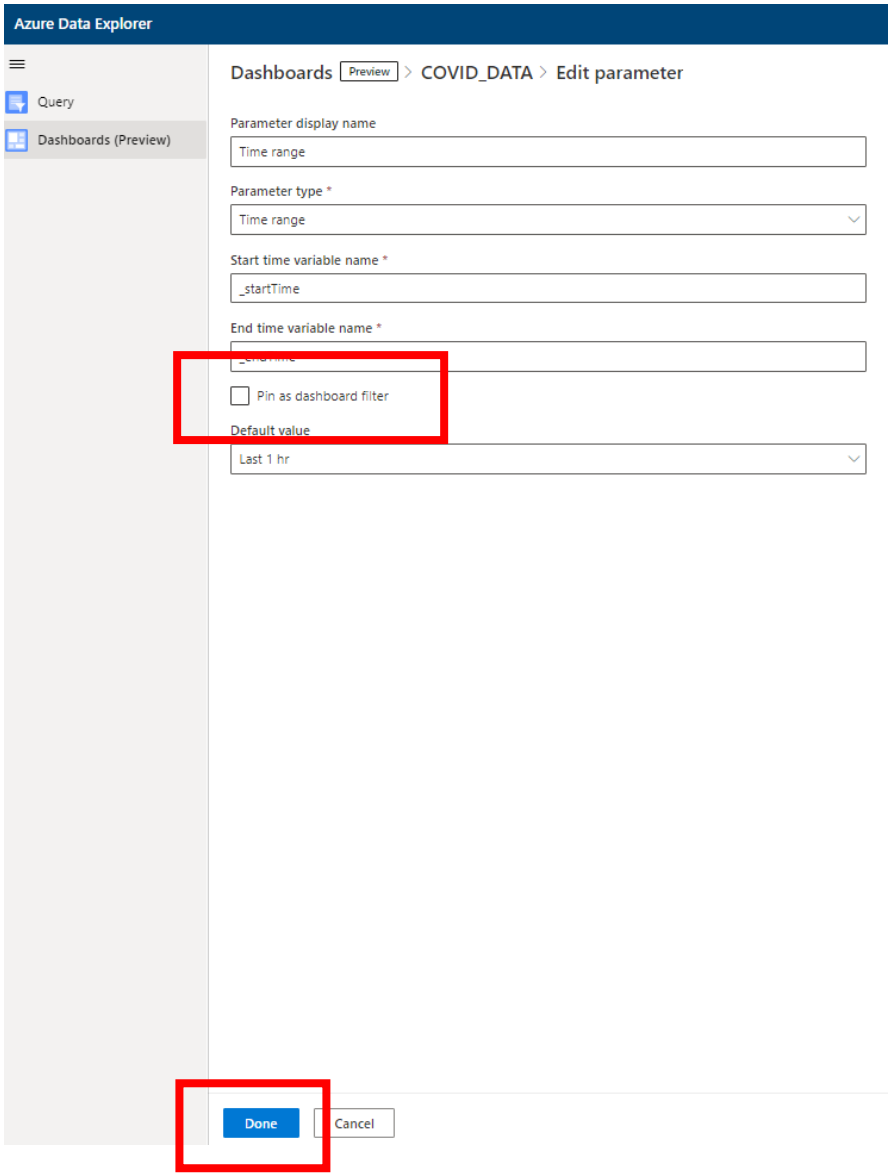
Time Range Parameter

This parameter may appear as a default parameter on your dashboard. It is not required for the COVID-19 data (or dashboard) as the data is date (day, 24hr) based. We will, therefore, disable this option from the dashboard:

- Next to the '**Time range**' parameter definition, select '**…**' and '**edit**' :



