



Manual Básico Azure para DE

Por Santos Nicanor Oliva Escobar



Índice

Índice.....	2
Introducción.....	4
Resource Group.....	5
Creación de un Grupo de Recursos	5
Service Principal [SP].....	8
Creación de Service Principal.....	8
Gestión de Secrets.....	11
Key Vault [KV]	12
Creación de Key Vault	12
Creación de Secretos	16
Storage Account	19
Creación de Storage Account.....	20
Configuración del Datalake	23
Gestión de Blobs.....	23
Gestión de permisos	25
Databricks.....	29
Creación de un área de trabajo de Azure Databricks.....	30
Creación de Clusters	34
Creación de Scope	36
Creación de un Cuaderno.....	38
Montar Datalake en Databricks.....	39
Plantilla.....	43
Tratamiento de Dataframes.....	44
Navegar File System	44
Equivalencia entre Sql – Spark	47
Vistas Temporales.....	49
Azure Data Factory [ADF]	50
Creación de Azure Data Factory	51
Integration Runtime	55
Tipos de Integration Runtime	55
Creación de Integration Runtime	56



Linked Services [LS]	59
Gestión de Datasets	68
Creación de un Pipeline	78
Gestión de Actividades	79
Gestión de Ejecución - Triggers.....	82
Anexo 3: Ejecución de una Notebook Databricks desde Data Factory	87
Linked Service de Databricks	87
Uso desde Data Factory	93
Anexo 1 : Permisos de Acceso a DataLake.....	94
Integración de Azure Active Directory (Azure AD) para blobs y colas.	95
Acceso a datos con una cuenta de Azure AD.....	96
Acceso a datos desde Azure Portal	96
Acceso a datos desde PowerShell o la CLI de Azure	97
Roles RBAC integrados para blobs y colas	98
Glosario	98
Autorizar con clave compartida [SAS].....	100
Especificando el encabezado de Autorización.....	100
Construyendo el string de firma	101
Administración del acceso de lectura anónimo a contenedores y blobs	103
Anexo 2: Implementación en Databricks	104



Introducción

A continuación, se explicará el How To de la prueba de concepto para Pami utilizando la suite de Azure en la nube.

Utilizaremos los siguientes recursos:

- Resource Group
- Data Lake
- Service principal
- Key Vault
- Data Factory



Resource Group

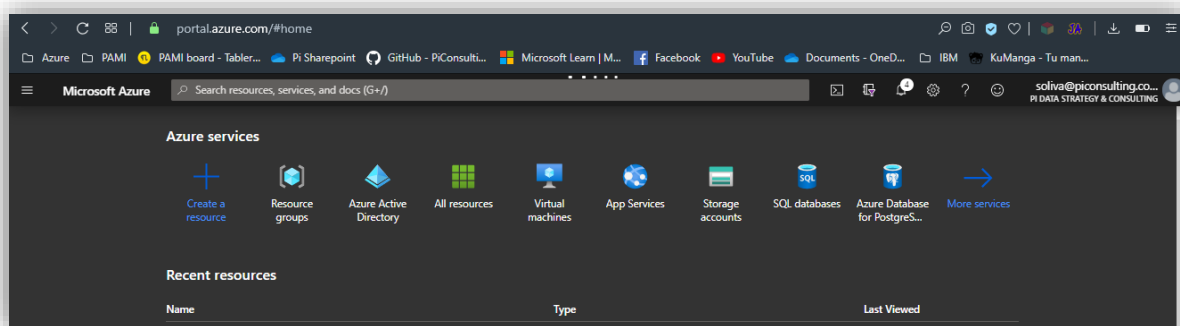
Los grupos de recursos permiten administrar todos sus recursos en una aplicación juntos.

Por lo general, un grupo contendrá recursos relacionados con una aplicación específica. Por ejemplo, un grupo puede contener un recurso del sitio web que aloja su sitio web público, una base de datos SQL que almacena datos relacionales utilizados por el sitio y una cuenta de almacenamiento que almacena activos no relacionales.

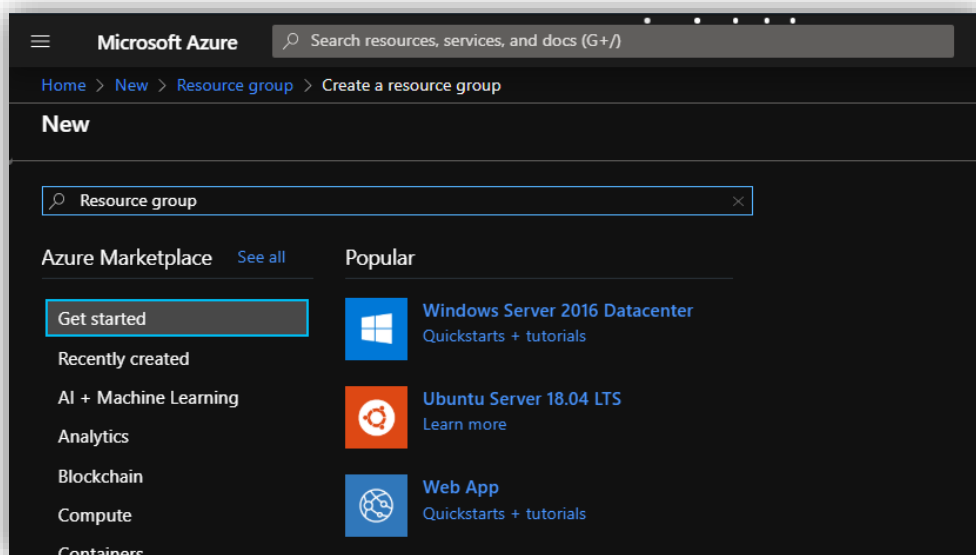
En nuestro caso, utilizaremos los recursos antes mencionados.

Creación de un Grupo de Recursos

Desde la página www.portal.azure.com hacemos click en crear un nuevo recurso



Buscamos la opción Resource Group y hacemos click en Create





Ya dentro de la configuración del recurso:

The screenshot shows the 'Create a resource group' page in the Microsoft Azure portal. The breadcrumb navigation at the top indicates the path: Home > New > Resource group > Create a resource group. The page title is 'Create a resource group'. Below the title, there are tabs for 'Basics', 'Tags', and 'Review + create'. The 'Basics' tab is active. A descriptive text explains that a resource group is a container for related resources. Under 'Project details', the 'Subscription' dropdown is set to 'Visual Studio Enterprise - MPN (e8fd21cd-aa84-4d86-ae5a-14bad7b02...)' and the 'Resource group' text box contains 'Pami-Test'. Under 'Resource details', the 'Region' dropdown is set to '(US) East US'. At the bottom, there are three buttons: 'Review + create' (highlighted in blue), '< Previous', and 'Next: Tags >'.

Seleccionamos la Suscripción de Azure en la que se va a implementar el área de trabajo.

Elegimos el nombre a utilizar, seleccionamos la región donde se armara el recurso, nosotros utilizamos East US por razones de precio.

Es importante que la región utilizada en el proyecto sea siempre la misma para mejor performance.

El siguiente paso es definir los Tags, estos no son obligatorios, pero permite un mayor seguimiento a cada recurso al buscar por etiqueta.

Estas siguen una relación de Name: Value, en cuyos campos se realizan sucesivamente filtros



The screenshot shows the 'Create a resource group' page in the Microsoft Azure portal, specifically the 'Tags' tab. The breadcrumb navigation is 'Home > New > Resource group > Create a resource group'. The page title is 'Create a resource group'. Below the title, there are tabs for 'Basics', 'Tags', and 'Review + create'. The 'Tags' tab is active. A message states: 'Apply tags to your Azure resources to logically organize them by categories. A tag consists of a key (name) and a value. Tag names are case-insensitive and tag values are case-sensitive. [Learn more](#)'. Below this, there is a table with three columns: 'Name', 'Value', and 'Resource'. The first row has 'application' in the 'Name' column, 'Test' in the 'Value' column, and 'Resource group' in the 'Resource' column. The second row has empty input fields for 'Name' and 'Value', and 'Resource group' in the 'Resource' column. At the bottom, there are three buttons: 'Review + create', '< Previous', and 'Next : Review + create >'.

Luego, se Valida y se crea

The screenshot shows the 'Create a resource group' page in the Microsoft Azure portal, specifically the 'Review + create' tab. The breadcrumb navigation is 'Home > New > Resource group > Create a resource group'. The page title is 'Create a resource group'. Below the title, there are tabs for 'Basics', 'Tags', and 'Review + create'. The 'Review + create' tab is active. A green banner at the top of the tab area says '✓ Validation passed.'. Below this, there is a 'Basics' section with the following details: 'Subscription: Visual Studio Enterprise – MPN', 'Resource group: Pami-Test_', and 'Region: (US) East US'. At the bottom, there are three buttons: 'Create', '< Previous', and 'Next >'.



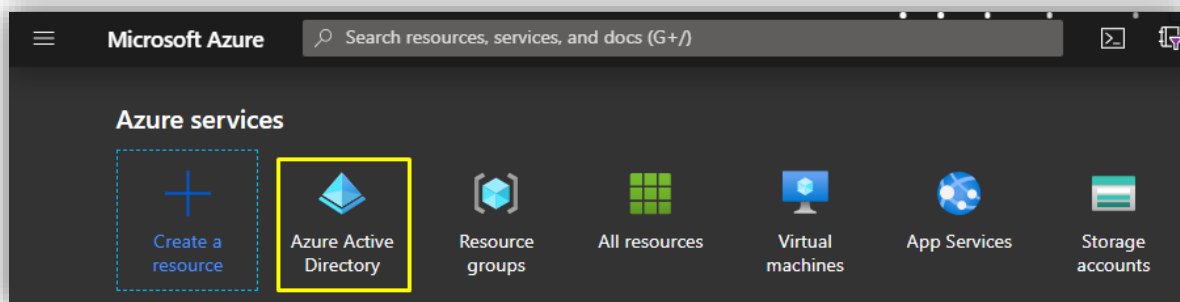
Service Principal [SP]

Un Service Principal [SP] es una identidad creada para su uso con aplicaciones, servicios hospedados y herramientas automatizadas que acceden a los recursos de Azure.

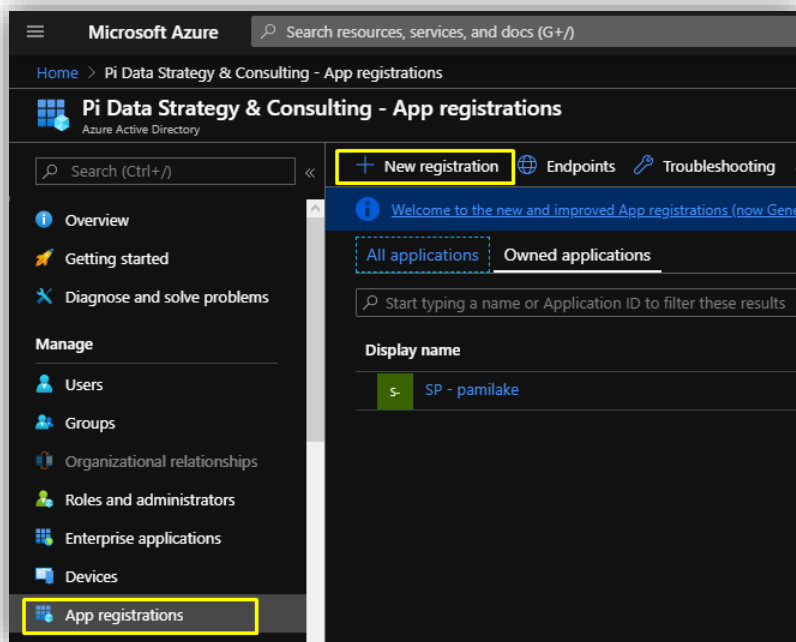
Este acceso está restringido por los roles asignados a la entidad de servicio, lo que permite controlar a qué recursos pueden tener acceso y en qué nivel. Por motivos de seguridad, se recomienda usar siempre entidades de servicio con las herramientas automatizadas, en lugar de permitirles iniciar sesión con una identidad de usuario.

Creación de Service Principal

Para crear uno hay que crear un Azure Active Directory



Hacer click en App registrations – New registration





Ingresamos el nombre del SP deseado y luego registrar

Register an application

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

Supported account types
Who can use this application or access this API?

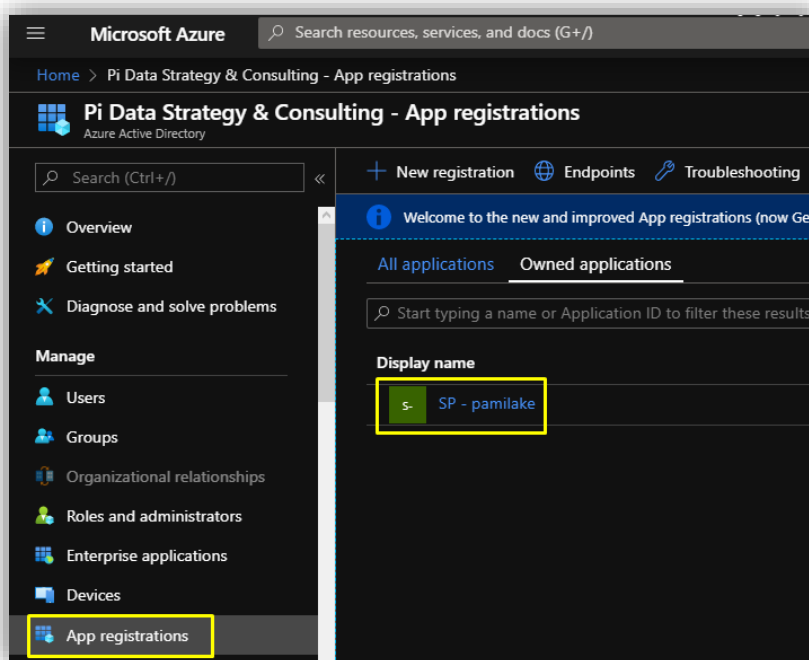
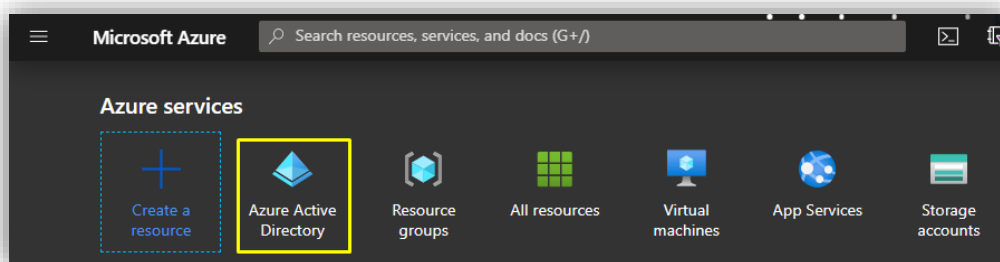
- ☒ Accounts in this organizational directory only (Pi Data Strategy & Consulting only - Single tenant)
- ☐ Accounts in any organizational directory (Any Azure AD directory - Multitenant)
- ☐ Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)

[Help me choose...](#)

