

AZURE DATA FACTORY SECURITY & AUTHENTICATION

This whitepaper covers different security options for ADF

Written By-

Blesson John (Data Solution Architect-Microsoft)

Issagha BA (Data Solution Architect-Microsoft)

Reviewed By-

Ye Xu (Senior Program Manager-ADF)

Gaurav Malhotra (Principal Program Manager-ADF)

Data Factory Security &
Authentication

Contents

- What is Azure Data Factory 2
- What is Service principal? 2
- Authentication to your data source in ADF using Service principal 2
 - Create a Service principal..... 2
 - Grant access to Service principal 2
- What is Managed Identity?..... 10
- Authentication to your data source in ADF using Managed Identity 10
 - Create a Managed Identity 11
- Create copy activity and linked service..... 17
- Using ACLs instead of RBAC 23
- Service principal vs Managed Identity 27

What is Azure Data Factory

More than ever before, security is one of the biggest concerns for companies. In the past, very few options existed when it came to passing credentials via code. Hardcoding credentials in configuration files or using plain text in code are some of the options. With the advent of cloud technology, we are witnessing a proliferation of generic users for application authentication. Azure addresses passing credential issue by using security features such Key vault, service principal and managed identity. This article is a step by step documentation on how to use service principal and managed identity when implementing data pipelines using Azure Data Factory.

What is Azure Data Factory

Azure Data Factory is a fully managed data integration service in the cloud. Data Factory allows you to easily create code-free and scalable ETL/ELT processes. More details available [here](#).

Azure Data Factory has more than 80 connectors. In this article, we'll discuss how to securely connect to the different data sources using Service principal and Managed Identity. We assume you are familiar with ADF.

What is Service principal?

Azure service principal is an identity that allows applications, automated processes and tools to access Azure resources. The role assigned to the service principal will define the level of access to the resources. It is possible to define the role at the subscription, resource group or resource level.

Authentication to your data source in ADF using Service principal

Create a Service principal

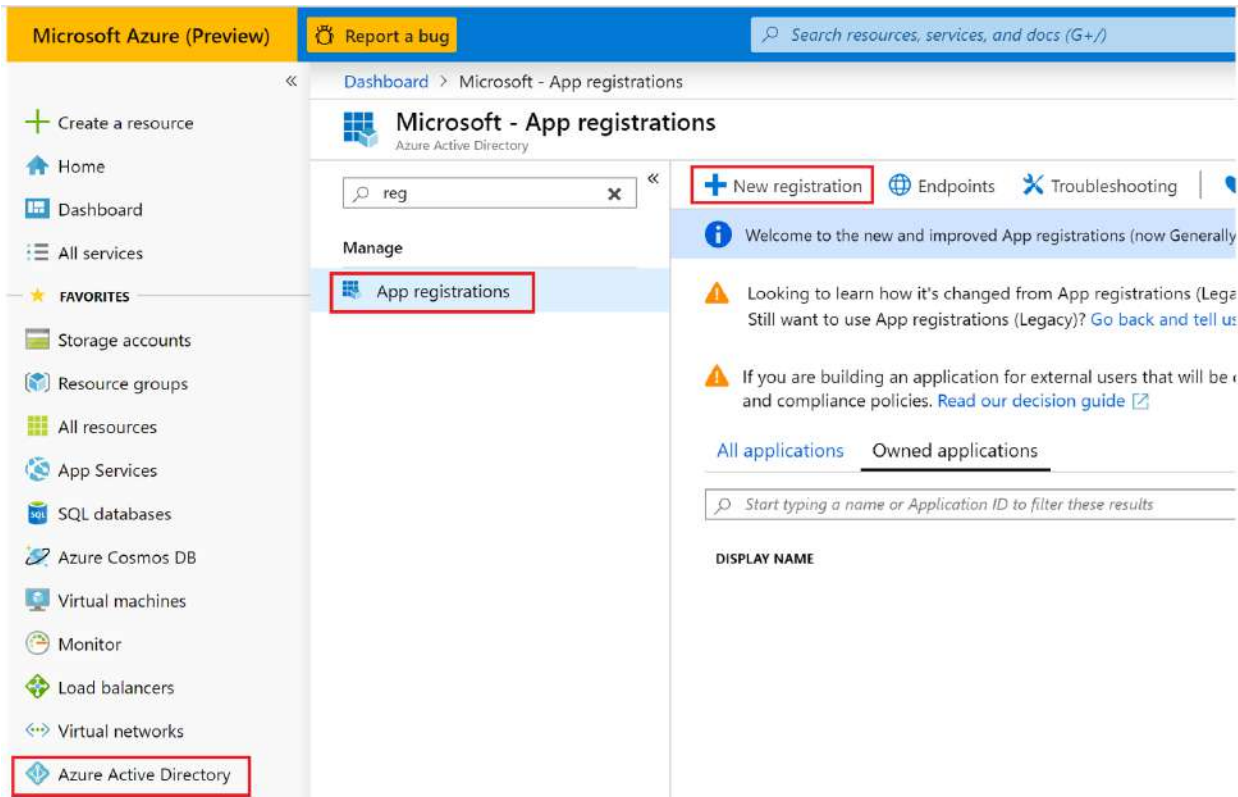
Note that it is possible to create a service principal using PowerShell and the Azure portal. In the article, we'll walk you through the creation of a Service using the Azure portal.

Grant access to Service principal

To create a service principal, you will first have to create an Azure Active Directory (AAD) Application and register the App.

Connect to the azure portal : www.portal.azure.com


Click on *Azure Active Directory* and select *new registration*



A new blade will appear after you select *new registration*.

Enter the name of your application

Register an application

 If you are building an application for external users that will be distributed by Microsoft, you must register as a first party application to meet all security, privacy, and compliance policies. [Read our decision guide](#)

* Name

The user-facing display name for this application (this can be changed later).

adfsecurity 

Supported account types

Who can use this application or access this API?

- Accounts in this organizational directory only (Microsoft only - Single tenant)
- Accounts in any organizational directory (Any Azure AD directory - Multitenant)
- Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)

[Help me choose...](#)

Redirect URI (optional)

We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.