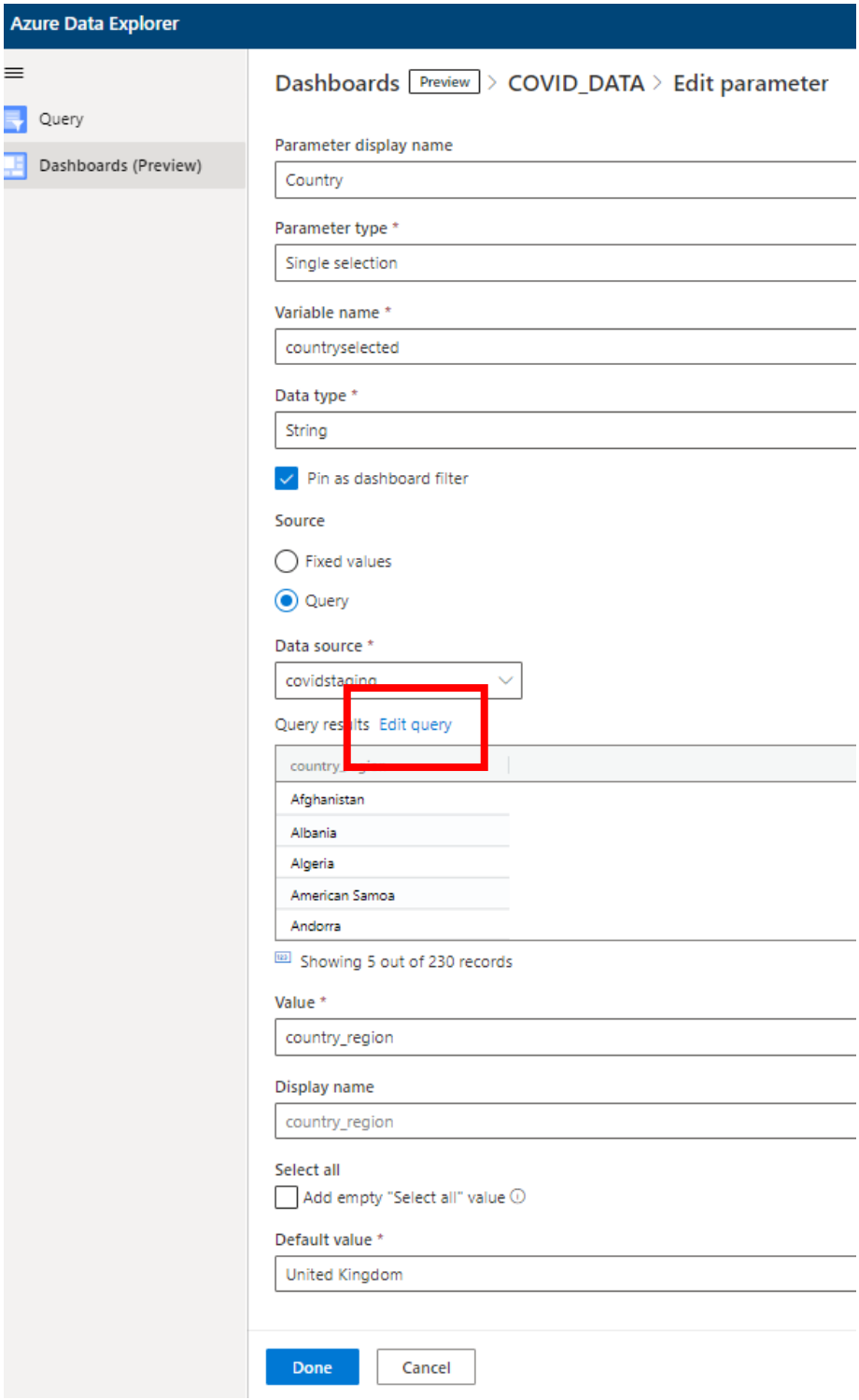
- Unselect the 'Pin as dashboard filter' option
- Select 'Done' to save:



Country Parameter

The **Country** parameter allows the dashboard to be context sensitive to the selected country value.

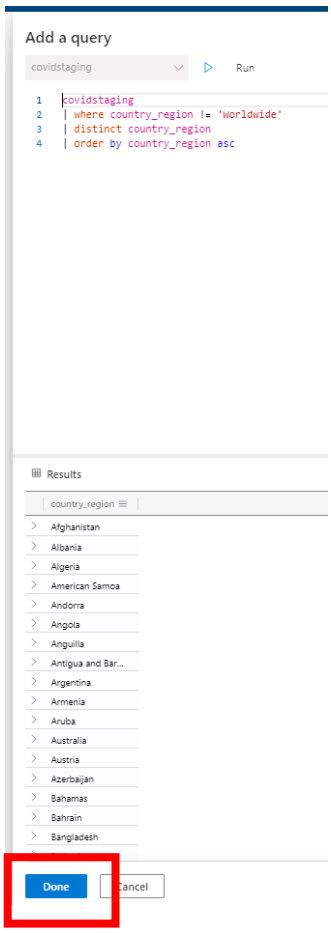
- Select '**Parameters**'
- Select '**+ New parameter**'
- Enter the fields as shown in the image below. The '**edit query**' KQL is shown beneath:



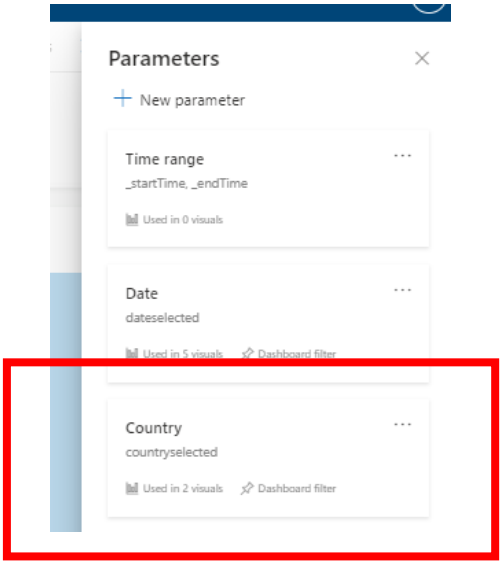
- Select '**edit query**' and enter the KQL below:

```
covidstaging
| where country_region != 'Worldwide'
| distinct country_region
| order by country_region asc
```

- Select ‘Done’ to save the KQL date query into the date parameter definition:



- Select ‘**Done**’ to then save the **Country** parameter itself. You should now have created the **Country** parameter:


Parameters
New parameter
Time range
_startTime, _endTime
Used in 0 visuals
Date
dateselected
Used in 5 visuals
Dashboard filter
Country
countryselected
Used in 2 visuals
Dashboard filter

Dashboard Visualisations

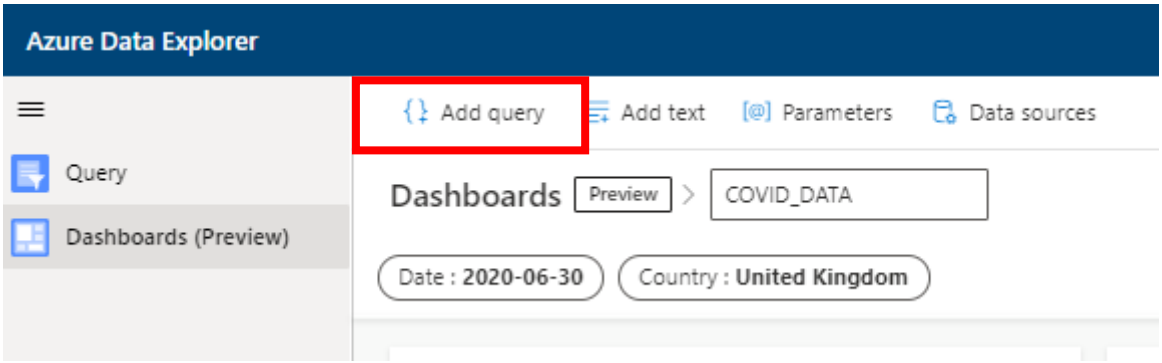
We can now add report visualisations to the dashboard.

The following KQL queries will be created. **All KQL can be found in the /downloads container in this repository. It is saved under the corresponding name.**