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

Register

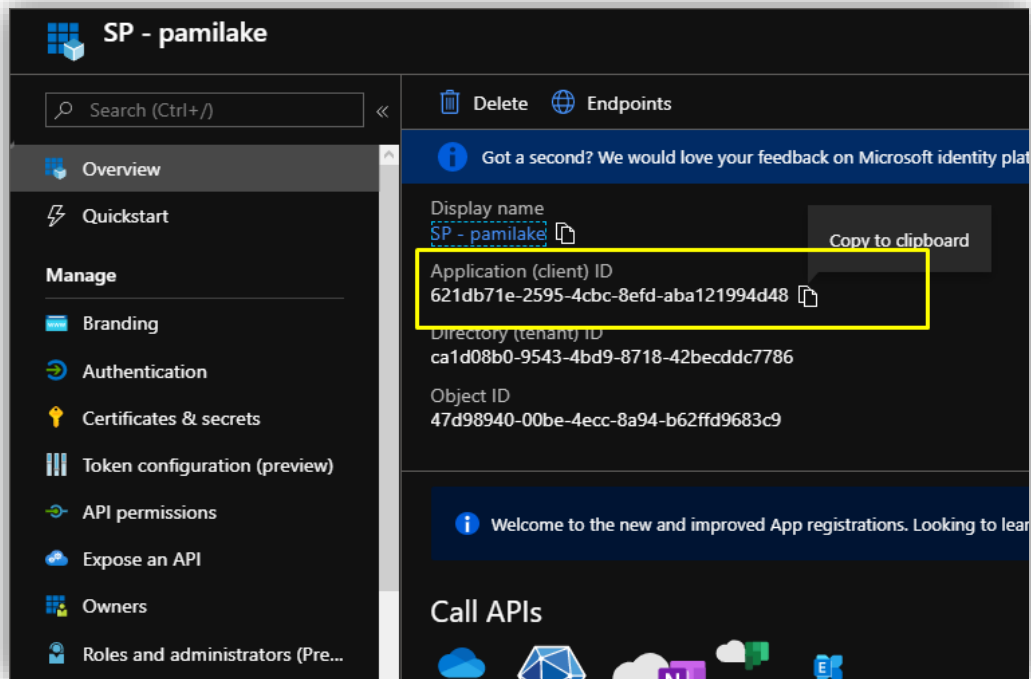
Para ingresar a la configuración del SP, desde Azure Active Directory





Para hacer uso de este SP necesitamos su ID y su Secret o Password, se acceden a los mismos de la siguiente manera:

Dentro de la configuración del SP, en el Overview, he recuadrado el ID

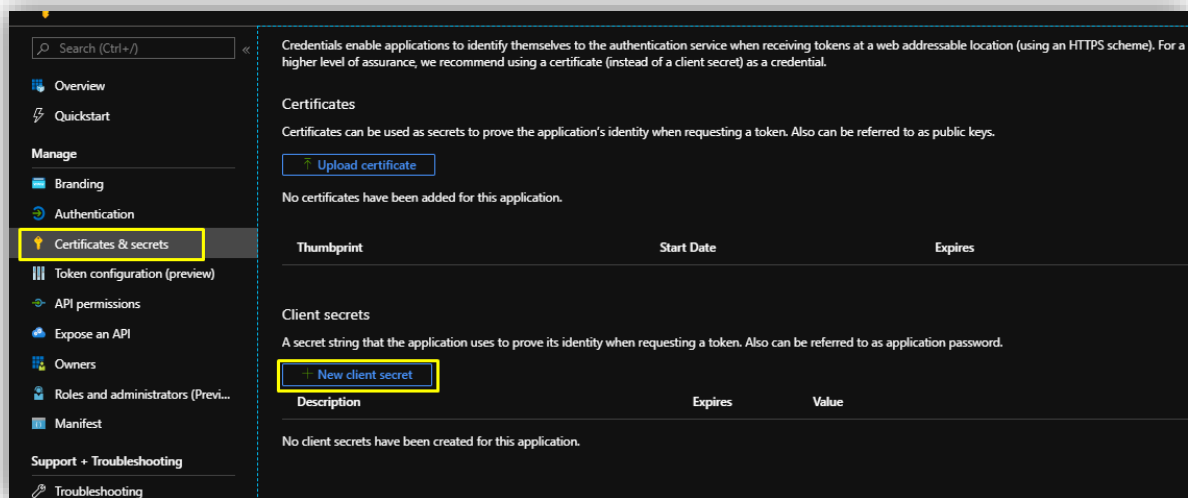


Gestión de Secrets

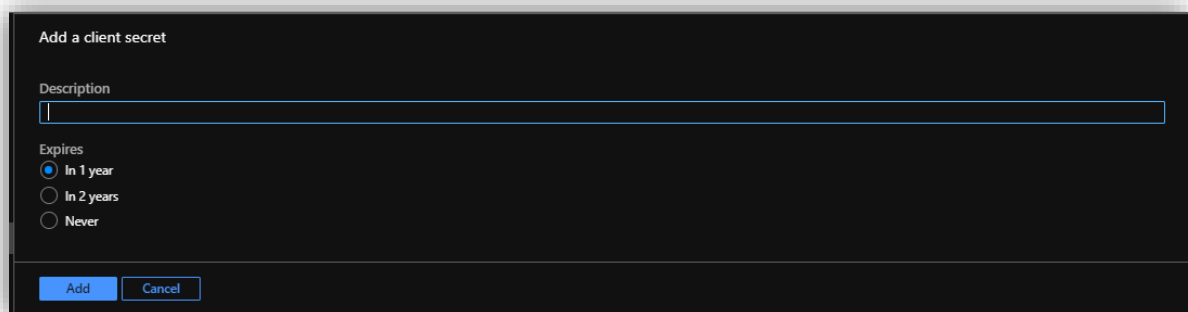
Para acceder al Sp, se necesitará un Secret del mismo.

Este se encuentra en Certificates & Secrets, estos se pueden programar para que expiren luego de cierto tiempo o que no lo hagan.

En nuestro caso, pegaremos en Values, el Secret del SP a trabajar, este lo tomamos accediendo al mismo haciendo click en Certificates & secrets – New client secret



Aquí podremos elegir el tiempo de expiración de este secreto, nosotros elegiremos Never.



Es importante guardar el Value del secret ya que el mismo se mostrará solo una vez. En caso de perderlo, se tendrá que crear un nuevo client secret

Para guardar esta información sensible utilizaremos el Key Vault.

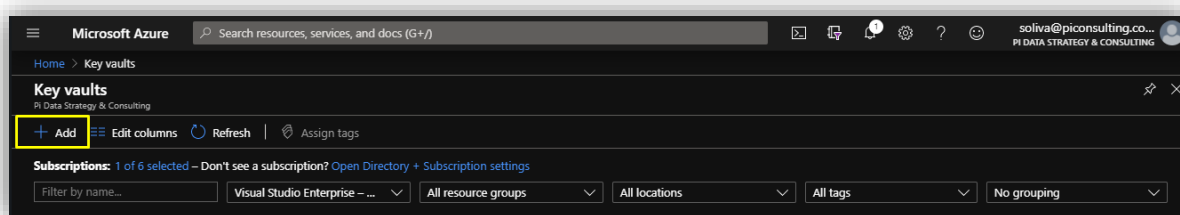
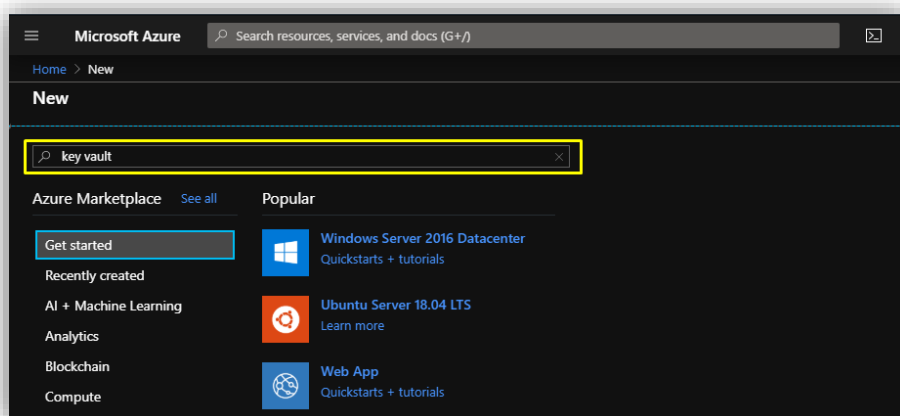
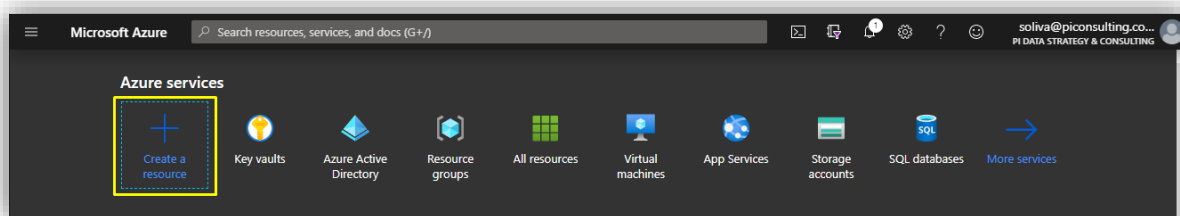


Key Vault [KV]

Con Azure Key Vault, puede proteger las claves de cifrado y los secretos de las aplicaciones, como las contraseñas, utilizando claves almacenadas en módulos de seguridad de hardware (HSM).

Creación de Key Vault

Para crear un Key Vault se deben seguir los siguientes pasos





Creando el Key Vault, seleccionaremos la suscripción y el grupo de recursos del proyecto, pondremos en Soft-delete [Enable] y 90 días para evitar el borrado accidental del key vault.

Microsoft Azure Search resources, services, and docs (G+/)

Home > Key vaults > Create key vault

Create key vault

Subscription * Visual Studio Enterprise – MPN (e8fd21cd-aa84-4d86-ae5a-14bad7b024cb) ▼

Resource group * Pami-Test ▼
[Create new](#)

Instance details

Key vault name * ① Test-Vault-Pami ✓

Region * East US ▼

Pricing tier * ① Standard ▼

Soft-delete ① Enable Disable

Retention period (days) * ① 90

Purge protection ① Enable Disable

Review + create < Previous Next : Access policy >

Podremos gestionar las políticas de acceso al momento de crear el key vault o también luego de finalizada la creación de este.

Microsoft Azure Search resources, services, and docs (G+/)

Home > Key vaults > Create key vault

Create key vault

Enable Access to:

- ☒ Azure Virtual Machines for deployment ①
- ☒ Azure Resource Manager for template deployment ①
- ☒ Azure Disk Encryption for volume encryption ①

+ Add Access Policy

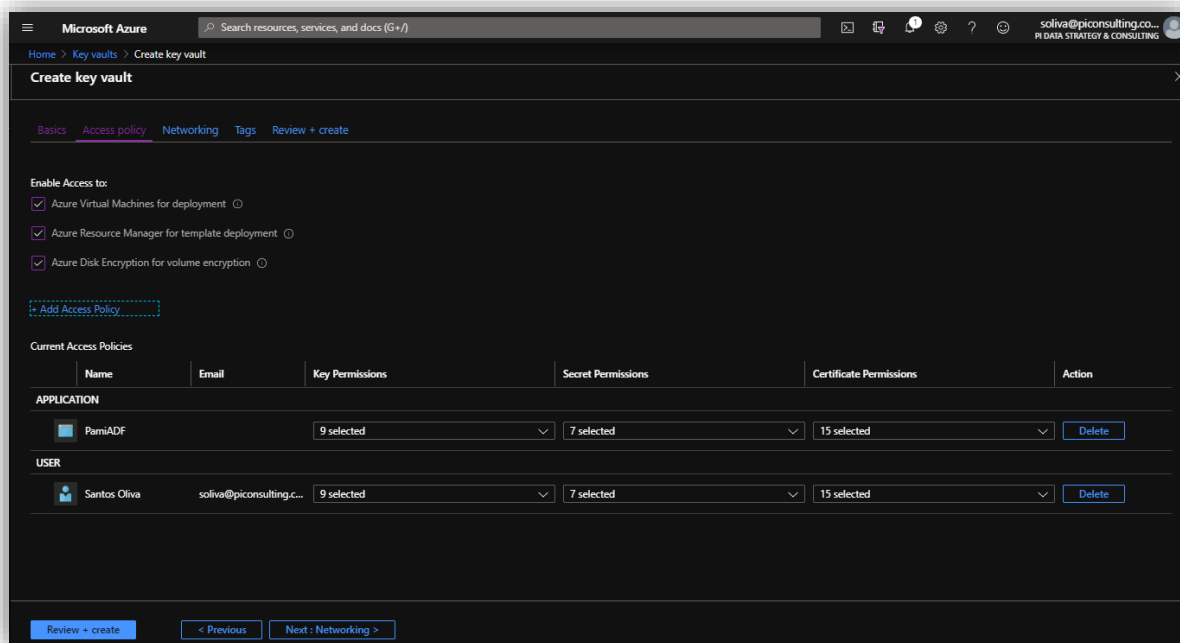
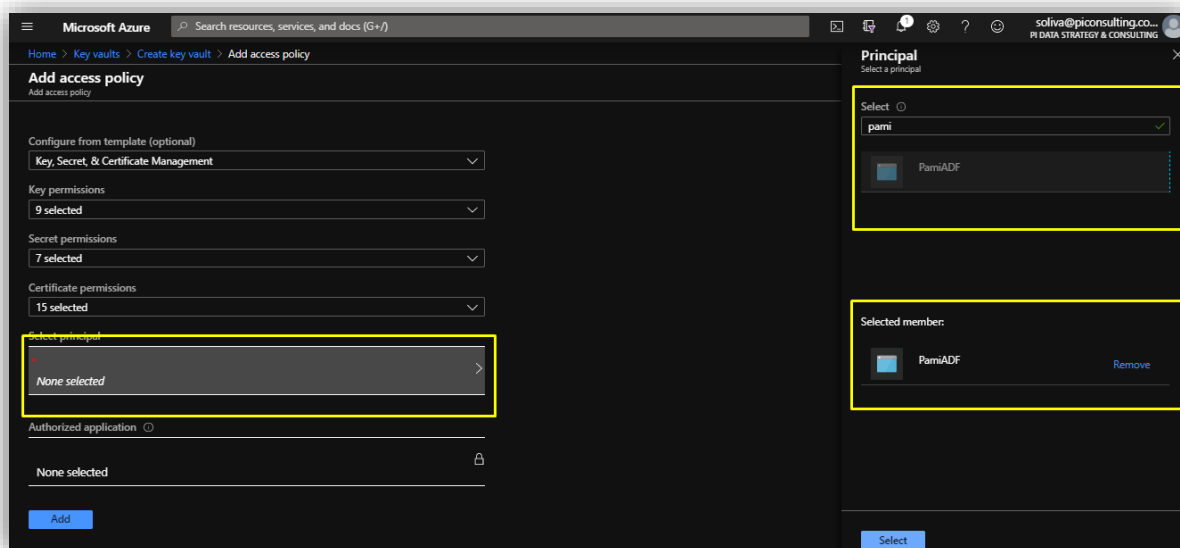
Current Access Policies