Web  

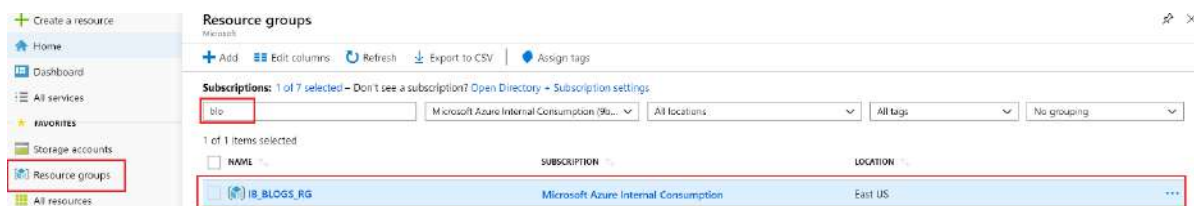
By proceeding, you agree to the [Microsoft Platform Policies](#)

[Register](#)

Select *register*.

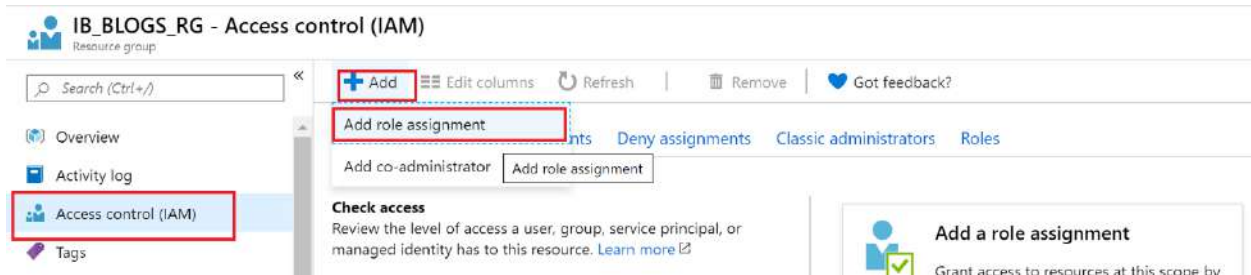
As mentioned above, the role assigned to the service principal will define the level of access to the resources. In this example, we'll assign the role to the service principal at the resource group level.

Find and select your resource group.



NAME	SUBSCRIPTION	LOCATION
IB_BLOGS_RG	Microsoft Azure Internal Consumption (96...)	East US

In the new blade, under *Access Control (IAM)* select *Add* to select *Add role assignment*



Select the role you want to assign to the service principal from the new screen.

In the assign access to dropdown list, select *Azure AD user, group, or service principal*.

In the select tab, find your application. You can enter the name of the App and, as it appears in the list, select it and click save

Role 
Contributor

Assign access to 
Azure AD user, group, or service principal

Select 
adfsecurity 

 Adfsecurity

 adfsecuritystoreprodus

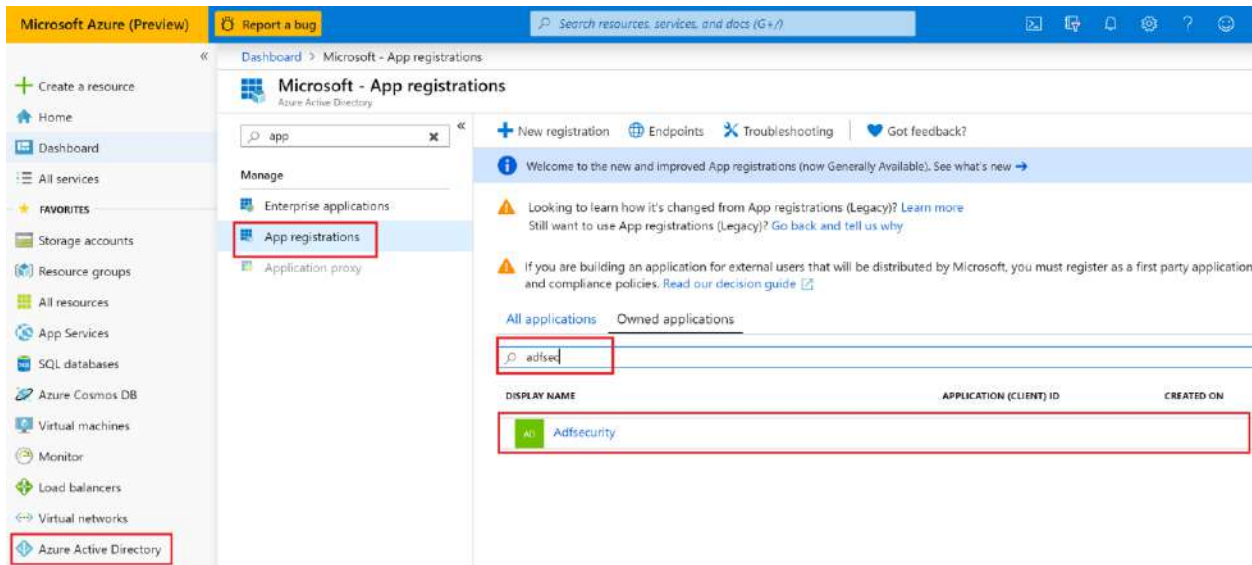
Selected members:

 Adfsecurity [Remove](#)

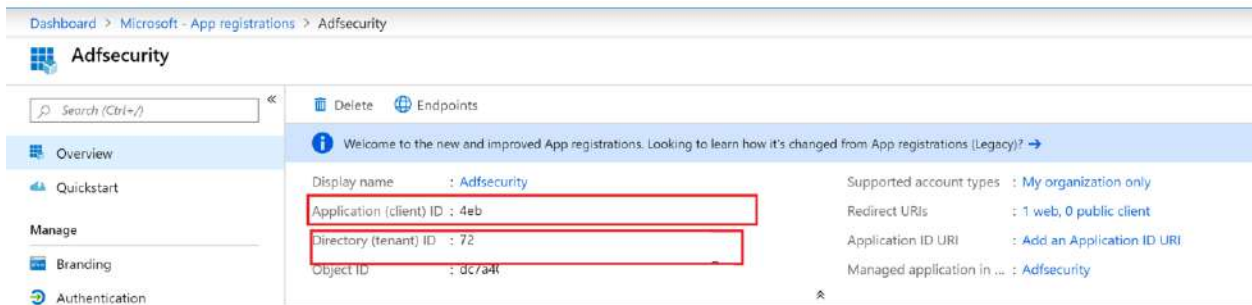
[Save](#) [Discard](#)

At this point, you almost are ready to start the configuration of your Data Factory. We just need to retrieve additional information to allow our Data Factory to authenticate. Not only we need the application id and the authentication key but we also need to generate a certificate and a secret.