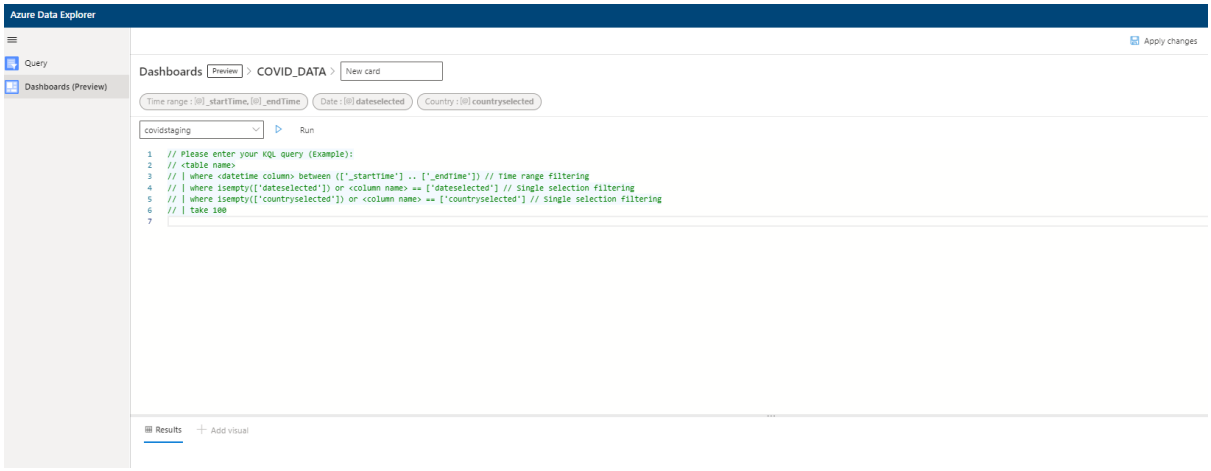
a) Top 20 Countries – Infections (date selected) = **KQL_dashboard_1.kql**
b) Top 20 Countries – Deaths (date selected) = **KQL_dashboard_2.kql**
c) Top 20 Countries – Recoveries (date selected) = **KQL_dashboard_3.kql**
d) COVID Statistics (date & country selected) = **KQL_dashboard_4.kql**
e) Total Worldwide Infections/Deaths/Recoveries = **KQL_dashboard_5.kql**
f) Infections Mapping (date & country selected) = **KQL_dashboard_6.kql**

Create a Dashboard Visualisation

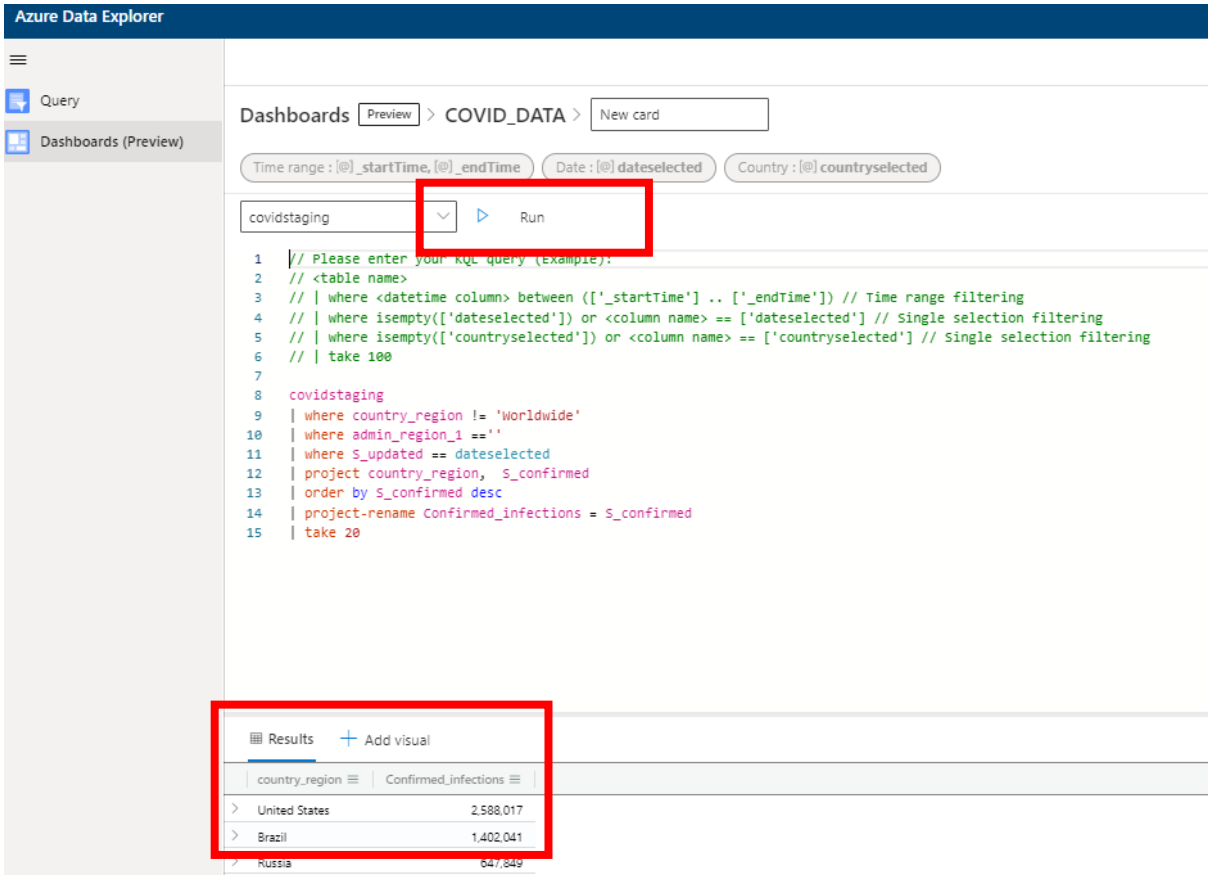
- Select '**{} Add Query**'