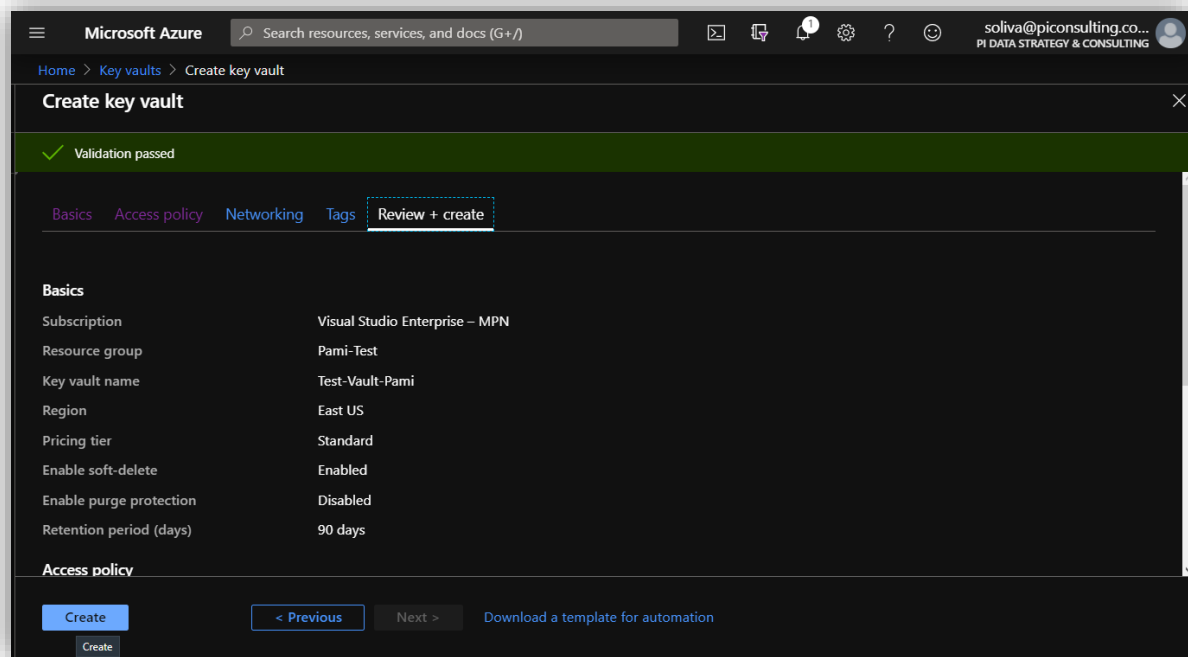
Name	Email	Key Permissions	Secret Permissions	Certificate Permissions	Action
USER					
Santos Oliva	soliva@piconsulti...	9 selected ▼	7 selected ▼	15 selected ▼	Delete

Review + create < Previous Next : Networking >



Primero seleccionamos el template de permisos dados al usuario o SP, luego hacemos click en Select Principal buscamos el o los deseados y finalizando hacemos click en Select y en Add



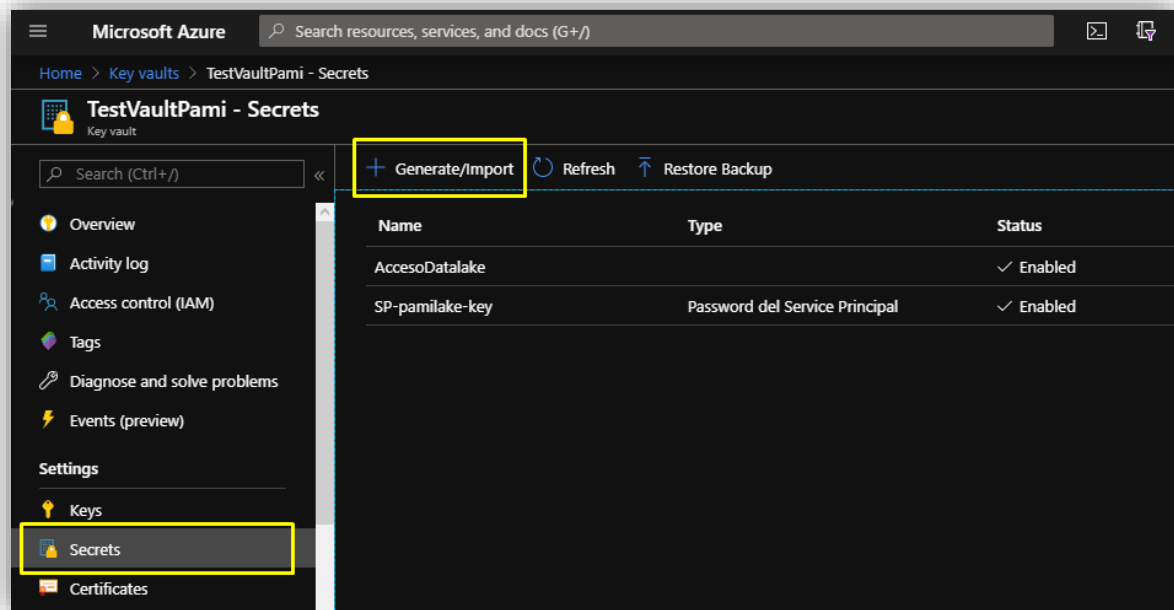


Una vez creado el Key Vault, y dado acceso a usuarios o SP (esto último puede modificarse posteriormente), procederemos a guardar secretos en nuestro Key Vault.



Creación de Secretos

Dentro del KV creado, vamos a Secrets – Generate/Import



Ingresamos el nombre del Secret y pegamos en Value el valor correspondiente

Upload options
Manual

Name * ⓘ
SP-PamilakeKey ✓

Value * ⓘ
Enter the secret.

Content type (optional)

Set activation date? ⓘ ☐

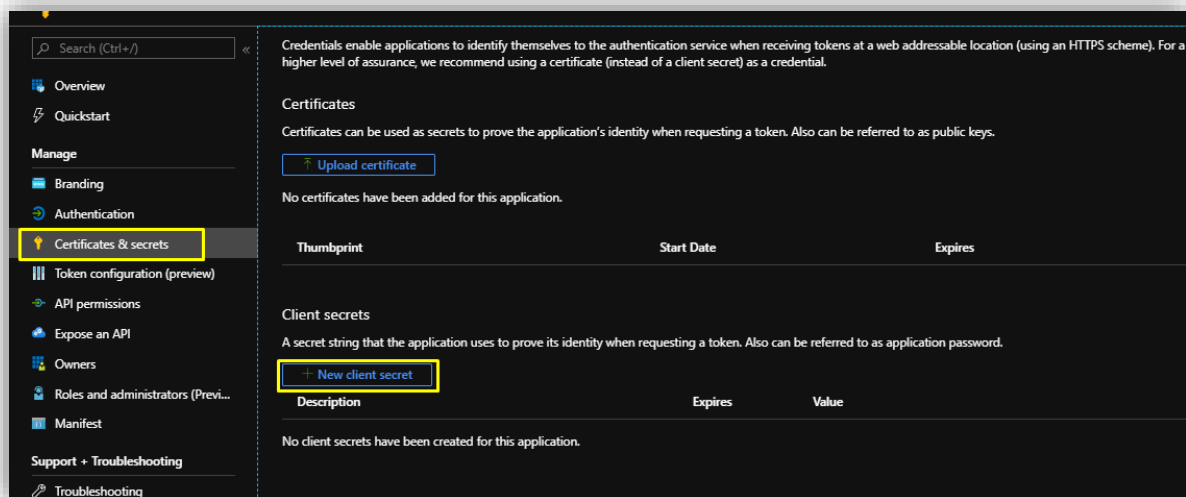
Set expiration date? ⓘ ☐

Enabled? ☒ Yes ☐ No

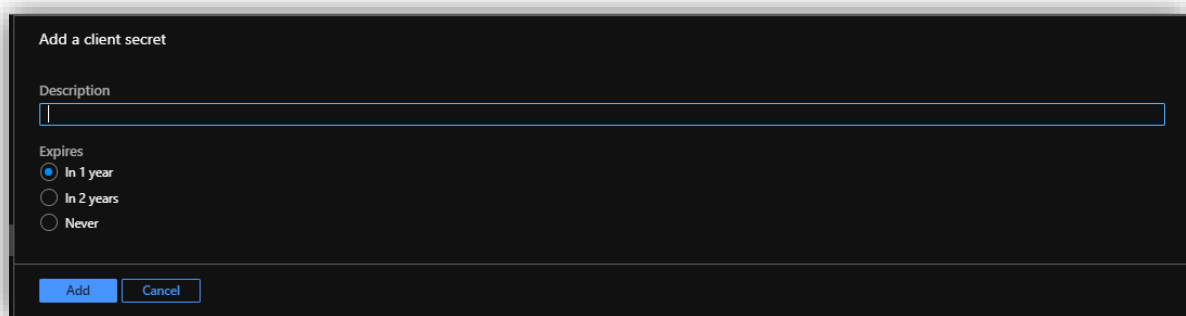
Create



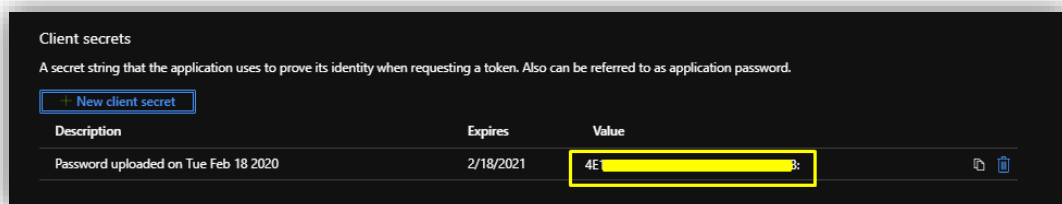
En nuestro caso, pegaremos en Values, el Secret del SP a trabajar, este lo tomamos accediendo al mismo haciendo click en Certificates & secrets – New client secret



Aquí podremos elegir el tiempo de expiración de este secreto, nosotros elegiremos Never.

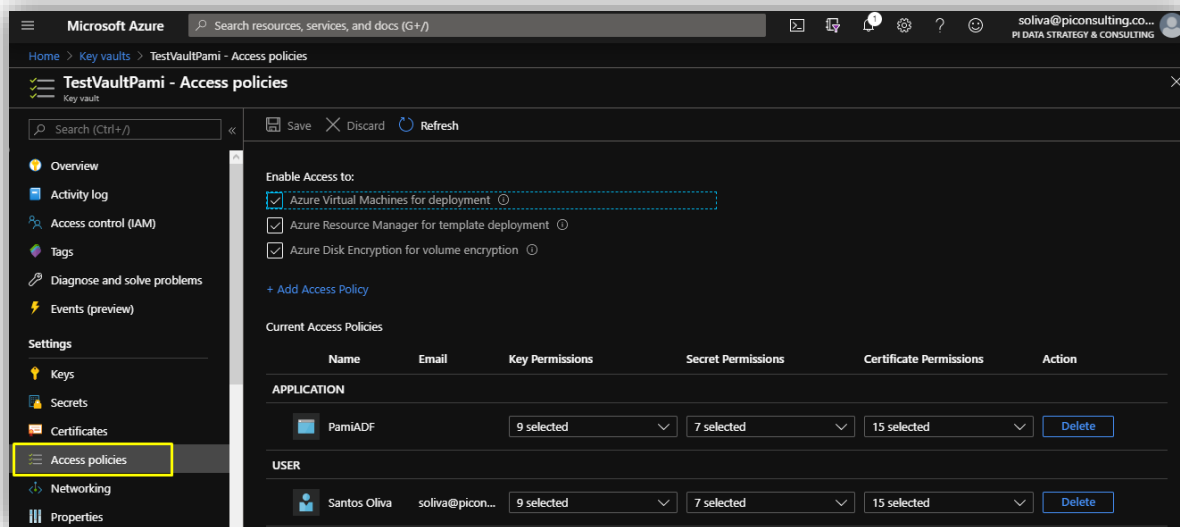


El valor resaltado será el valor a guardar. En caso de perderse, se podrá generar un nuevo client secret.





Para gestionar las políticas de acceso una vez creado el KV, solo basta con ir a la pestaña Access policies dentro de Settings.



Para este proyecto, hasta este punto, hemos guardado 2 secretos: el Key del DataLake y el Key del Service Principal



Storage Account

Microsoft Azure proporciona soluciones escalables y duraderas de almacenamiento, respaldo y recuperación en la nube para cualquier dato, grande o pequeño. Funciona con la infraestructura que ya tiene para mejorar de manera rentable sus aplicaciones existentes y su estrategia de continuidad comercial, y proporcionar el almacenamiento requerido por sus aplicaciones en la nube, incluidos texto no estructurado o datos binarios como video, audio e imágenes.

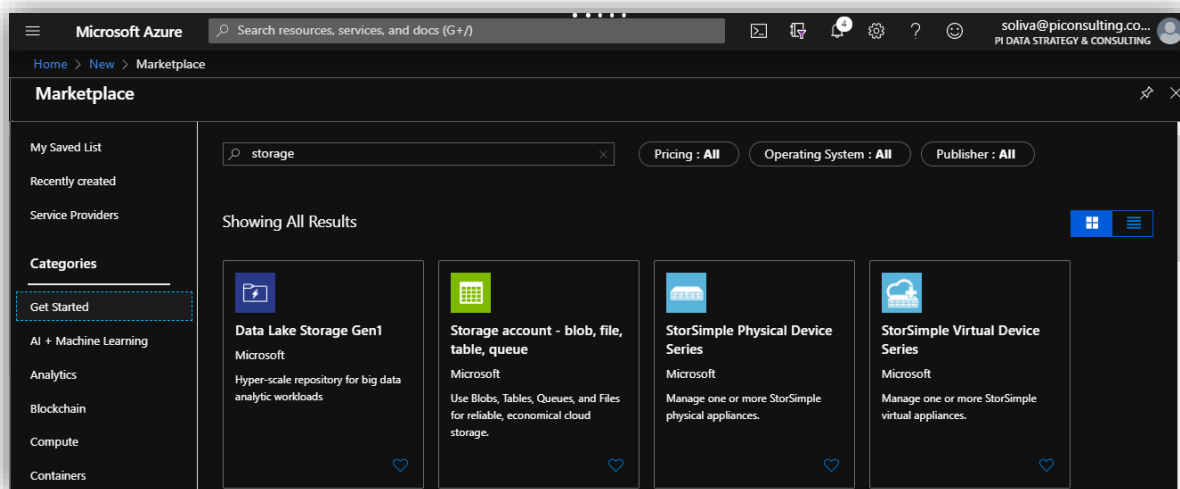
En nuestro caso utilizaremos el storage Gen2 cuyos archivos se guardarán como un File System