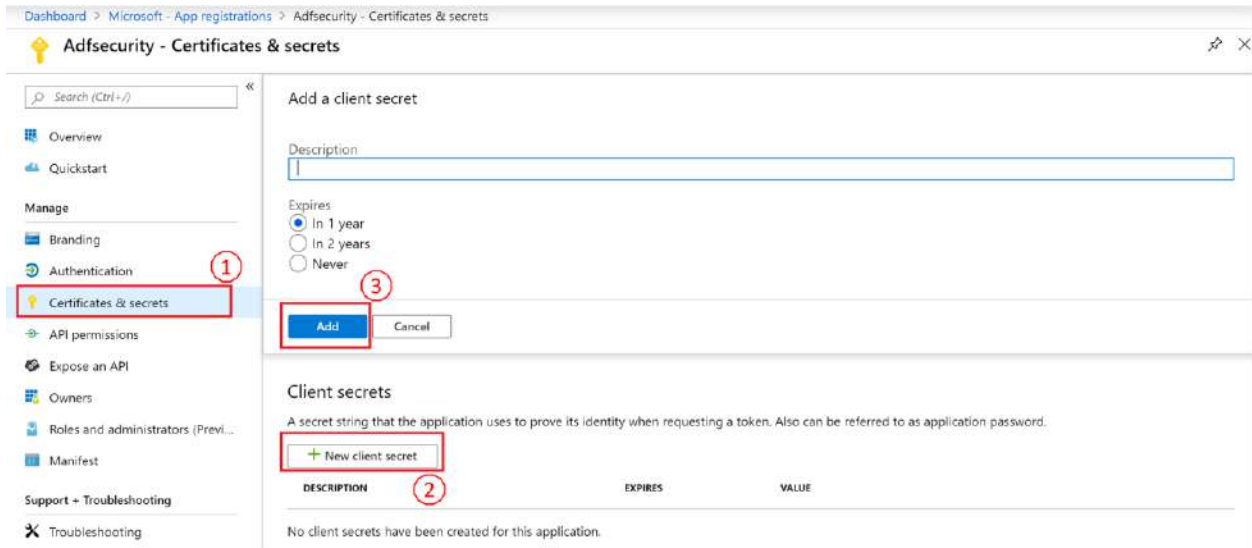
To get the application id and authentication key, click on *Azure Active Directory* in the main menu of the portal. Select App registrations and search and select your application



In the overview page of the new blade, copy the Directory (Tenant) Id and the Application (Client) Id



Let's generate the certificate that ADF will use to authenticate



Client secrets

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.



Copy and save this value as it will not be displayed going forward.

Configure your Linked Service

Once the Application created and registered, you can go back to your Data Factory and configure the linked service.

In this document, we'll show how to configure a linked service to an Azure Blob Storage, in a copy activity as an example.

In the author tab of ADF, select an existing pipeline or create a new one. In the *Activities* section, drag and drop *Copy data* under *Move & transform*.

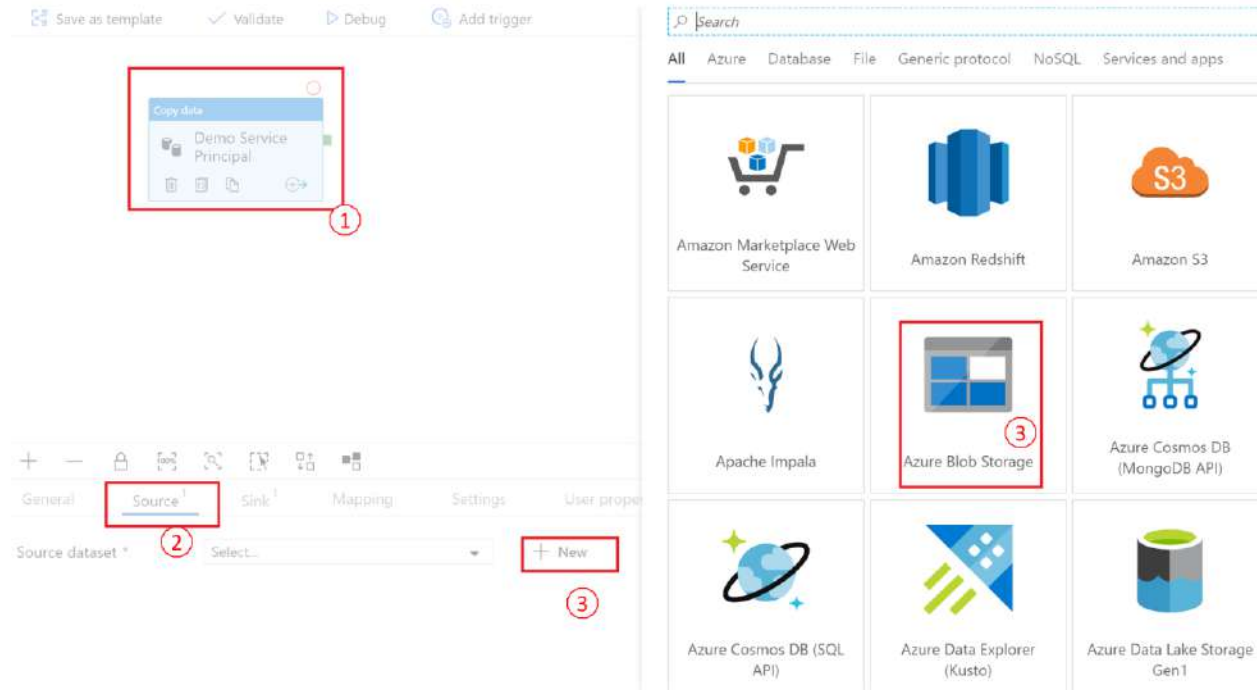


Let's create a new dataset. While creating the dataset, we'll be prompted to create a linked service.

To create a dataset, select the *copy data* activity. In the properties, select *Source* and click on *New* to create a new dataset. In the new blade, select *Azure Blob Storage* as a source.

In the new Window, select *CSV DelimitedText* as format and click on continue.

Under the properties window, enter the name of the dataset and under Linked Service select *New*.



Provide the requested information as indicated in the screenshot below

New linked service (Azure Blob Storage)

Name *

LS_AzureBlobStorage

Description

Connect via integration runtime *

AutoResolveIntegrationRuntime

Authentication method

Service Principal

Account selection method

From Azure subscription

Enter manually

Azure subscription

My Internal Subscription

Storage account name *

ibblogs

Tenant *

2f98

Service principal ID *

4ebd79

Service principal key

Azure Key Vault

Service principal key *

.....