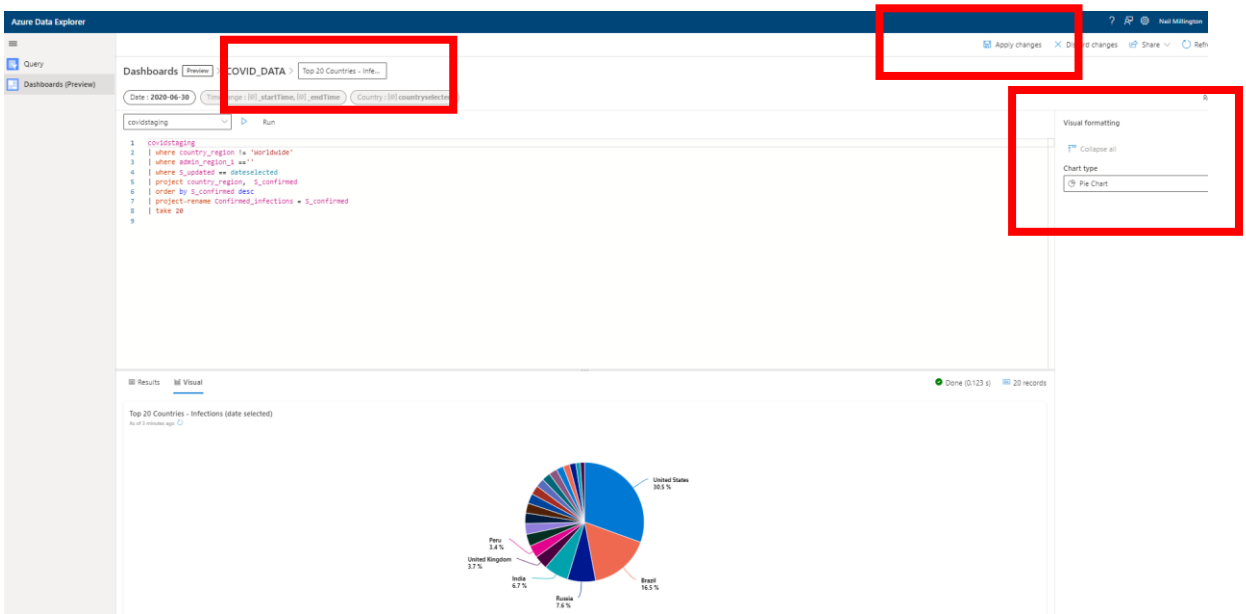
- Insert the KQL into the query panel as per the list above



- Select '**Run**'
- Once the results are displayed from the query in '**Results**' tab, select '**+ Add visual**':



- Enter the visualisation card title in the panel, and,
- Select the required visual under '**Visual Formatting**'. In this case [KQL_dashboard_1.kql] this is a pie chart:



- Select '**Apply Changes**' to save the visualisation card. Resize and position the visualisation on the dashboard as needed.

Remaining Dashboard Visualisations

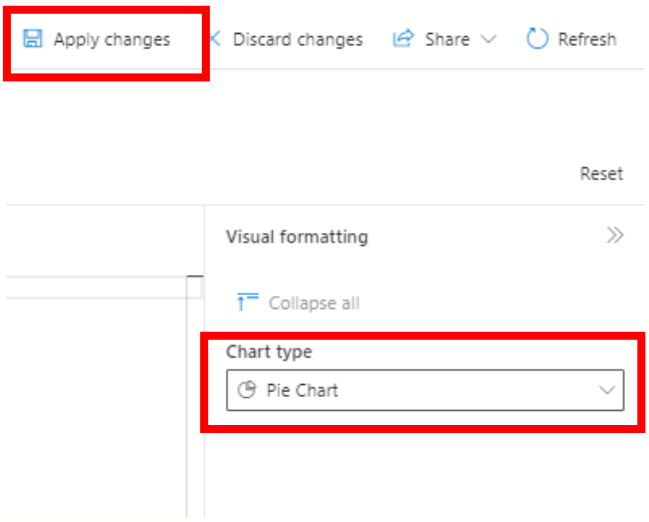
Repeat the above steps [in section '**Create a Dashboard Visualisation**'] for the remaining visualisations:

a) Top 20 Countries – Infections (date selected) = **KQL_dashboard_1.kql** ****created ****
b) Top 20 Countries – Deaths (date selected) = **KQL_dashboard_2.kql**
c) Top 20 Countries – Recoveries (date selected) = **KQL_dashboard_3.kql**
d) COVID Statistics (date & country selected) = **KQL_dashboard_4.kql**
e) Total Worldwide Infections/Deaths/Recoveries = **KQL_dashboard_5.kql**
f) Infections Mapping (date & country selected) = **KQL_dashboard_6.kql**

Specific visualisation properties are shown as follows:

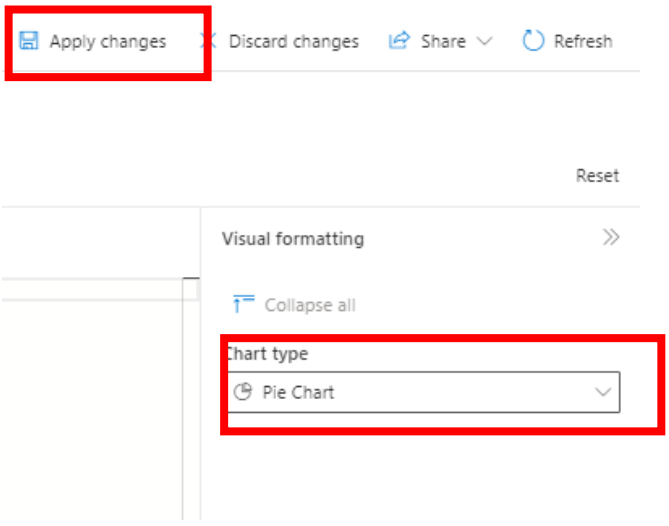
b) Top 20 Countries – Deaths (date selected)

Pie chart – as per (a):



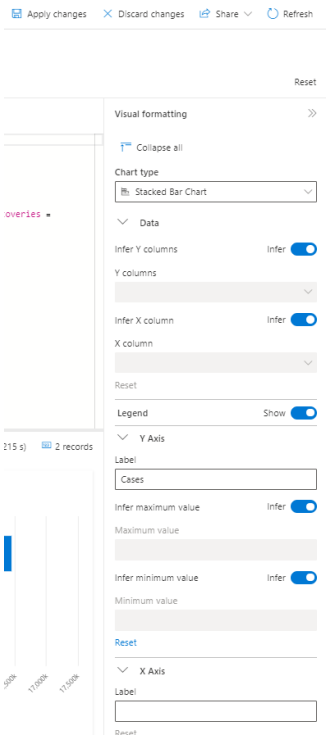
c) Top 20 Countries – Recoveries (date selected)

Pie chart – as per (a):



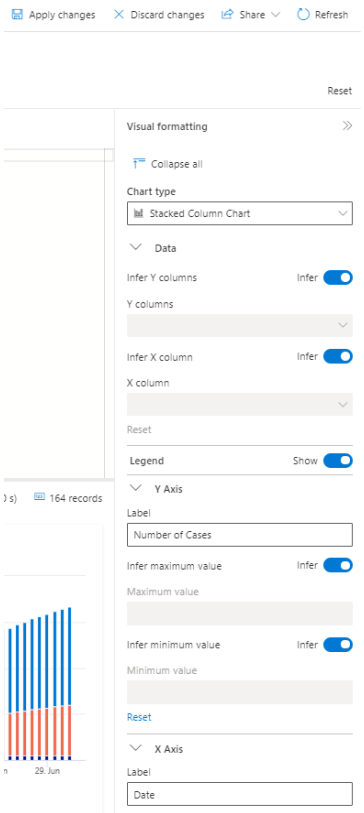
d) COVID Statistics (date & country selected)