En un futuro referenciaremos al File system del Storage account como Data Lake

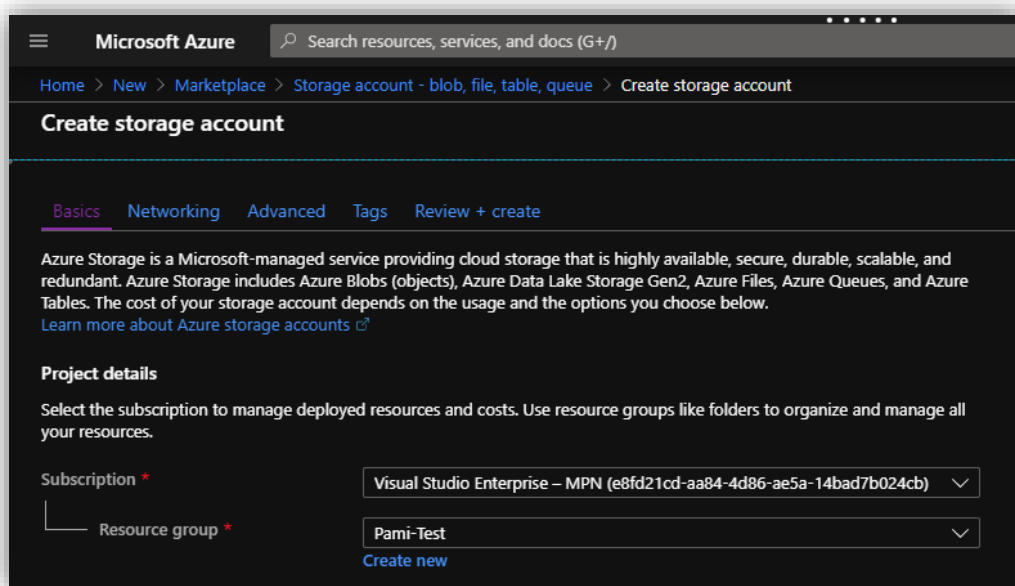


Creación de Storage Account

Para crear uno, vamos al Marketplace, y buscamos Storage Account – blob, file, table, queue y luego clickeamos en Crear



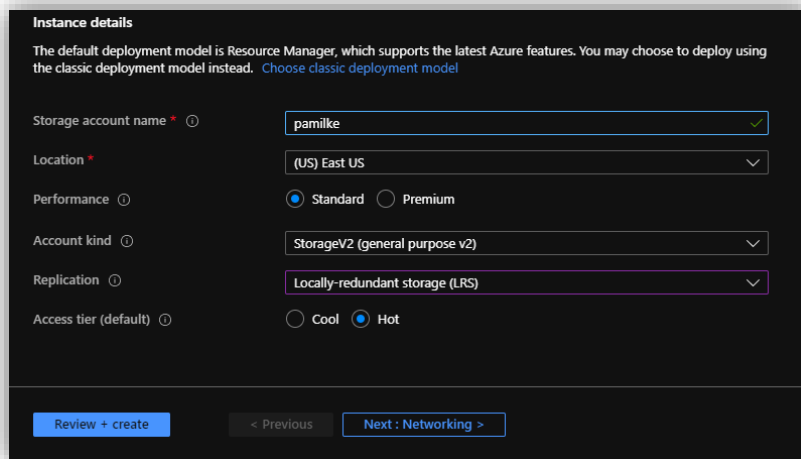
En la configuración del storage seleccionamos la suscripción donde se creará el recurso y el grupo de Recursos donde se localizaría el mismo



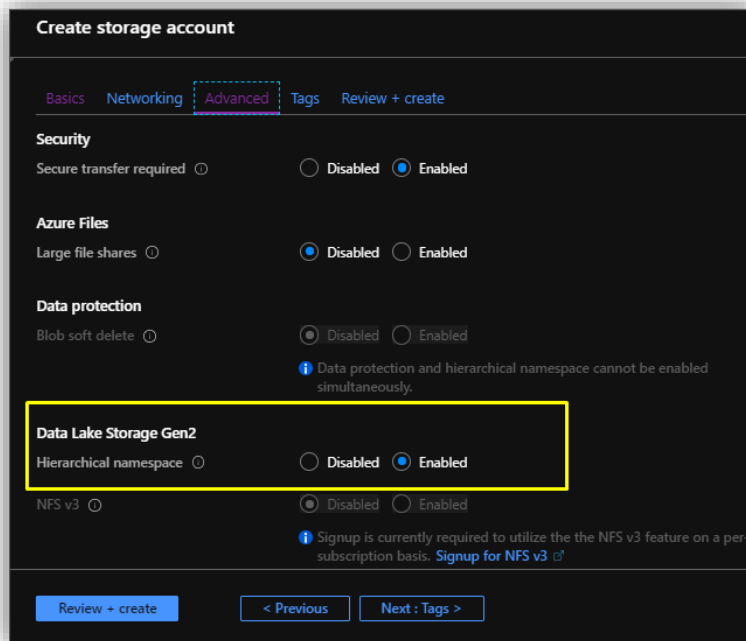
También daremos nombre al Storage, seleccionamos la localización, que debe ser East Us.

El rendimiento seleccionado será Standard y la replicación LRS por su relación precio-beneficio.

El tier de acceso sera Hot para tener un acceso mas rapido a la informacion.



Luego, en Advanced, habilitamos el nombre Jerárquico para tener carpetas reales dentro del Storage





Finalmente creamos el Storage

The screenshot shows the 'Create storage account' wizard in the Azure portal. The 'Review + create' tab is selected and highlighted with a dashed blue border. A green banner at the top indicates 'Validation passed'. The wizard is divided into sections: Basics, Networking, and Advanced. The Basics section contains the following configuration details:

Property	Value
Subscription	Visual Studio Enterprise – MPN
Resource group	Pami-Test
Location	(US) East US
Storage account name	pamilke
Deployment model	Resource manager
Account kind	StorageV2 (general purpose v2)
Replication	Locally-redundant storage (LRS)
Performance	Standard
Access tier (default)	Hot

The Networking section shows:

Property	Value
Connectivity method	Public endpoint (all networks)

The Advanced section is currently empty. At the bottom, there are four buttons: 'Create' (blue), '< Previous' (blue), 'Next >' (disabled), and 'Download a template for automation' (blue).



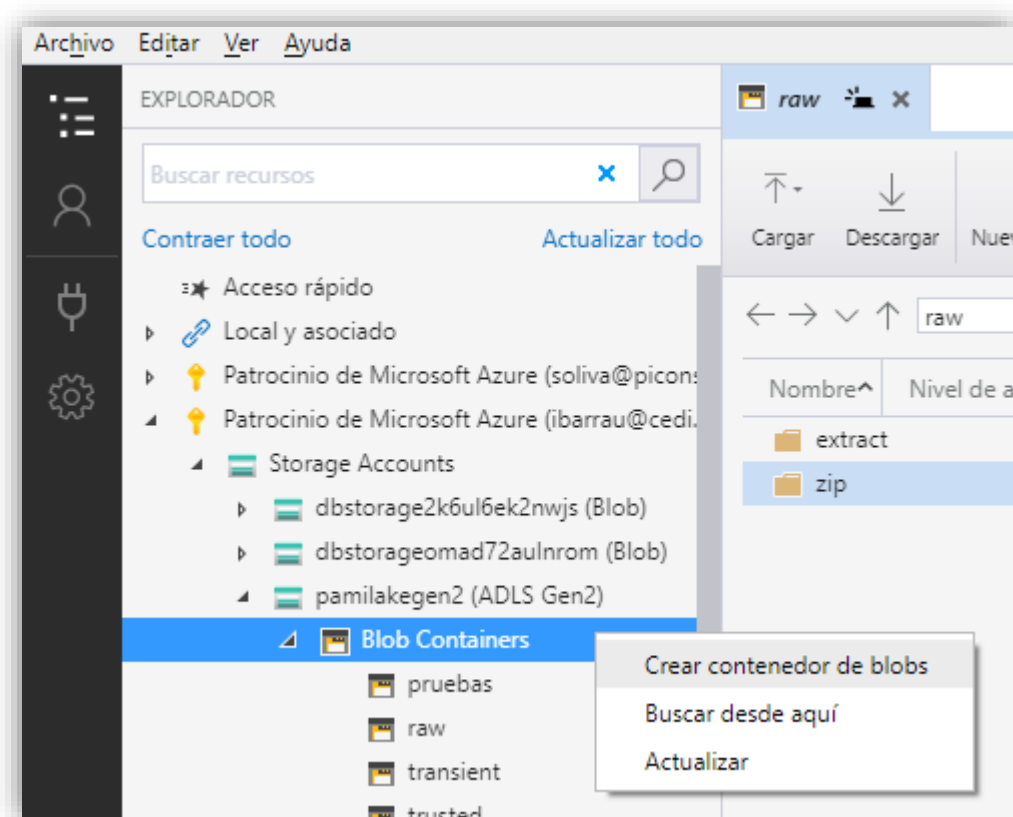
Configuración del Datalake

Utilizando el programa Microsoft Azure Storage Explorer, Logueamos nuestro usuario y podremos gestionar nuestro Datalake

Gestión de Blobs

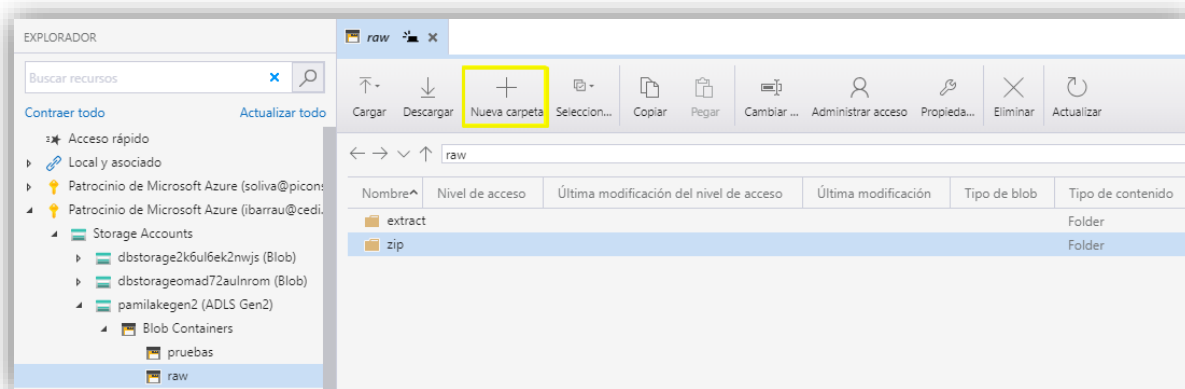
Utilizaremos Blob Containers para almacenar todo tipo de archivos, para crear uno en nuestra Storage Account, seguiremos los siguientes pasos:

- Hacer click derecho en Blob Containers
- Crear contenedor de blobs
- colocaremos el nombre que le daremos





Para crear una carpeta dentro del mismo simplemente se hará click en nueva carpeta y se colocará su nombre



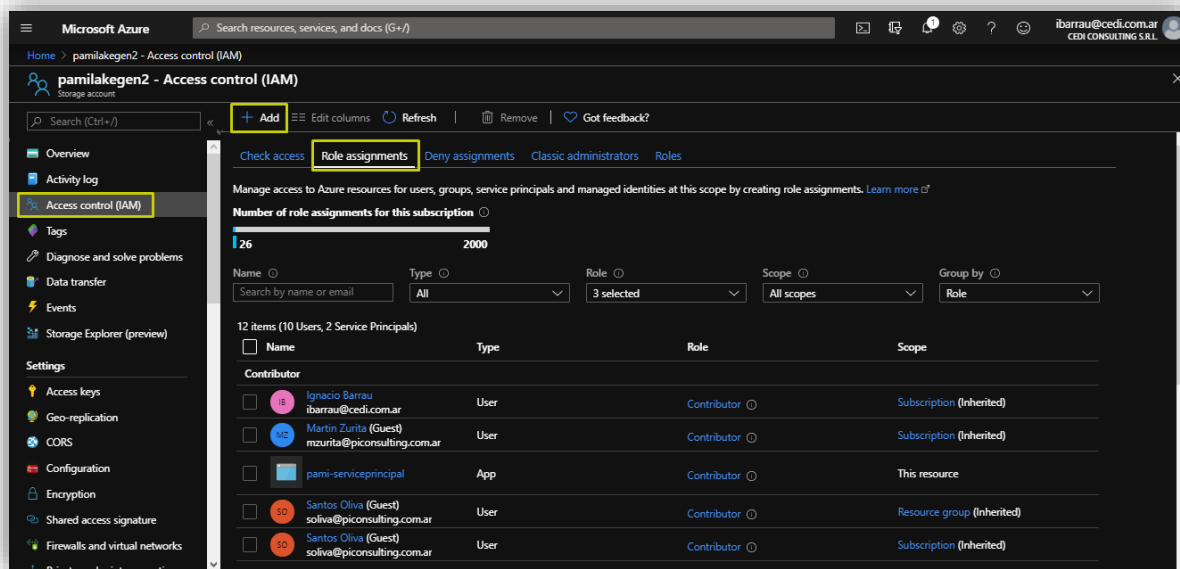


Gestión de permisos

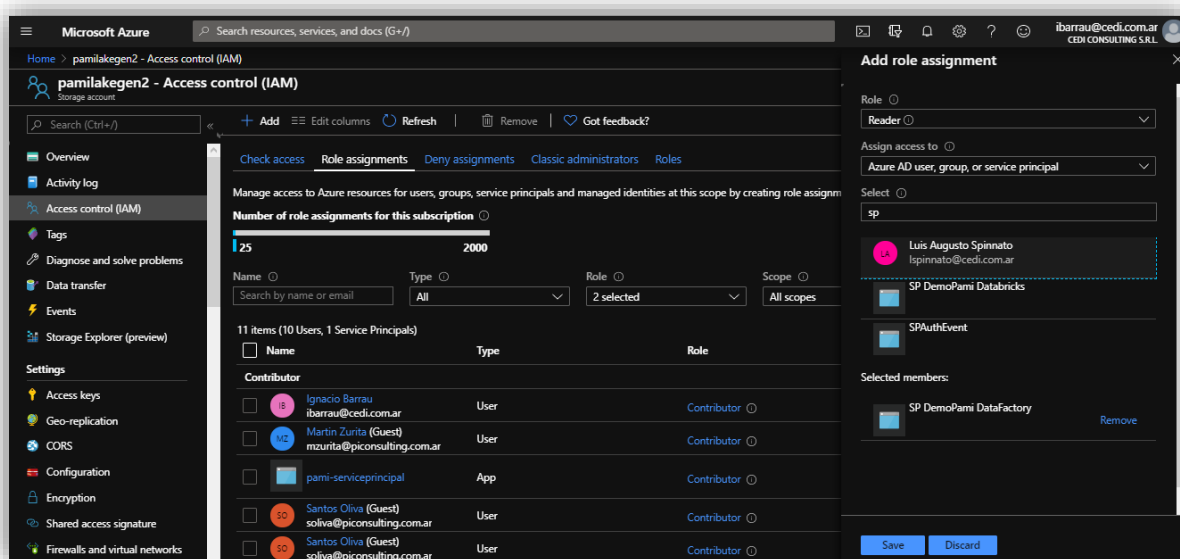
En este tipo de proyectos es muy importante gestionar los permisos de cada usuario, en el ejemplo lo haremos con un SP, que como previamente mencionamos, es un usuario programático.