Test connection

To linked service

To file path

✔ Connection successful

Create

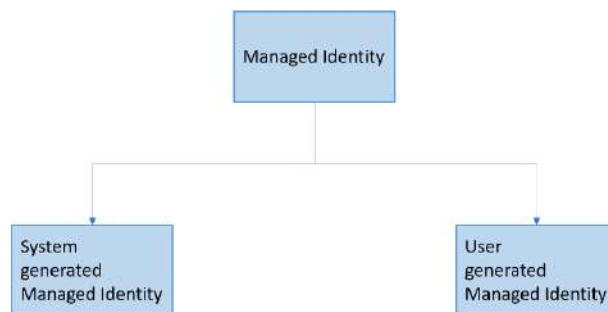
🔗 Test connection

Cancel

What is Managed Identity?

When you do not want to store credentials such as login details or keys within the code, you rely on managed identity. You can authenticate to different services in Azure without having to store credentials in services such as Azure keyVault. Managed Identity is the new name for MSI (Managed Service Identity). More details can be found [here](#).

There are two types of managed identity



Depending on the method used to create the ADF, the Managed Identity is created automatically whenever an ADF v2 is provisioned. When using SDK/REST API to create ADF, the identity session must be set to true to create MI automatically.

Authentication to your data source in ADF using Managed Identity

The scenario that we will explore is to copy data from one folder within ADLS gen2 storage to another folder within ADLS gen2 storage. We will be using managed identity to authenticate between ADLS and ADF. Let us start by creating an ADF using the portal. Once ADF has been created, you will be able to find the Managed Identity Application ID and Managed Identity Object ID by looking at the properties tab within ADF.

Create a Managed Identity

Home > ADFDev > ADFDev007 - Properties

ADFDev007 - Properties

Data factory (V2)

Search (Ctrl+/)

- Overview
- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems

Settings

- Locks

General

- Properties**

Getting Started

- Quick start

Monitoring

- Alerts
- Metrics
- Diagnostic settings

Support + troubleshooting

- Resource health
- New support request

Data factory: ADFDev007

Location: westeurope

Subscription: ba66f898-5c3d-473b-953e-fefd816b4264

Resource group: ADFDev

Provisioning state: Succeeded

Managed Identity Object ID: 34eee58a-e507-4c1c-a3c6-48b9b35ae0f5 Copy to clipboard

Managed Identity Tenant: 72f988bf-86f1-41af-91ab-2d7cd011db47

Managed Identity Application ID: 197dadb7-a573-4735-82c5-158f66004ace

The next step is to create an ADLS gen 2 with hierarchical namespace enabled.

Basics Networking **Advanced** Tags Review + create

Security

Secure transfer required ? Disabled Enabled

Data protection

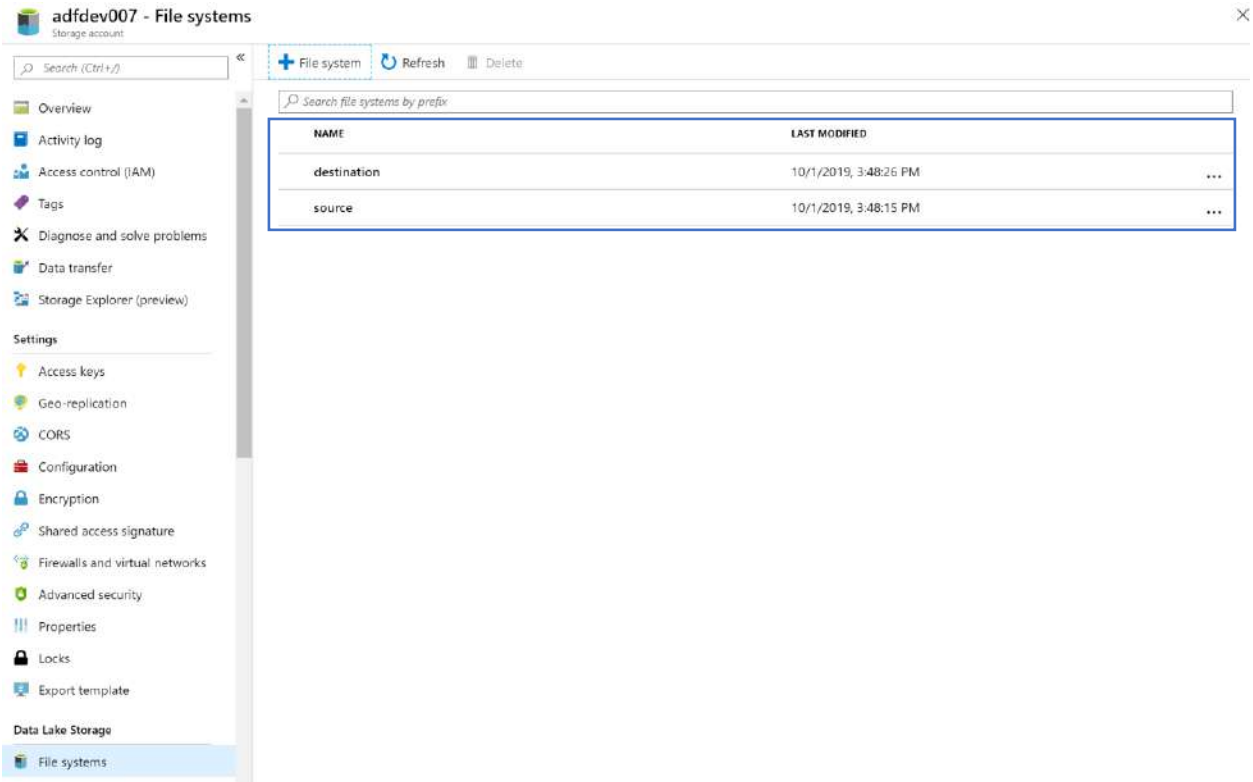
Blob soft delete ? Disabled Enabled

? Blob soft delete and hierarchical namespace cannot be enabled simultaneously.

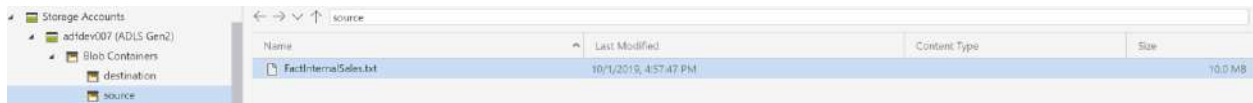
Data Lake Storage Gen2

Hierarchical namespace ? Disabled Enabled

Create two folders named **source** and **destination**. The folder structure for ADLS gen 2 looks like the one below



Upload a text file into the source folder using Azure storage explorer. Azure Data Explorer is a free tool to easily manage your storage accounts. You can download it [here](#) It can be any text file.



ADLS Gen2 supports both RBAC and POSIX-like access control lists (ACLs). The key thing to note is that RBAC is very coarse permission. The lowest level of permission that can be assigned is at a container level.