Stacked Bar Chart:



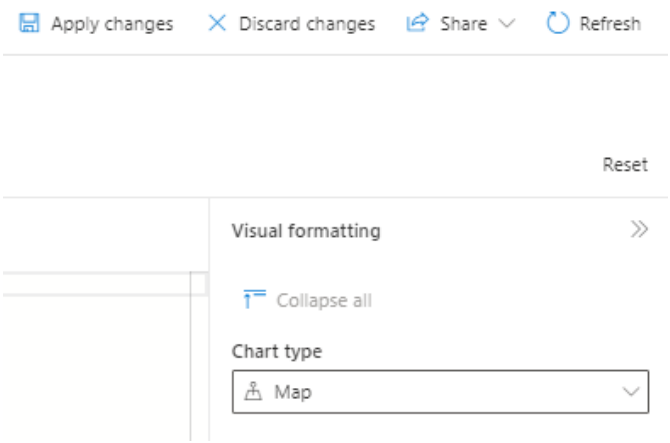
e) Total Worldwide Infections/Deaths/Recoveries

Stacked Column Chart:

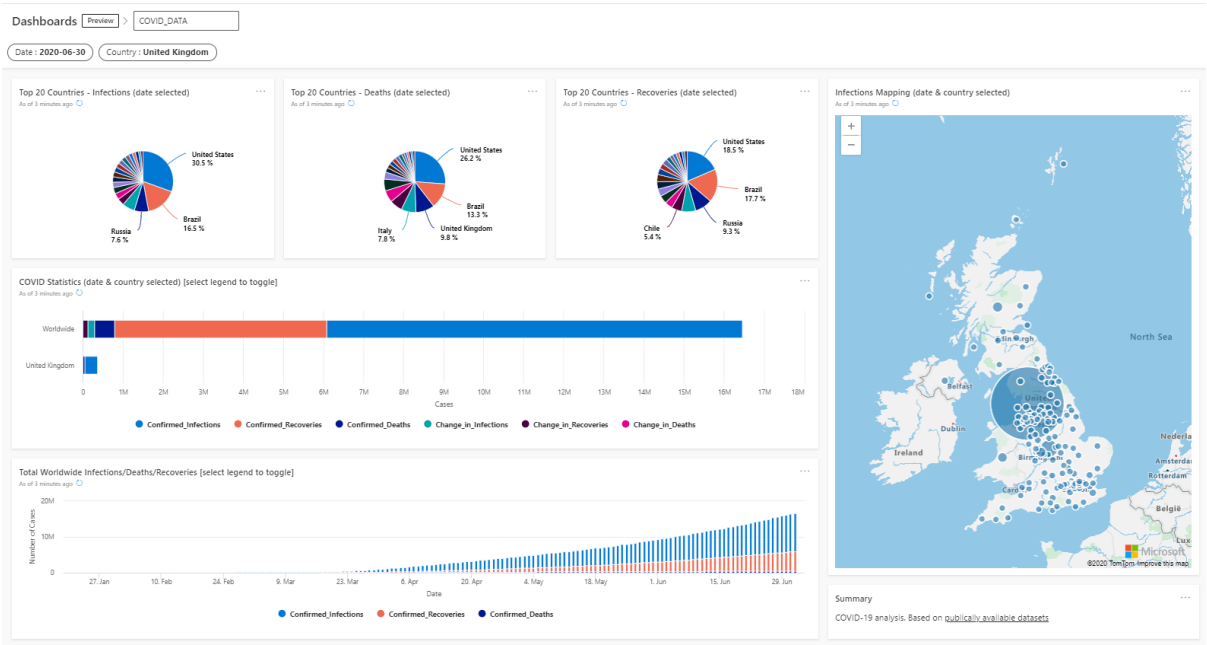


f) Infections Mapping (date & country selected)

Map:



The completed dashboard should look like the graphic below. If not, check the KQL behind each visualisation card definition and the visualisation properties (were appropriate):



Activity 8 – Wrap-up

You have completed Workshop 1. Now try to enhance the activities as follows:

- Try different file formats (JSON or JSON Lines) or larger volumes of files
- Experiment with KQL to perform richer queries and visualisations
- Expand your reporting & visualisations to include Power BI

Clean-up

Remember to stop both the ADX cluster and the ADF trigger to prevent Azure charges being incurred unnecessarily.