Lo primero que hay que hacer, es darle permisos de lectura al lake en general, esto se realiza desde el portal de Azure, dentro del Datalake,

Access control – Role Assignments- Add

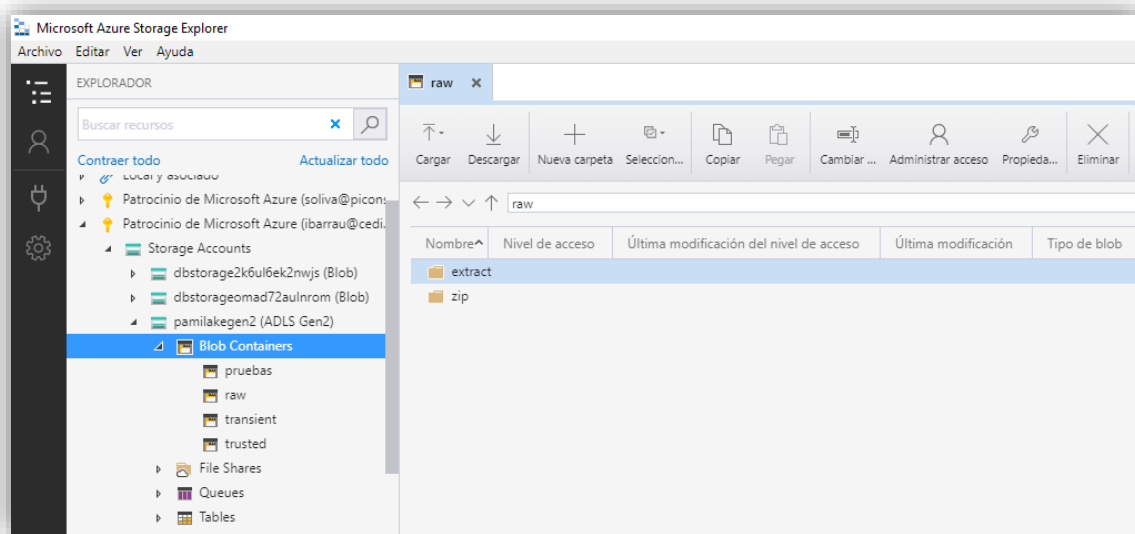


Luego seleccionaremos el SP al cual le daremos permisos, y el rol que le otorgaremos. Después haremos click en Save

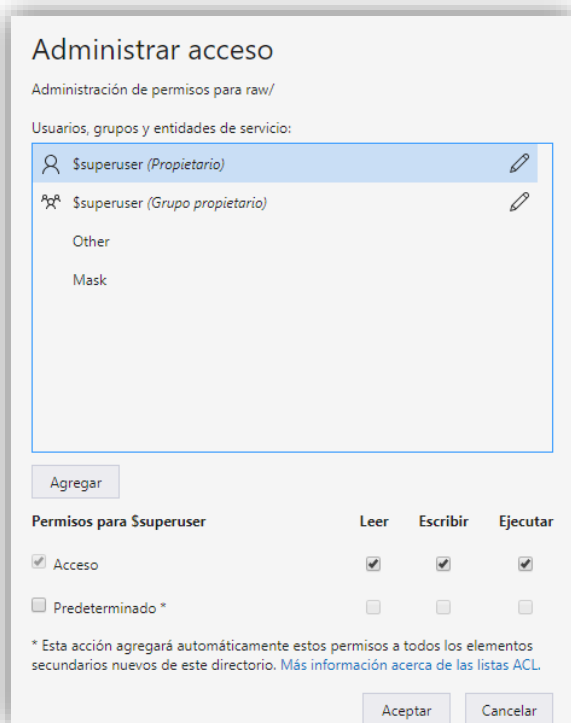


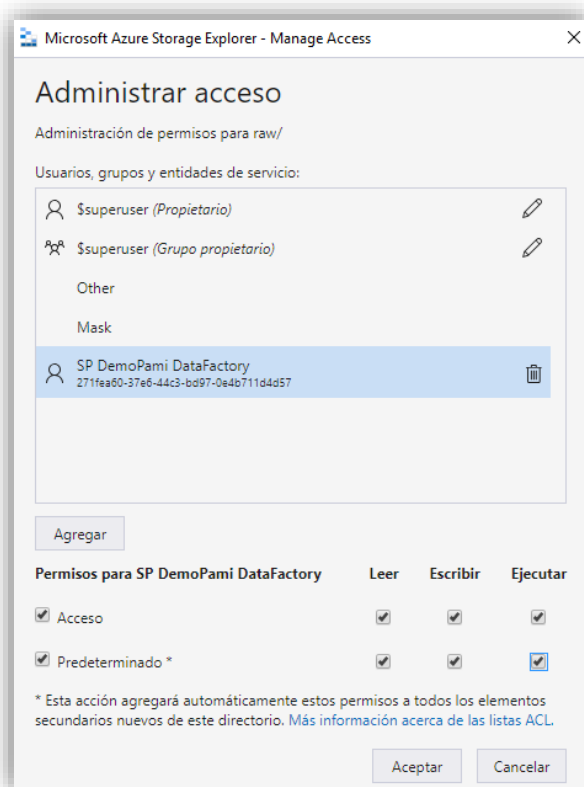
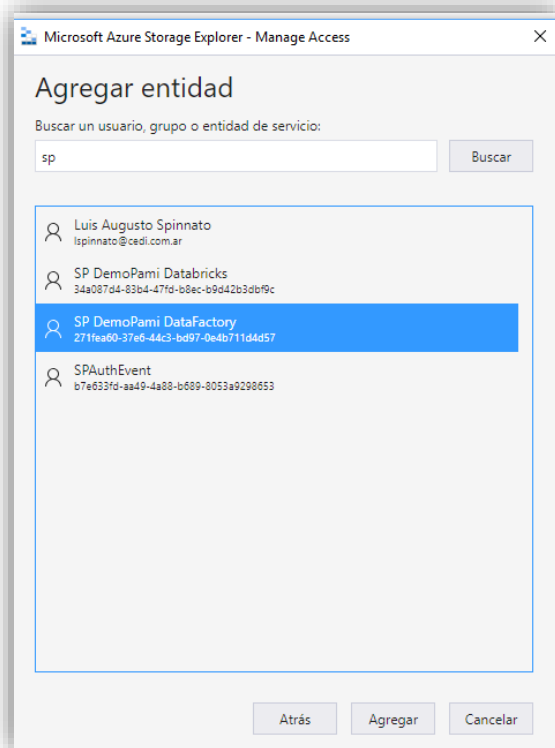


Utilizando la aplicación de Microsoft Azure Storage Explorer



Iremos a Blob Containers y haremos click derecho en
RAW – Administrar Acceso – Agregar







Seleccionaremos el acceso que le daremos a Storage, y el predeterminado que se le dará a las carpetas y archivos que se crearan dentro del mismo

Gestionaremos nuestros SP de la siguiente manera:

- El utilizado por DataFactory tendrá acceso al Storage en su integridad, con permisos de lectura, escritura y ejecución.
- El utilizado por DataBricks, tendrá solo permiso de lectura en Raw, mientras que en el resto del DataLake, tendrá permisos totales.



Databricks

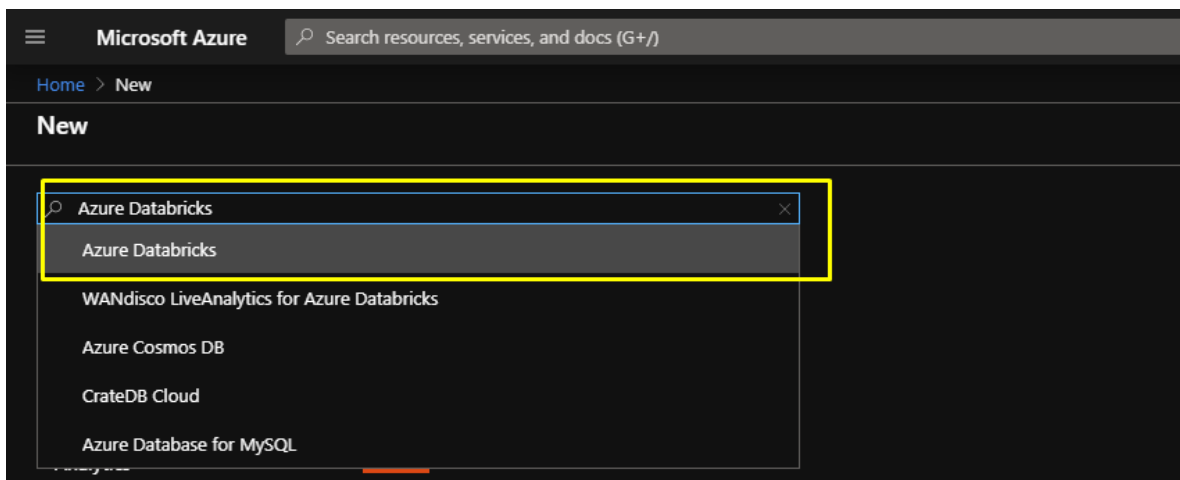
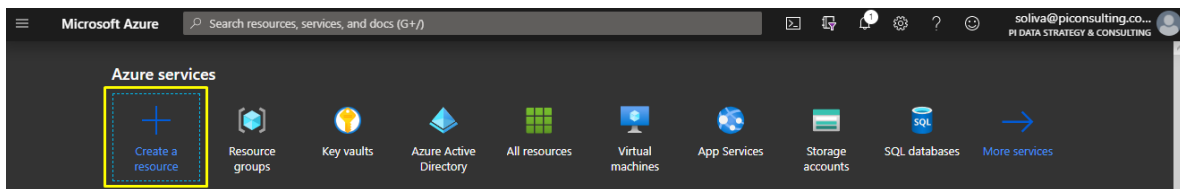
Azure Databricks es una versión totalmente administrada del motor de análisis y procesamiento de datos de código abierto Apache Spark. Azure Databricks es una plataforma de macrodatos y aprendizaje automático basada en la nube, segura y de nivel empresarial.

Databricks proporciona un entorno de área de trabajo como servicio de Apache Spark orientado a cuadernos, lo que facilita la administración de clústeres y el examen de los datos de forma interactiva.



Creación de un área de trabajo de Azure Databricks

1. Abra Azure Portal.
2. Haga clic en **Crear un recurso** en la parte superior izquierda.
3. Busque "Databricks"
4. Seleccione *Azure Databricks*.
5. En la página de Azure Databricks, seleccione *Crear*.





Para crear el recurso, se solicitará la suscripción y el grupo de recursos donde se creará, el nombre que le daremos a nuestro Workspace, la localización de este y el Plan de tarifa, este último se recomienda establecerlo en Premium para tener mejor manejo de usuarios y roles.

Luego haremos click en Review + Create, luego se validará y creará.

Microsoft Azure Search resources, services, and docs (G+)

Home > Pami-Test > New > Azure Databricks > Azure Databricks Service

Azure Databricks Service

Basics * Networking Tags Review + Create

Project Details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *

Resource group * [Create new](#)

Instance Details

Workspace name *

Location *

Pricing Tier *

[Review + Create](#) [Next : Networking >](#)

Azure Databricks Service

✓ Validation Succeeded

Basics * Networking Tags Review + Create

Summary

Basics

Workspace name	Pami-Databricks-Test
Subscription	Visual Studio Enterprise - MPN
Resource group	Pami-Test
Location	East US
Pricing Tier	premium

Networking

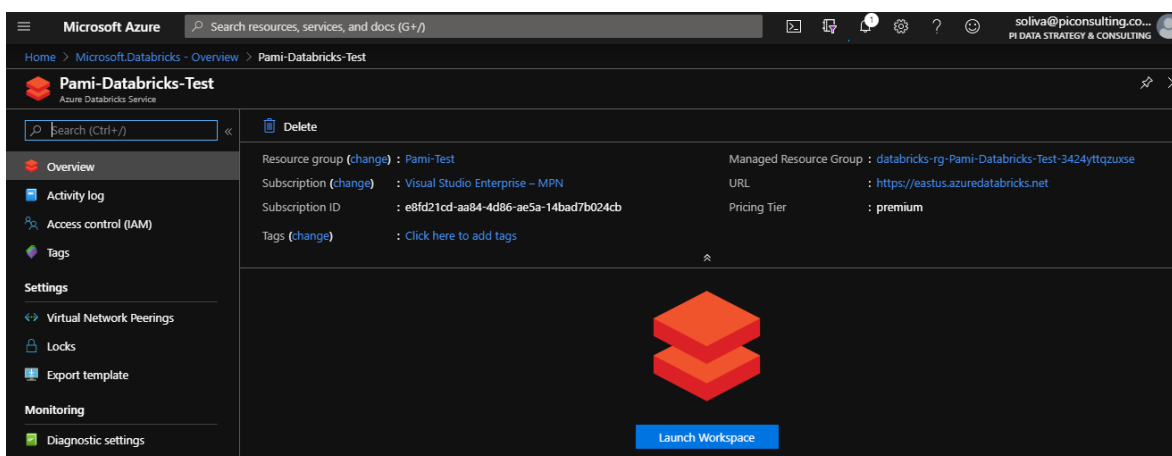
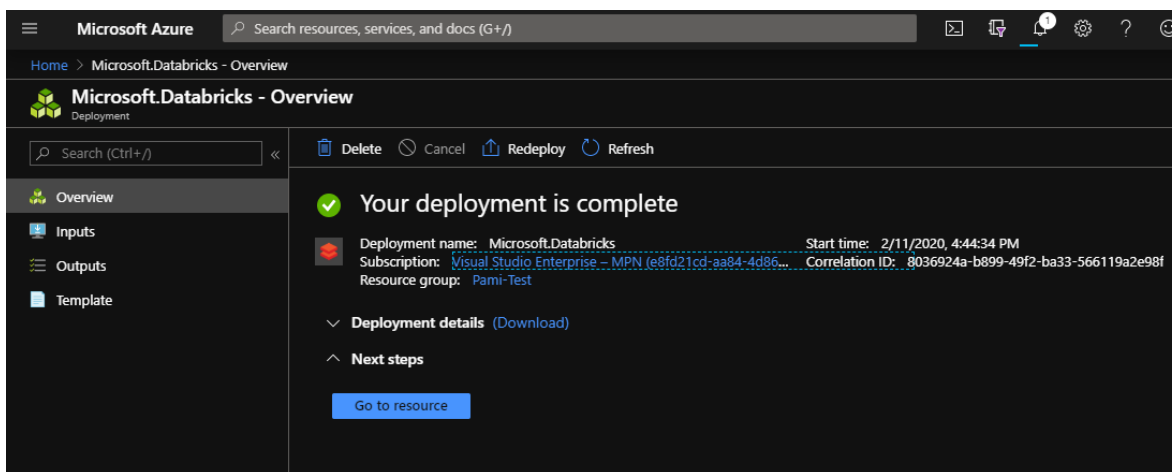
Deploy Azure Databricks workspace in your own Virtual Network (VNet)	No
--	----

[Create](#) [Previous : Tags](#)



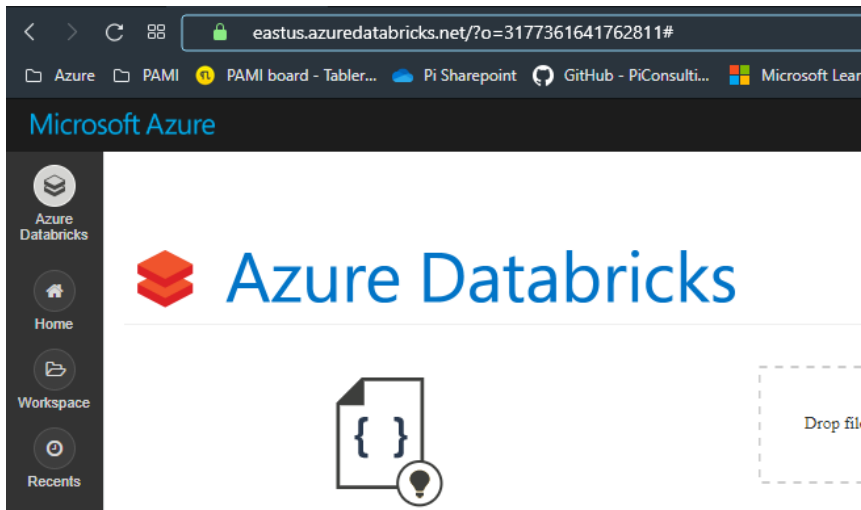
Para acceder al recurso creado,

- 1- iremos a la pestaña Overview
- 2- haremos click en Go to resource
- 3- Una vez dentro del recurso, haremos click en Launch Workspace

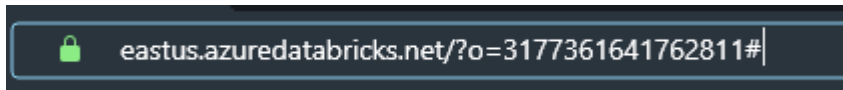




El Workspace de Databricks se abra en una pestaña nueva



Esta sera la URL que identificara nuestro Databricks





Creación de Clusters

Los cuadernos están respaldados por clústeres o equipos en red que colaboran para procesar los datos. El primer paso es crear un clúster.