RBAC permission is evaluated first and if permissions are valid, ACLs are not checked, and access is granted. In short, RBAC supersedes ACLs. To provide ACL permission use **Managed Identity Object ID**. To provide RBAC permission use **Managed Identity Application ID**.

One can use this managed identity for Data Lake Storage Gen2 authentication. It allows this Azure Data factory to access and copy data to or from ADLS Gen2. Copy the Managed Identity Application ID from properties tab of Azure Data Factory.

Home > ADFDev > ADFDev007 - Properties

ADFDev007 - Properties

Data factory (V2)

Search (Ctrl+F)

- Overview
- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems

Settings

- Locks

General

- Properties**

Getting Started

- Quick start

Monitoring

- Alerts
- Metrics
- Diagnostic settings

Support + troubleshooting

- Resource health
- New support request

Data factory: ADFDev007

Location: westeurope

Subscription: ba66f898-5c3d-473b-953e-fefd816b4264

Resource group: ADFDev

Provisioning state: Succeeded

Managed Identity Object ID: 34ee58a-e507-4c1c-a3c6-48b9b35ae0f5 Copy to clipboard

Managed Identity Tenant: 72f988bf-86f1-41af-91ab-2d7cd011db47

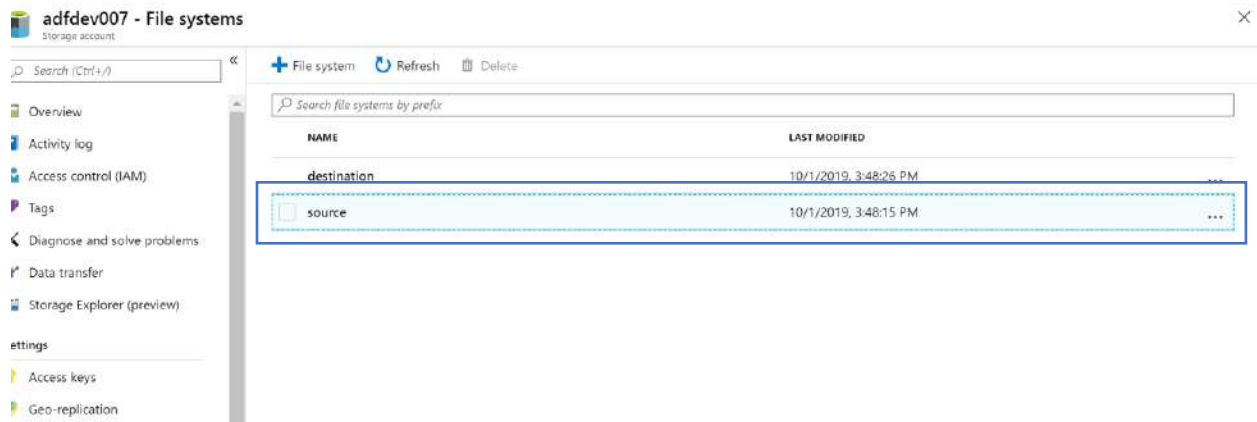
Managed Identity Application ID: 197dadb7-a573-4735-82c5-158f68004ace

Grant access to Managed Identity

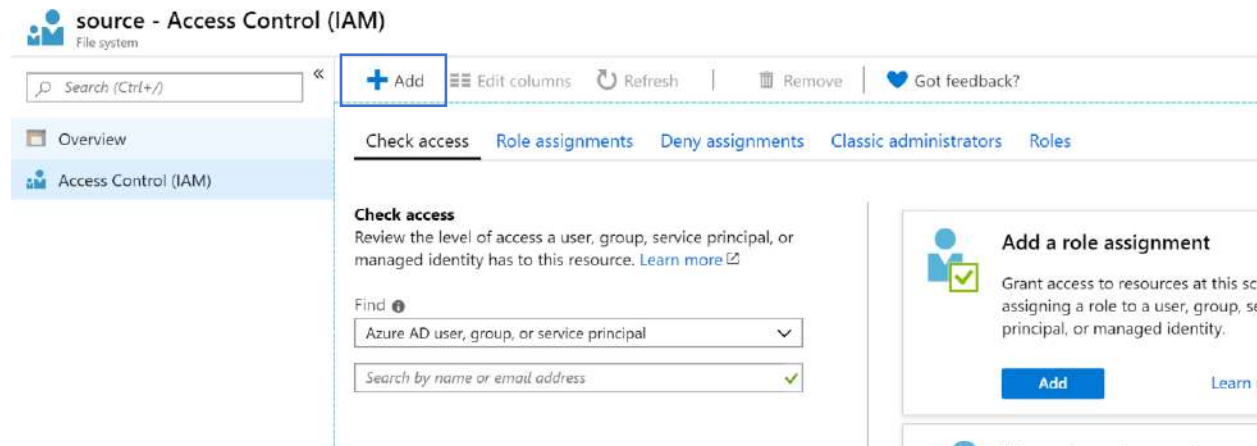
Now, we need to grant appropriate RBAC permission to the ADF Application ID on ADLS Gen2 folders- source and destination. Since we are copying file from source folder, the required permission is **Storage Blob Data Reader** role and for destination folder it is **Storage Blob Data Contributor** role. You will also need to grant "**Storage Blob Data Reader**" permission at account level* to be able to test connection and browse folder from ADF when setting up linked service.

*if you do not give **Storage Blob Data Reader RBAC** permission at the account level, you **will not** be able to test connection or browse to folder. You will need to trust that it will work if the right permissions are given at the folder level. For more details refer [this](#).

Click on the source folder



Select the RBAC for the source folder and click add



Add role assignment



Role ⓘ

Storage Blob Data Reader



Assign access to ⓘ

Azure AD user, group, or service principal



Select ⓘ

197dadb7-a573-4735-82c5-158f68004ace



ADFDev007

Once the ADF name is selected, you will be able to save the selection.

Selected members:

	ADFDev007	Remove
---	-----------	------------------------

[Save](#) [Discard](#)

After saving the changes, check whether an entry is present in the Access Control (IAM) tab for the folder **source**

STORAGE BLOB DATA READER

	ADFDev007	App	Storage Blob Data Reader ⓘ
---	-----------	-----	--

Do the same for the folder **destination** and the role is **Storage Blob Data Contributor**

Add role assignment ✕


Role ⓘ
Storage Blob Data Contributor

Assign access to ⓘ
Azure AD user, group, or service principal

Select ⓘ
197dadb7-a573-4735-82c5-158f68004ace ✓

No users, groups, or service principals found.

Selected members:

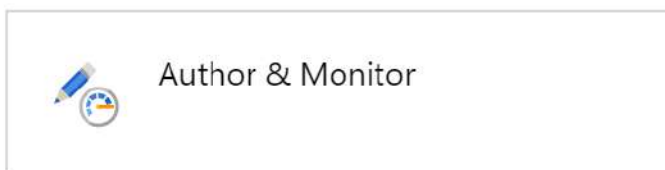
 ADFDev007	Remove
---	------------------------

Once saved, check the role assigned to the Managed Identity Application ID

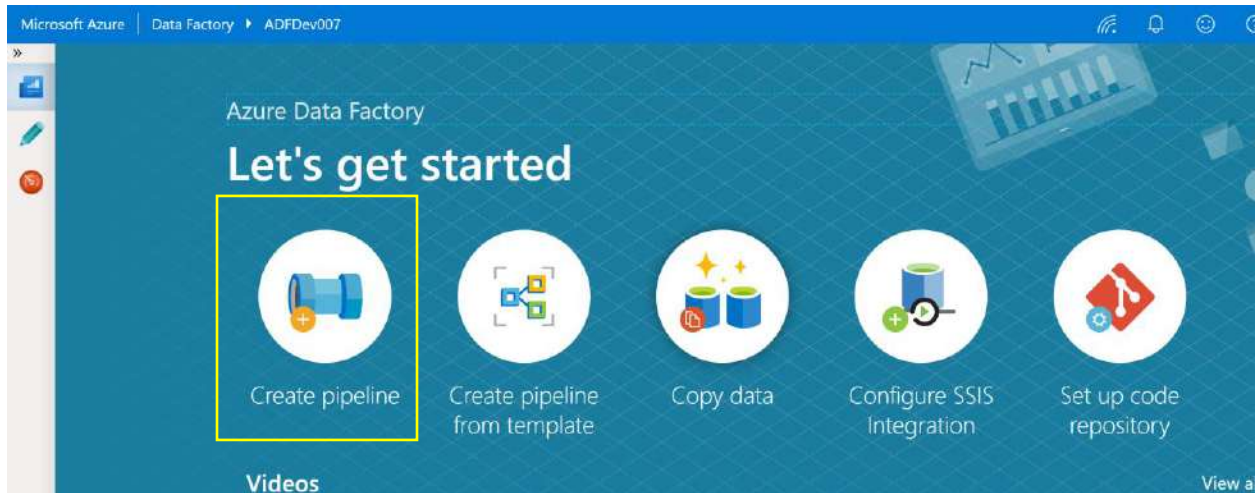


Create copy activity and linked service

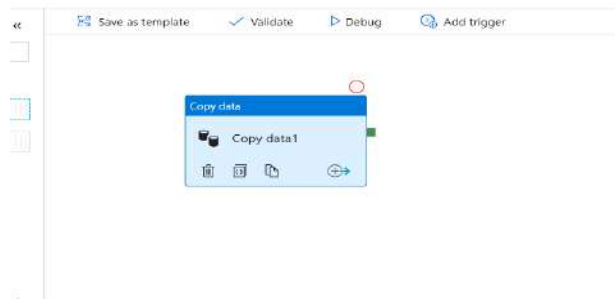
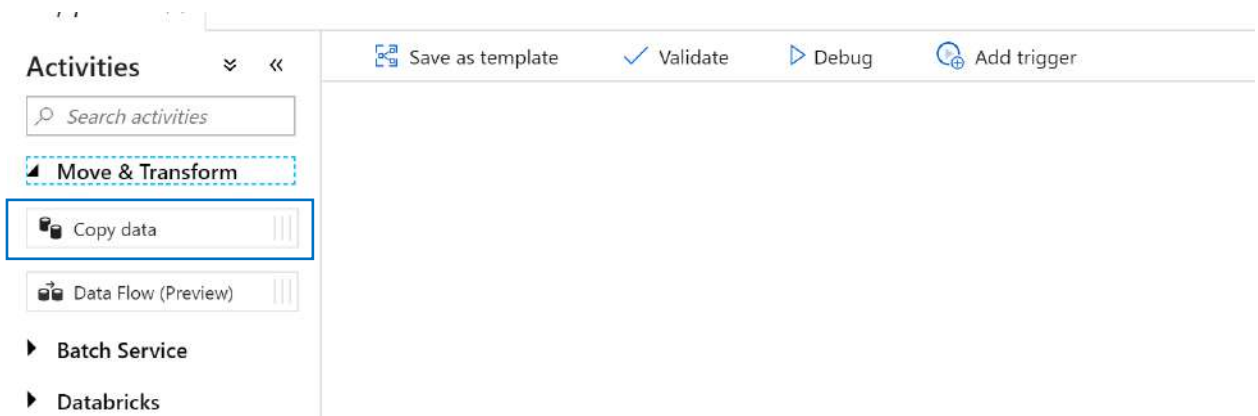
Go to the Azure Data Factory and click author & Monitor-



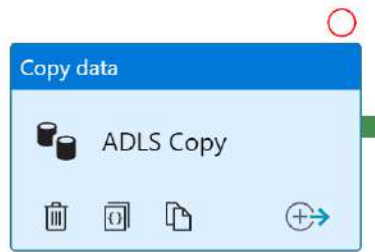
Click Create pipeline



Drag and drop Copy Data activity by expanding “Move & Transform”



Rename the copy data activity to “ADLS Copy”



+

—

🔒

100%

🔍

🖱️

↕️

🗑️

General

Source¹

Sink¹

Mapping

Settings

User properties

Name *

ADLS Copy

Documentation

Description

Timeout

7.00:00:00 ⓘ

Retry

0 ⓘ

Retry interval

30 ⓘ

Secure output

ⓘ

Now, click on the Source tab and select **NEW** set the source to be the ADLS Gen 2 folder that holds the text file