Necesitaremos crear un cluster para que ejecute nuestras notebooks

Dentro del Workspace, haremos click en la pestaña Clusters y dentro de la misma en Create Cluster

The image displays two screenshots of the Azure Databricks web interface. The top screenshot shows the main workspace with the 'Clusters' tab highlighted in the sidebar. The bottom screenshot shows the 'Clusters' page with the 'Create Cluster' button highlighted.

Top Screenshot: Azure Databricks Main Workspace

- Header: Microsoft Azure, PORTAL, soliva@piconsulting.com.ar
- Left Sidebar: Azure Databricks, Home, Workspace, Recents, Data, **Clusters** (highlighted), Jobs, Search.
- Main Content Area:
 - Azure Databricks logo and title.
 - Three main actions: Explore the Quickstart Tutorial, Import & Explore Data, Create a Blank Notebook.
 - Common Tasks: New Notebook, Create Table, New Cluster.
 - Recents: Recent files appear here as you work.
 - Documentation: Databricks Guide, Getting Started, SparkR Overview.

Bottom Screenshot: Azure Databricks Clusters Page

- Header: Microsoft Azure, PORTAL, soliva@piconsulting.com.ar
- Left Sidebar: Same as the top screenshot, with 'Clusters' highlighted.
- Main Content Area:
 - Clusters Pools tabs.
 - + Create Cluster button (highlighted).
 - Filters: All, Created by me, Accessible by me, Filter.
 - Interactive Clusters: 0 clusters, 0 pinned. No clusters found.
 - Automated Clusters: No clusters found.



A este nuevo cluster:

- Le daremos nombre,
- Seleccionaremos el modo de Cluster (nosotros utilizaremos High Concurrency para que este disponible para el uso de varios usuarios simultáneamente),
- Seleccionamos la versión de Runtime acorde a las versiones de los lenguajes que utilizaremos (se recomienda la última versión estable)
- Seleccionamos en **Python Version** la versión 3, que es la que utilizaremos
- Tildaremos **Enable autoscaling** para tener un manejo automático de recursos acorde a las necesidades de procesamiento
- Tildaremos **Terminate after** y pondremos 40 minutos, para programar el apagado del cluster en caso de inactividad de este
- En **Worker Type** seleccionaremos Standard_DS3_v2 con un mínimo de 1 Worker y un máximo de 2 Workers
- En **Driver type** seleccionaremos Same as Worker

Microsoft Azure

Create Cluster

New Cluster

Cancel Create Cluster

1-2 Workers: 14.0-28.0 GB Memory, 4-8 Cores, 0.75 DBU
1 Driver: 14.0 GB Memory, 4 Cores, 0.75 DBU

Cluster Name
Cluster_pami-Test

Cluster Mode
High Concurrency

Pool
None

Databricks Runtime Version
Latest stable (Scala 2.11)

Python Version
3

Autopilot Options
☒ Enable autoscaling
☒ Terminate after 40 minutes of inactivity

Worker Type
Standard_DS3_v2 14.0 GB Memory, 4 Cores, 0.75 DBU

Min Workers Max Workers
1 2

Driver Type
Same as worker 14.0 GB Memory, 4 Cores, 0.75 DBU



Creación de Scope

Los scopes gestionan el acceso a key vaults desde Databricks, para crear uno nuevo, es necesario agregar a la URL de nuestro recurso el sufijo “secrets/createScope”

URL:

URL + sufijo

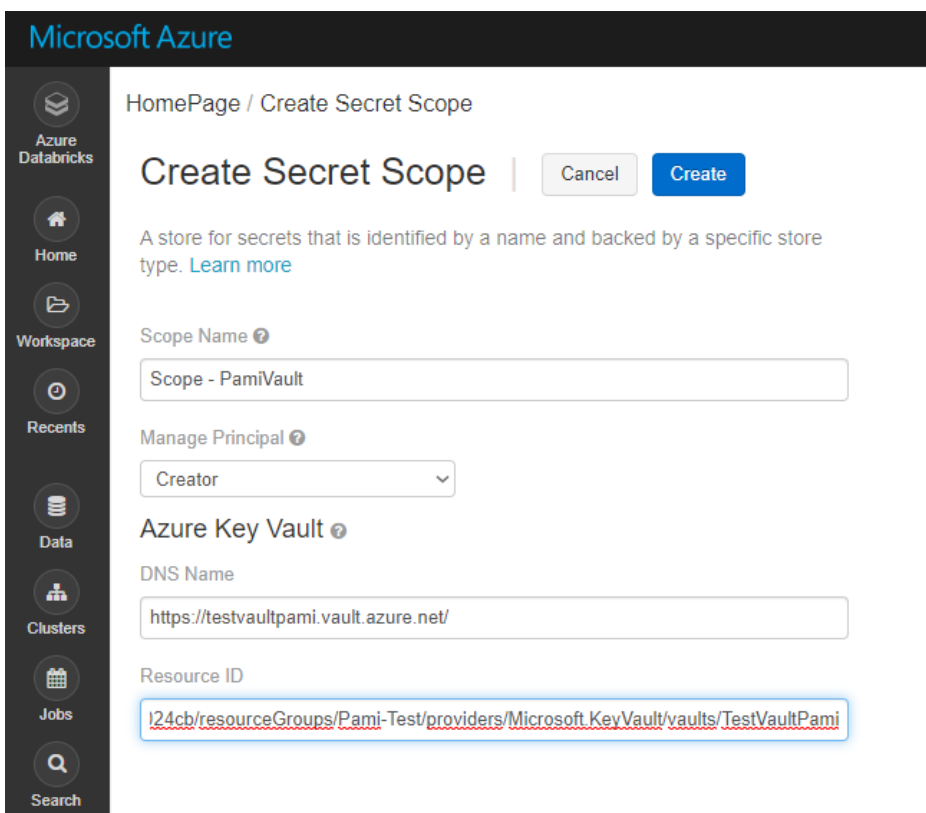
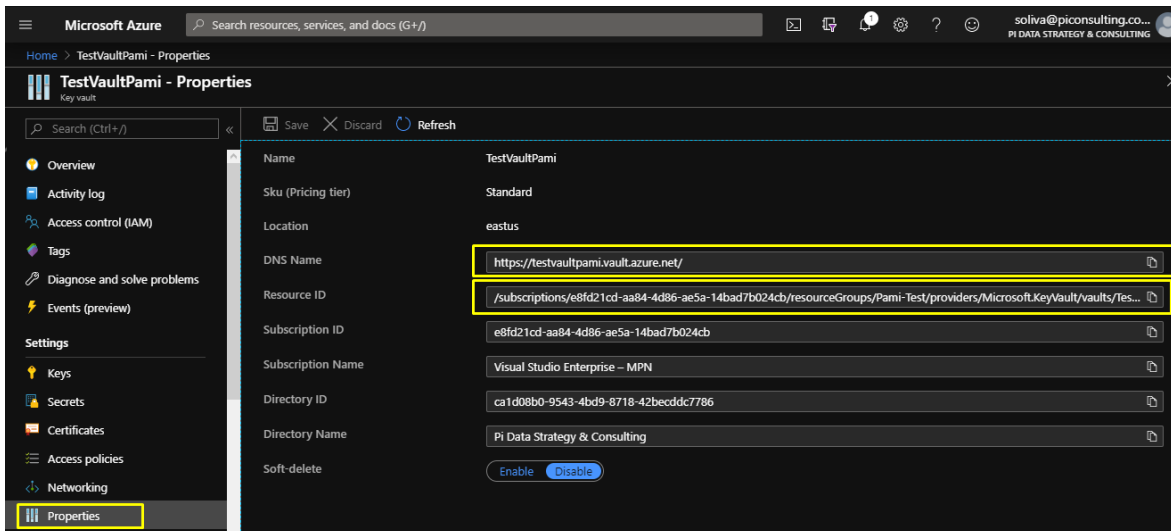
Aquí tendremos que cargar los datos del scope:

- Le daremos nombre
- Seleccionaremos el **Manage Principal** (Creador)
- Cargaremos los datos del Key Vault

Importante: guardar el nombre del Scope creado



Los datos del Key Vault requeridos se encuentran dentro del recurso, en la pestaña **Propiedades** en los campos marcados.





Creación de un Cuaderno

1. En el menú de la izquierda del área de trabajo de Databricks, seleccione **Inicio**.
2. Haga clic con el botón derecho en la carpeta principal.
3. Seleccione **Crear**.
4. Seleccione **Notebook** (Cuaderno).
5. Asigne el nombre **Primer cuaderno** al cuaderno.
6. Establezca el **lenguaje** en **Python**.
7. Seleccione el clúster al que se va a asociar este cuaderno.

Puede usar cuadernos de Apache Spark para:

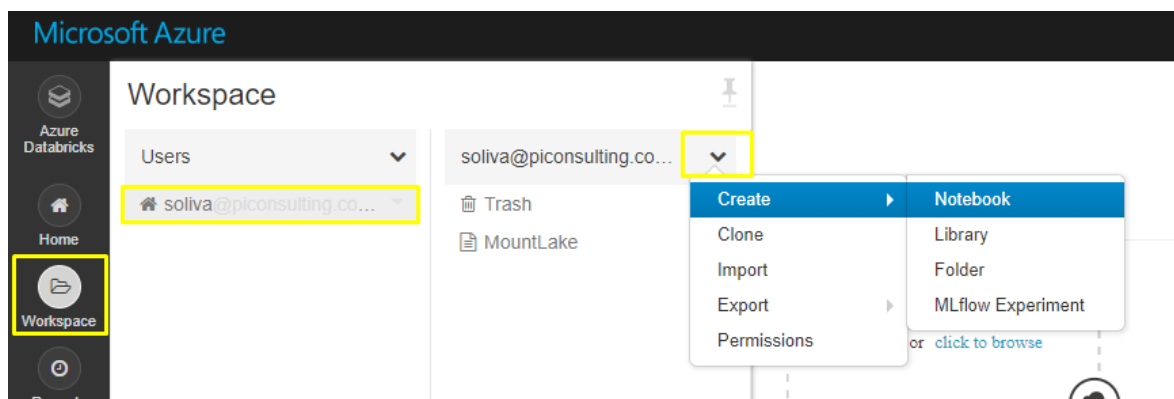
- Leer y procesar archivos de gran tamaño y conjuntos de datos
- Consultar, examinar y visualizar conjuntos de datos
- Unir distintos conjuntos de datos encontrados en Data Lake
- Entrenar y evaluar modelos de Machine Learning
- Procesar flujos de datos en directo
- Realizar análisis de grandes conjuntos de datos gráficos y redes sociales



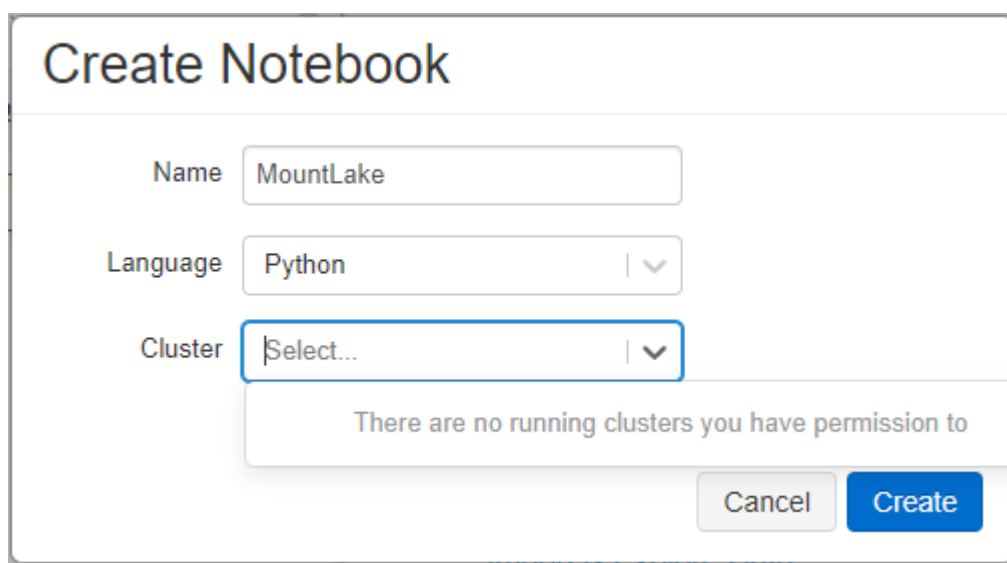
Montar Datalake en Databricks

Se le dice "montar" a la gestión de acceso del Datalake desde Databricks, esto se realiza de la siguiente manera.