General

Source¹

Sink¹

Mapping

Settings

User properties

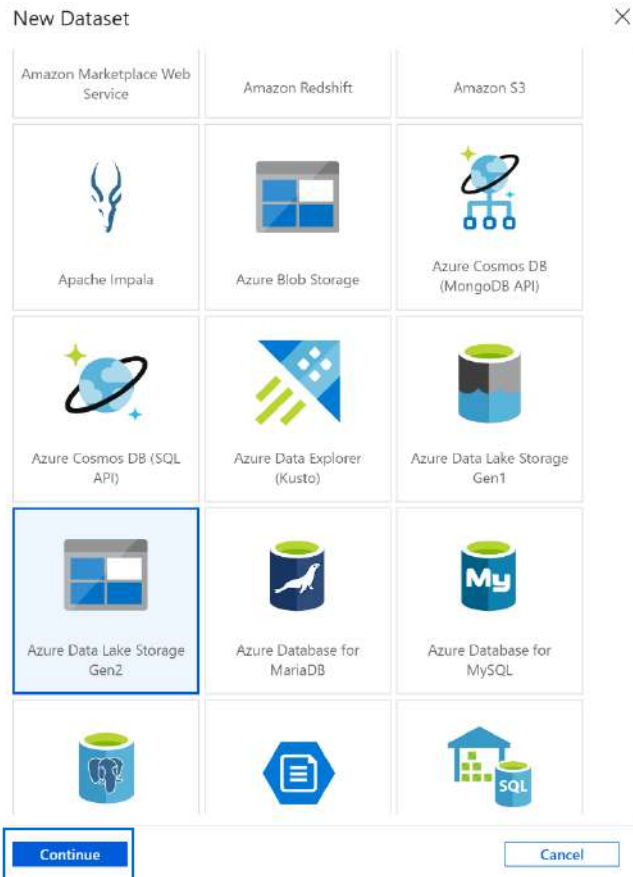
Source dataset *

Select...

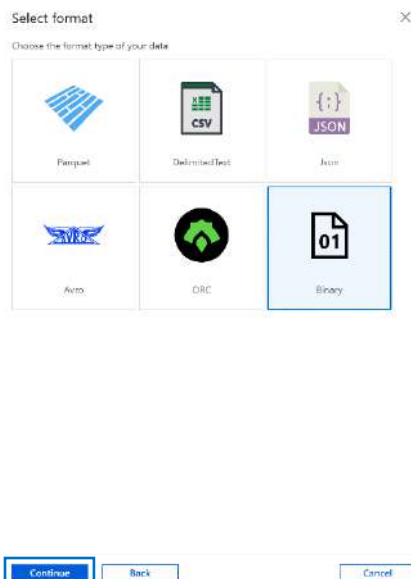
+ New

👁️ Preview data

Select the ADLS gen 2 storage and click continue



Select binary copy and click continue



Rename the activity to SourceFolder and select the **+new** in linked services

Set properties



Name

SourceFolder

Linked service *

Select...

Filter...

Select...
+ New

Change Authentication method to Managed Identity and set rest of the information as appropriate

New Linked Service (Azure Data Lake Storage Gen2) ✕

Name *
source001

Description

Connect via integration runtime *
AutoResolveIntegrationRuntime

Authentication method
Managed Identity

Account selection method
 From Azure subscription Enter manually

Azure subscription
Select all

Storage account name *
adfdev007

Managed identity application ID: 197dadb7-a573-4735-82c5-158f68004ace
Grant data factory managed identity access to your Azure Data Lake Storage Gen2. [Details](#)

Annotations
+ New

▶ Advanced

Create Test connection **Cancel**

To test the connection, press the test connection button and “Connection successful” will come up if permissions/access is correct.

✔ Connection successful

Create  **Test connection** **Cancel**

Browse to the file you want to copy and advance to the next page

Set properties ✕

Name

Linked service *

[Edit connection](#)

File path
 / /

OK **Next -> Advanced** **Back** **Cancel**

Repeat the same for the destination folder and run the ADF in debug mode to test whether the file copy works. If all the permissions were set correctly then the files get copied.



Save as template

✓ Validate

▶ Debug

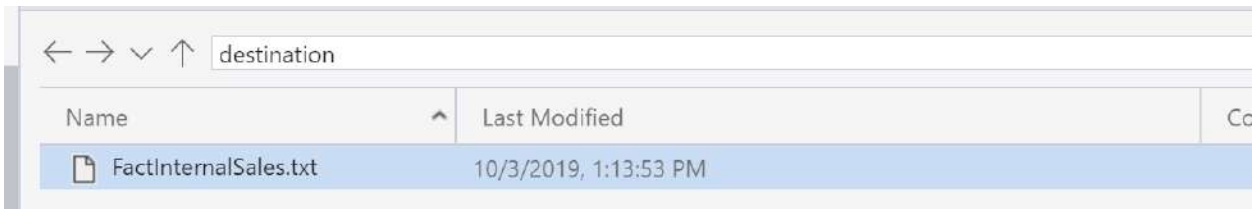


Add trigger

ADF execution in debug mode

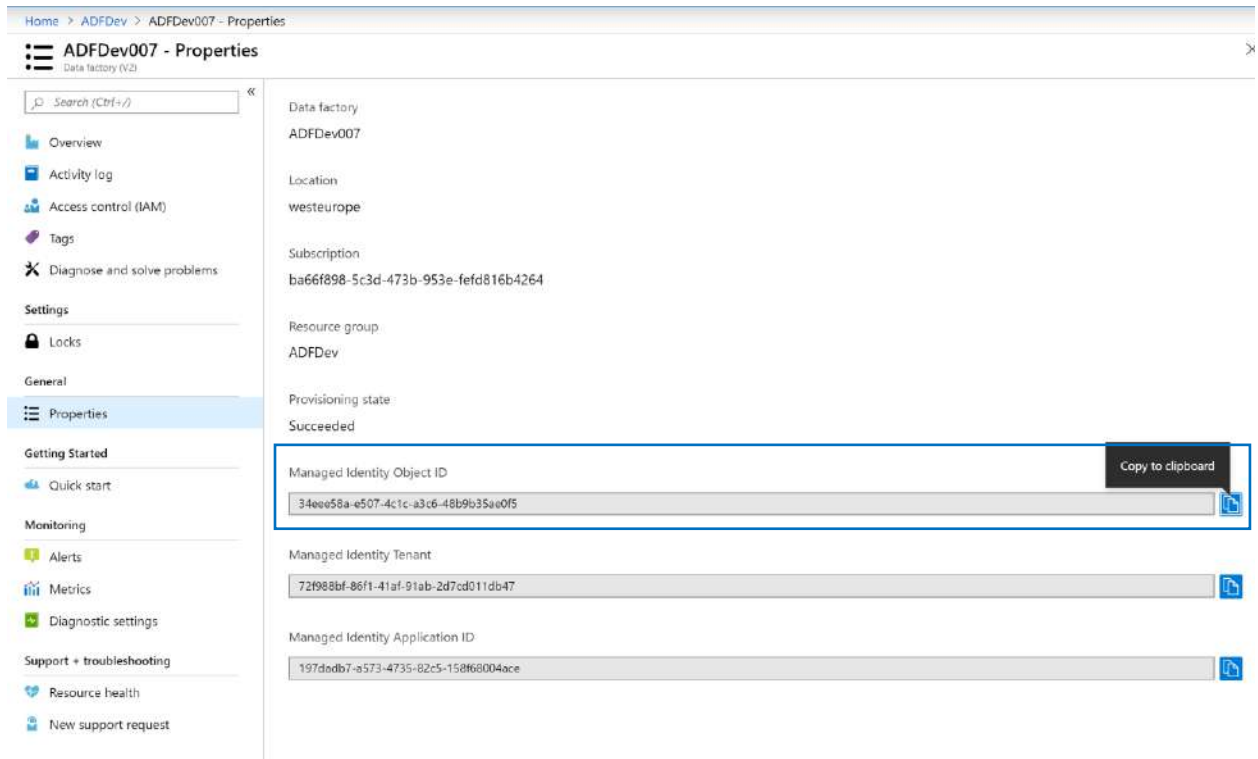
NAME	TYPE	RUN START	DURATION	STATUS	ACTIONS	RUN ID
ADLS Copy	Copy	10/03/2019 1:13 PM	00:00:05	✓ Succeeded		b24f2b23-ce66-4de1-9b3a-31c60188052c

File viewed from storage explorer

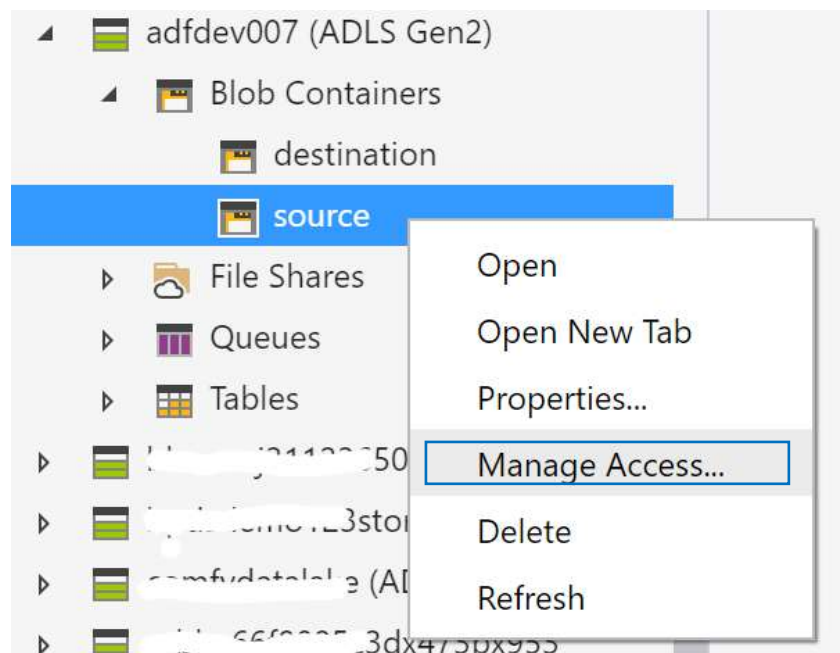


Using ACLs instead of RBAC



ACL or Access Control can be done at a very granular level. You grant permission to a specific file and thereby giving you more control over the permission that is granted to service principal/Managed Identity. If you would like to have granular control, use ACL. You do not have to give any RBAC permission. We are copying file from source folder; the required permission is **execute** on the source folder and **read** permission on the file to be copied. For destination folder **execute** and **write** permission need to be granted. The permission is granted to **Managed Identity Object ID** and not **Managed Identity Application ID**.



Open storage explorer and right click the folder for which the access needs to be managed



Add the **Managed Identity Object ID** and grant the execute permission

 **ADFDev007**
34eee58a-e507-4c1c-a3c6-48b9b35ae0f5 

Permissions for: ADFDev007

	Read	Write	Execute
<input checked="" type="checkbox"/> Access	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> Default *	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

* This will automatically add these permissions to all new children of this directory.
[Learn more about default ACLs.](#)

Add user, group, or service principal:

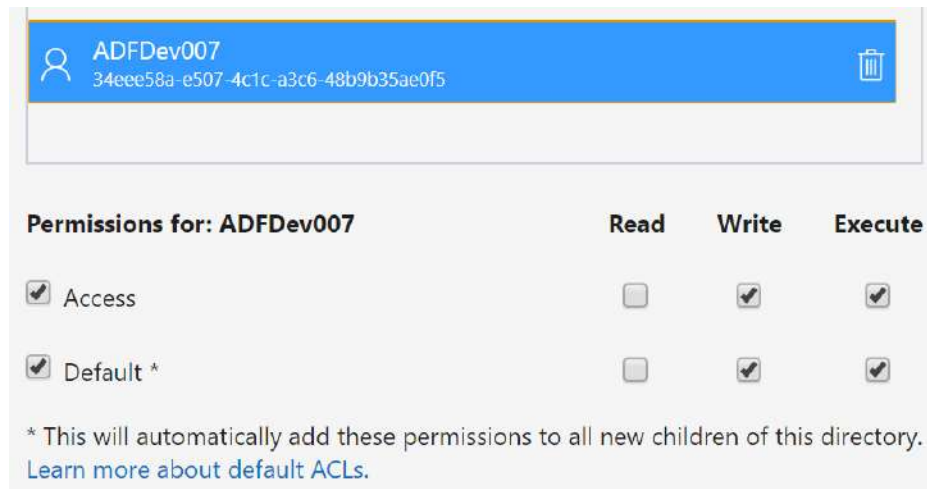
You need to grant read permission on the file to be copied

 **ADFDev007**
34eee58a-e507-4c1c-a3c6-48b9b35ae0f5 

Permissions for: \$superuser

	Read	Write	Execute
<input checked="" type="checkbox"/> Access	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Finally, on the destination folder grant **execute** and **write** permission.



You will also need to grant “**Storage Blob Data Reader**” permission at account level* to be able to test connection and browse folder from ADF when setting up linked service.

*if you do not give **Storage Blob Data Reader RBAC** permission at the account level, you **will not** be able to test connection or browse to folder. You will need to trust that it will work if the right permissions are given at the folder level. For more details refer [this](#).

Once all the above activities are complete, you can follow the steps provided in “Create copy activity and linked service” session.

Service principal vs Managed Identity

We worked our way through this document to see how we can use each service principal and managed identity. Both are secure and serve the customers well. Managed Identity is fully managed. This gives Managed Identity an edge in organizations where fully managed service is a priority. Hence, our recommendation is to use Managed Identity whenever it is supported by the service you are using. For the services not supporting managed identity, use service principal.

For a list of Azure services that support the managed identities for Azure resources feature, see [Services that support managed identities for Azure resources](#).