Primero crearemos una notebook que ejecutara el script correspondiente; esto se hace desde la pestaña Workspace, se selecciona el usuario y luego Create - Notebook

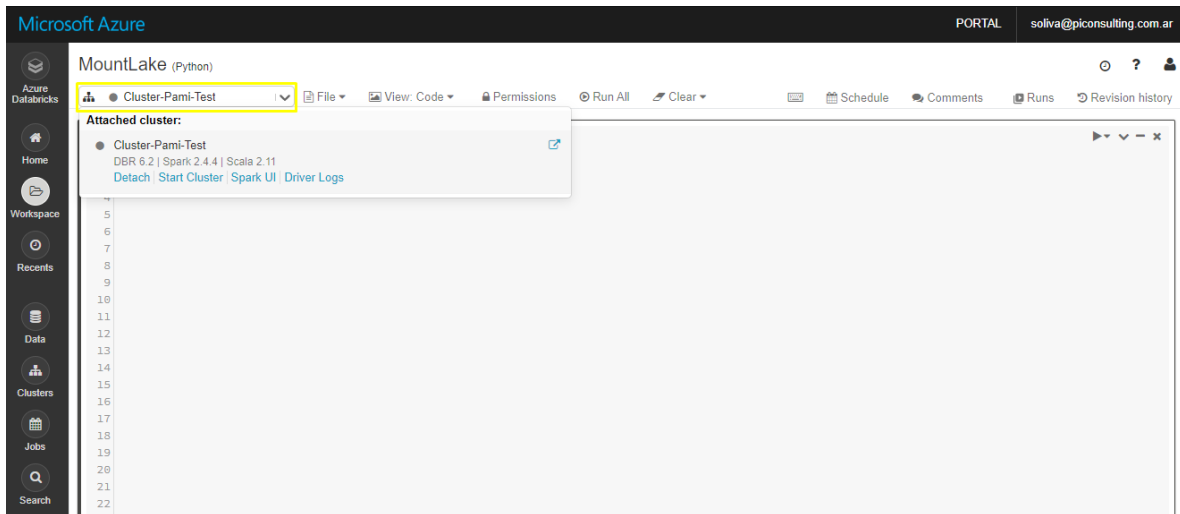


Al momento de crearla, solo te deja seleccionar entre los Clusters activos, se puede crear sin seleccionarlo y luego hacerlo desde la notebook

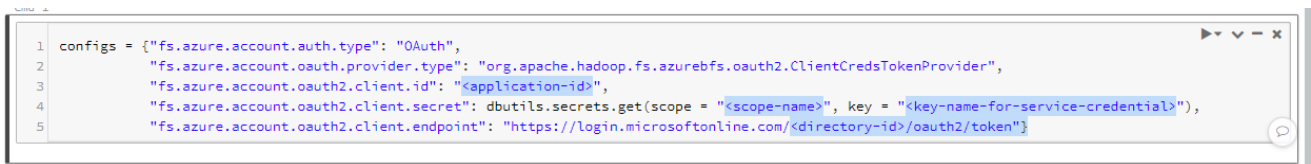




Una vez Creada la notebook, podremos seleccionar que cluster ejecutara nuestros comandos



Primero cargaremos la configuración, modificando los campos resaltados en el siguiente screenshot



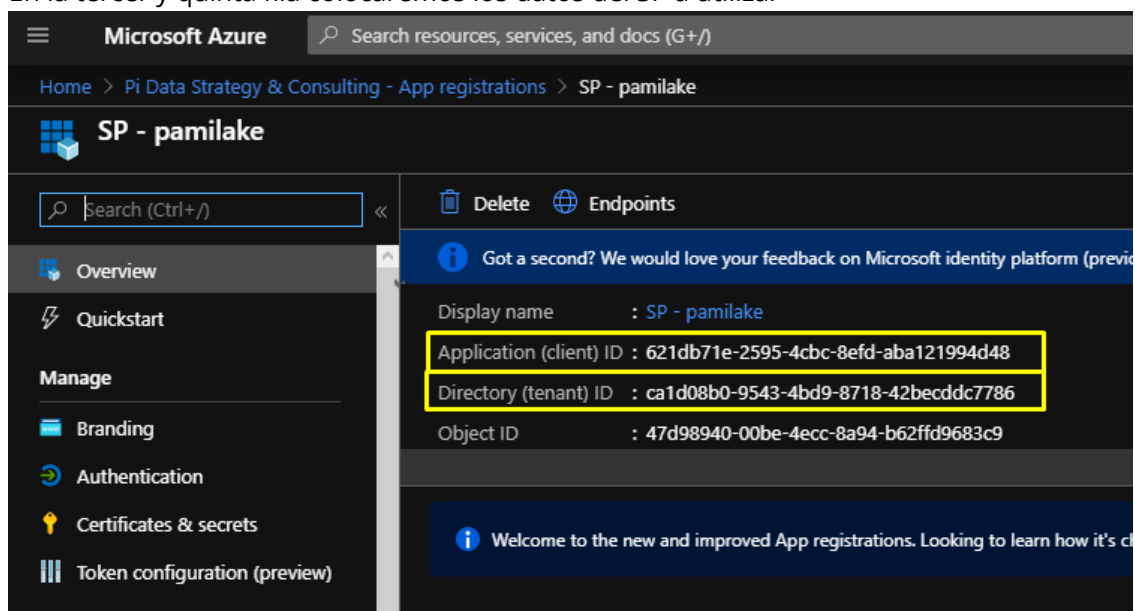
Nos quedara de la siguiente manera:

En la cuarta fila pondremos el nombre del Scope a utilizar y el nombre del secreto guardado en el key vault a utilizar

```
configs = {"fs.azure.account.auth.type": "OAuth",
           "fs.azure.account.oauth.provider.type": "org.apache.hadoop.fs.azurebfs.oauth2.ClientCredsTokenProvider",
           "fs.azure.account.oauth2.client.id": "621db71e-2595-4cbc-8efd-aba121994d48",
           "fs.azure.account.oauth2.client.secret": dbutils.secrets.get(scope = "Scope - PamiVault", key = "SP-pamilake-key"),
           "fs.azure.account.oauth2.client.endpoint": "https://login.microsoftonline.com/ca1d08b0-9543-4bd9-8718-42becddc7786/oauth2/token"}
# Optionally, you can add <directory-name> to the source URI of your mount point.
```




En la tercer y quinta fila colocaremos los datos del SP a utilizar



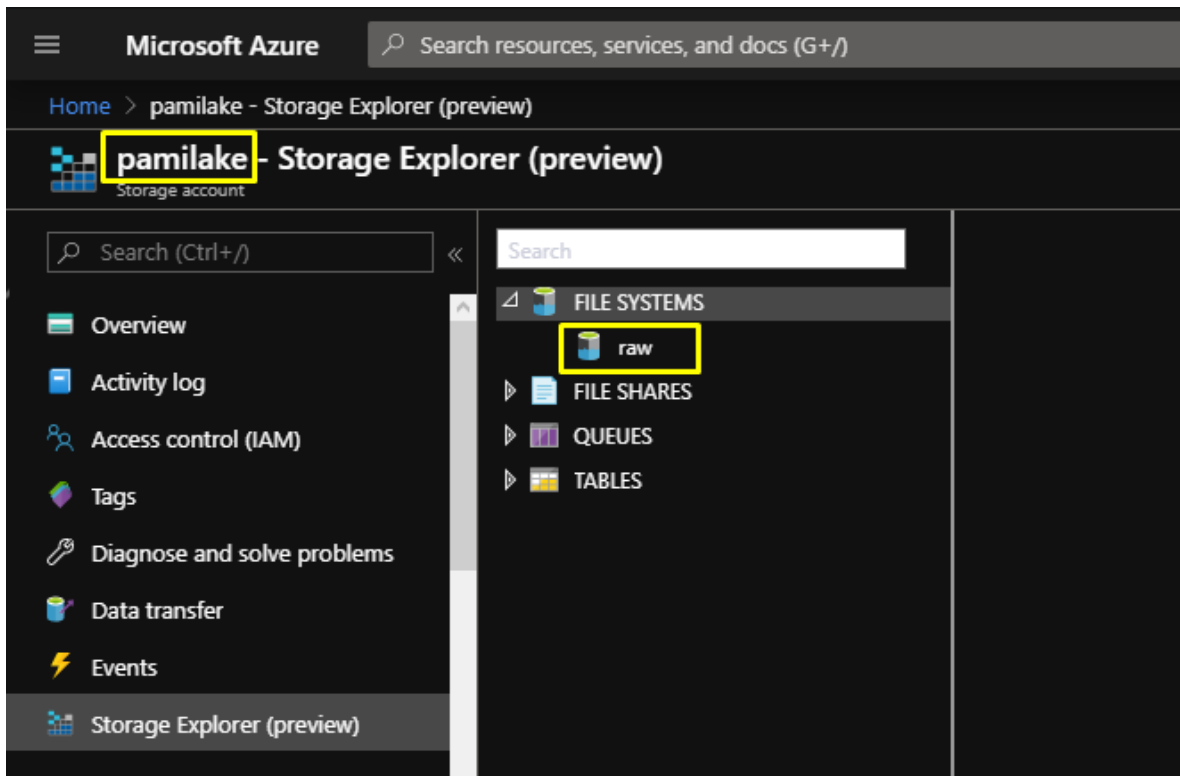
En la cuarta fila, colocaremos el nombre del Scope creado y el nombre del Key a consultar

Ya creando el script para montar, esta será la plantilla

```
dbutils.fs.mount(  
    source = "abfss://<file-system-name>@<storage-account-name>.dfs.core.windows.net/",  
    mount_point = "/mnt/<mount-name>",  
    extra_configs = configs)
```

Que luego de editar según corresponda nos quedara de esta manera

```
1 dbutils.fs.mount(  
2     source = "abfss://raw@pamilake.dfs.core.windows.net/",  
3     mount_point = "/mnt/rawdata",  
4     extra_configs = configs)
```



Primero colocaremos el nombre del file system a montar, el nombre de la cuenta de storage (mostrados en el anterior screenshot), y el nombre que se le quiera dar al punto donde se montara (se define a discreción del admin).

En este punto, se podrá acceder al contenido del Data lake como si se estuviera recorriendo un file system que este cargado en el mount point.



Plantilla

```
# Nombre del scope utilizado para acceder al key vault que almacena los datos
scope_name = "key-vault-secrets"

# Nombre del Blob del Lake donde queremos acceder
filesystemName = "trusted"

# Id de aplicación, se genera con el scope y el nombre de la misma
application_id = dbutils.secrets.get(scope_name, "sp-dashboards-client-id")

# Id de directorio, generado también a partir del scope
directory_id = dbutils.secrets.get(scope_name, "azure-tenant-id")

# Nombre del secreto al cual se quiere acceder
secret = dbutils.secrets.get(scope_name, "sp-dashboard-client-key-db")


configs = {"fs.azure.account.auth.type": "OAuth",
           "fs.azure.account.oauth.provider.type":
"org.apache.hadoop.fs.azurebfs.oauth2.ClientCredsTokenProvider",
           "fs.azure.account.oauth2.client.id": application_id,
           "fs.azure.account.oauth2.client.secret": secret,
           "fs.azure.account.oauth2.client.endpoint": "https://login.microsoftonline.com/" +
directory_id + "/oauth2/token"}


adlsGen2AccountName = dbutils.secrets.get(scope_name, "ADLS-Gen2-Account-Name")
abfsUri = "abfss://" + filesystemName + "@" + adlsGen2AccountName + ".dfs.core.windows.net/"


dbutils.fs.mount(source = abfsUri ,
                 mount_point = "/mnt/" + filesystemName ,
                 extra_configs = configs)
```



Tratamiento de Dataframes

Los Dataframes derivan de estructuras de datos conocidas como conjuntos de datos distribuidos resilientes (RDD). Los RDD y los marcos de datos son colecciones de datos distribuidas inmutables.

Analizando el significado de cada una de sus iniciales, tendremos:

- **Resiliente:** son tolerantes a fallas, por lo que si parte de su operación falla, Spark recupera rápidamente el cálculo perdido.
- **Distribuido:** los RDD se distribuyen a través de máquinas en red conocidas como clúster.
- **DataFrame:** una estructura de datos donde los datos se organizan en columnas con nombre, como una tabla en una base de datos relacional, pero con optimizaciones más completas.

Sin las columnas con nombre y los tipos declarados proporcionados por un esquema que detalle la metadata, Spark no sabría cómo optimizar la ejecución de ningún cálculo. Dado que los marcos de datos tienen un esquema, utilizan el Optimizador de Catalyst para determinar la forma óptima de ejecutar su código.

Navegar File System

Para más información, visitar

<https://docs.databricks.com/data/databricks-file-system.html>

Para ver los directorios cargados

1 %fs mounts	
mountPoint	source
/mnt/pruebas	abfss://pruebas@pamilakegen2.dfs.core.windows.net/
/mnt/trusted	abfss://trusted@pamilakegen2.dfs.core.windows.net/
/mnt/transientData	abfss://transient@pamilakegen2.dfs.core.windows.net/
/mnt/raw	abfss://raw@pamilakegen2.dfs.core.windows.net/
/	DatabricksRoot
/databricks-datasets	databricks-datasets
/databricks-results	databricks-results

Command took 7.21 seconds -- by ibarrau@cedi.com.ar at 19/2/2020 10:20:16 on Cluster A



Para navegar los directorios cargados

1	%fs ls /mnt/pruebas
---	---------------------

path	name
dbfs:/mnt/pruebas/BENEFICIO/	BENEFICIO/
dbfs:/mnt/pruebas/CONSUMO/	CONSUMO/
dbfs:/mnt/pruebas/Extracts/	Extracts/
dbfs:/mnt/pruebas/LU_MSC_RTF/	LU_MSC_RTF/

1	%fs ls /mnt/pruebas/Test-Origen/
---	----------------------------------

path
dbfs:/mnt/pruebas/Test-Origen/Customer Churn.csv

Para acceder el archivo desde Spark, el comando será:

1	DataframeTest = spark.read.csv("/mnt/pruebas/Test-Origen/Customer Churn.csv",header=True,sep=";")
---	---

► (1) Spark Jobs

► DataframeTest: pyspark.sql.dataframe.DataFrame = [LoyaltyID: string, Customer ID: string ... 19 more fields]

Para observarlo

1	display(DataframeTest)
---	------------------------

► (1) Spark Jobs

LoyaltyID	Customer ID	Senior Citizen	Partner	Dependents	Tenure	Phone Service	Multiple Lines	Internet Service	Online Security	Online Backup	Device Protection
318537	7590-VHVEG	No	Yes	No	1	No	No phone service	DSL	No	Yes	No
152148	5575-GNVDE	No	No	No	34	Yes	No	DSL	Yes	No	Yes
326527	3668-QPYBK	No	No	No	2	Yes	No	DSL	Yes	Yes	No
845894	7795-CFOCW	No	No	No	45	No	No phone service	DSL	Yes	No	Yes

< Showing the first 1000 rows.



Para observar la metadata o schema de la tabla utilizaremos el siguiente comando

```
1 dataframeTest.printSchema()
```

```
root
|-- LoyaltyID: string (nullable = true)
|-- Customer ID: string (nullable = true)
|-- Senior Citizen: string (nullable = true)
|-- Partner: string (nullable = true)
|-- Dependents: string (nullable = true)
|-- Tenure: string (nullable = true)
|-- Phone Service: string (nullable = true)
|-- Multiple Lines: string (nullable = true)
|-- Internet Service: string (nullable = true)
|-- Online Security: string (nullable = true)
|-- Online Backup: string (nullable = true)
|-- Device Protection: string (nullable = true)
|-- Tech Support: string (nullable = true)
|-- Streaming TV: string (nullable = true)
|-- Streaming Movies: string (nullable = true)
|-- Contract: string (nullable = true)
|-- Paperless Billing: string (nullable = true)
|-- Payment Method: string (nullable = true)
```



Equivalencia entre Sql – Spark

Para más funciones de Spark visitar

<http://spark.apache.org/docs/2.0.0/api/python/pyspark.sql.html>

SQL	DataFrame (Python)
<code>SELECT col_1 FROM myTable</code>	<code>df.select(col("col_1"))</code>
<code>DESCRIBE myTable</code>	<code>df.printSchema()</code>
<code>SELECT * FROM myTable WHERE col_1 > 0</code>	<code>df.filter(col("col_1") > 0)</code>
<code>..GROUP BY col_2</code>	<code>..groupBy(col("col_2"))</code>
<code>..ORDER BY col_2</code>	<code>..orderBy(col("col_2"))</code>
<code>..WHERE year(col_3) > 1990</code>	<code>..filter(year(col("col_3")) > 1990)</code>
<code>SELECT * FROM myTable LIMIT 10</code>	<code>df.limit(10)</code>
<code>display(myTable)</code> (text format)	<code>df.show()</code>
<code>display(myTable)</code> (html format)	<code>display(df)</code>

Ejemplos

Según nuestros datos, ¿qué mujeres nacieron después de 1990?

```

1  from pyspark.sql.functions import year
2  display(peopleDF
3      .select("firstName", "middleName", "lastName", "birthDate", "gender")
4      .filter("gender = 'F'")
5      .filter(year("birthDate") > "1990")
6  )

```

¿Cuántas mujeres llamadas Mary nacen cada año?

```

1  marysDF = (peopleDF.select(year("birthDate").alias("birthYear"))
2      .filter("firstName = 'Mary' ")
3      .filter("gender = 'F' ")
4      .orderBy("birthYear")
5      .groupBy("birthYear")
6      .count()
7  )

```



Comparar popularidad de dos nombres desde 1990

```
1 from pyspark.sql.functions import col
2 dordonDF = (peopleDF
3     .select(year("birthDate").alias("birthYear"), "firstName")
4     .filter((col("firstName") == 'Donna') | (col("firstName") == 'Dorothy'))
5     .filter("gender == 'F' ")
6     .filter(year("birthDate") > 1990)
7     .orderBy("birthYear")
8     .groupBy("birthYear", "firstName")
9     .count()
10 )
11 display(dordonDF)
```




Vistas Temporales

En DataFrames, las vistas temporales se utilizan para hacer que el DataFrame esté disponible para SQL y que funcione con la sintaxis SQL sin problemas.

Una vista temporal le da un nombre para consultar desde SQL, pero a diferencia de una tabla, solo existe durante la sesión Spark. Como resultado, la vista temporal no se transferirá cuando reinicie el clúster o cambie a una nueva computadora portátil. Tampoco aparecerá en el botón Datos en el menú en el lado izquierdo de una computadora portátil Databricks que proporciona un fácil acceso a bases de datos y tablas.

La declaración en las siguientes celdas crea una vista temporal que contiene los mismos datos.

Esta vista temporal se creará a partir de un objeto Dataframe.

```
1 womenBornAfter1990DF.createOrReplaceTempView("womenBornAfter1990")
```

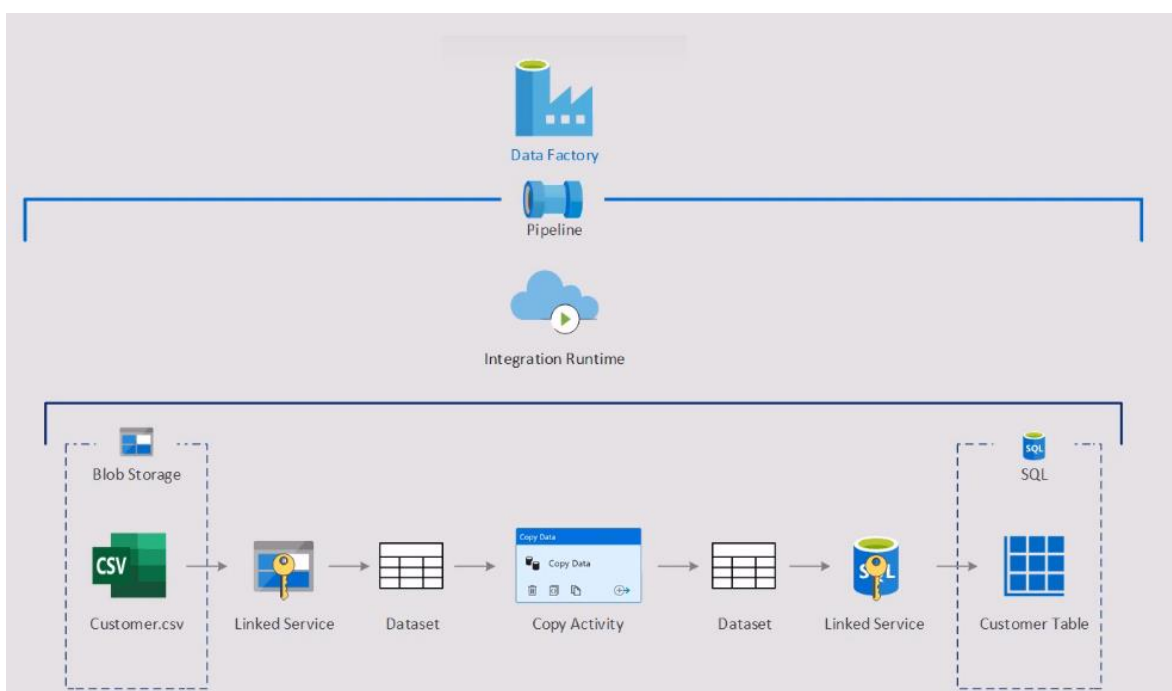
Luego de crearla, se le pueden hacer consultas como si fuera una tabla

```
1 display(spark.sql("SELECT count(*) FROM womenBornAfter1990 where firstName = 'Mary' "))
```

Azure Data Factory [ADF]

Microsoft Azure Data Factory es un servicio de integración de datos basado en la nube que automatiza el movimiento y la transformación de datos. Puede crear, implementar, programar y monitorear rápidamente pipelines de datos de alta disponibilidad y tolerantes a fallas.

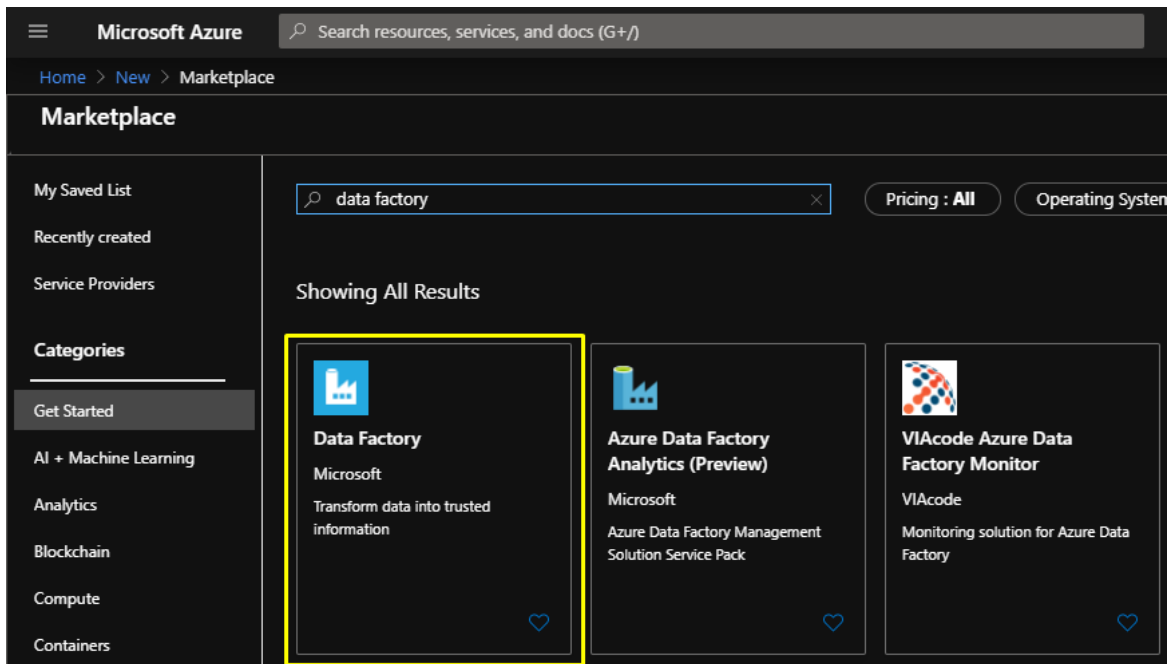
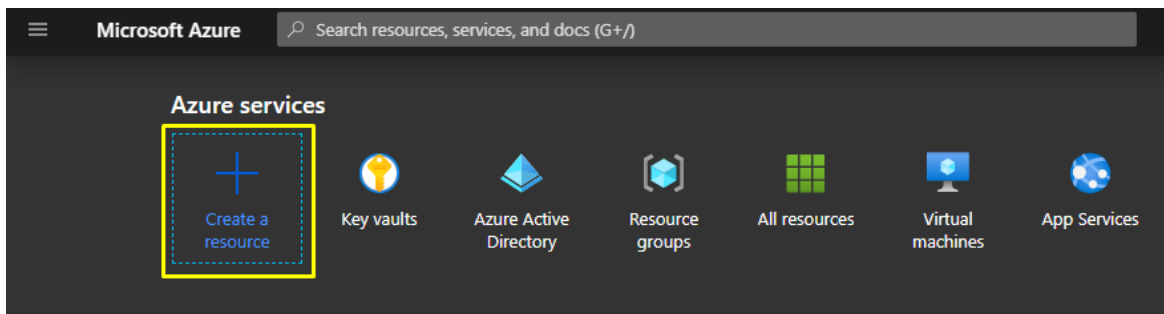
La necesidad de activar el movimiento de datos por lotes o de configurar una programación regular es un requisito para la mayoría de las soluciones de análisis. Azure Data Factory (ADF) es el servicio que se puede usar para cumplir dicho requisito. ADF proporciona un servicio de integración de datos basado en la nube que organiza el movimiento y la transformación de datos entre varios almacenes de datos.





Creación de Azure Data Factory

Para crear un Azure Data Factory (ADF) procederemos según lo indicado en las diapositivas





Microsoft Azure Search resources, services, and docs (G+/)

Home > New > Marketplace > Data Factory > New data factory

New data factory

Name *
Pami-ADF

Version ⓘ
V2

Subscription *
Visual Studio Enterprise – MPN (e8fd21cd-aa84-4d86-ae5a-14bad7b024cb)

Resource Group *
Pami-Test
[Create new](#)

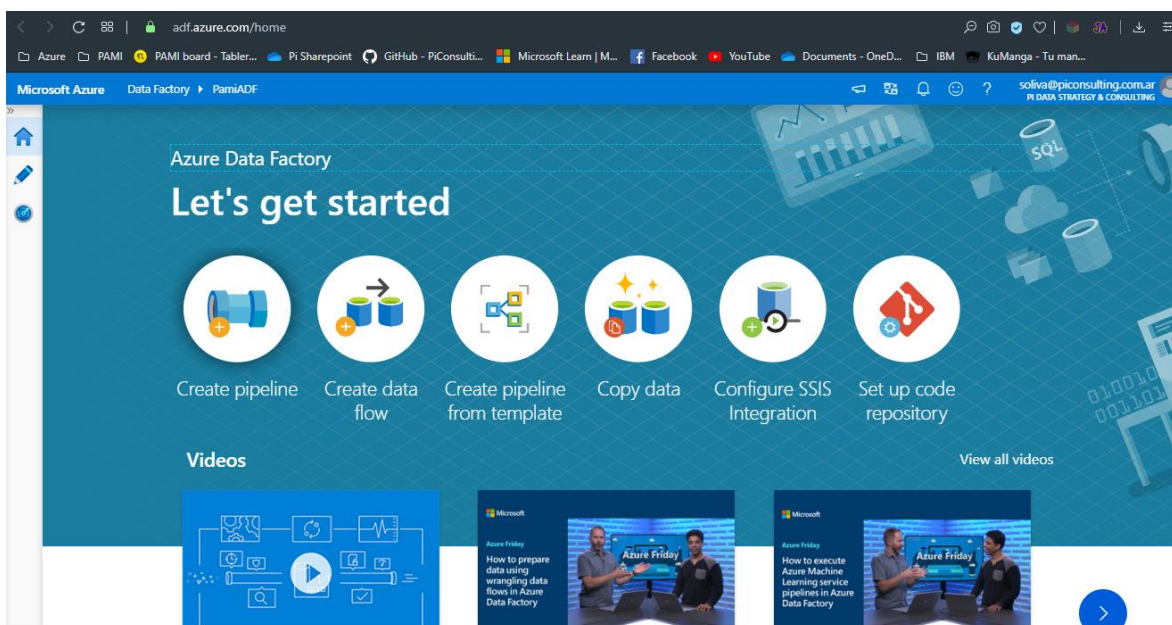
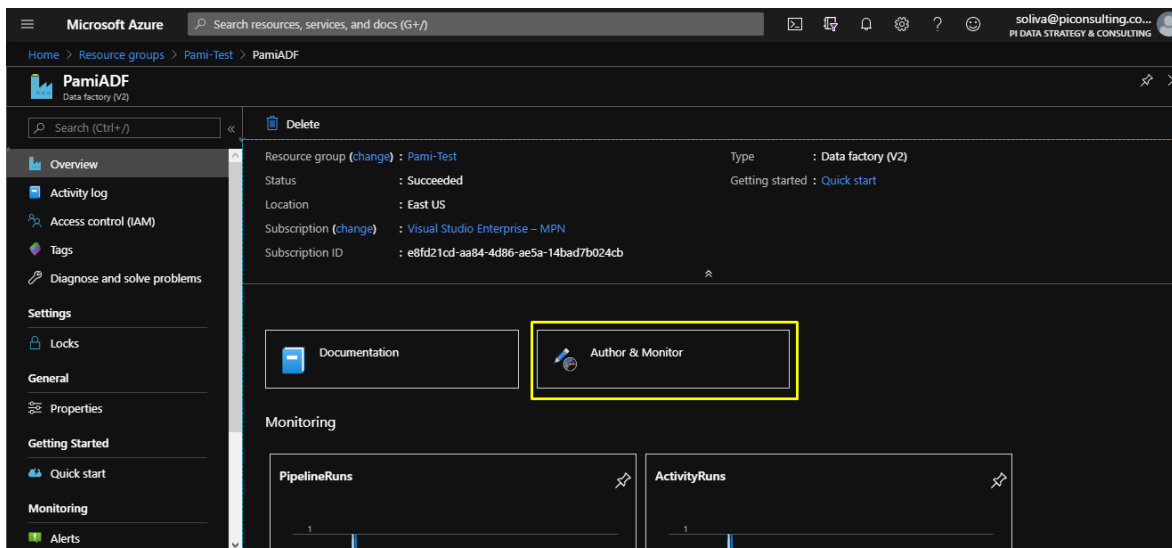
Location * ⓘ
East US

Enable GIT ⓘ
☐

Create



Para gestionar el comportamiento de ADF, entraremos al mismo haremos click en Author & Monitor, se abrirá una nueva ventana



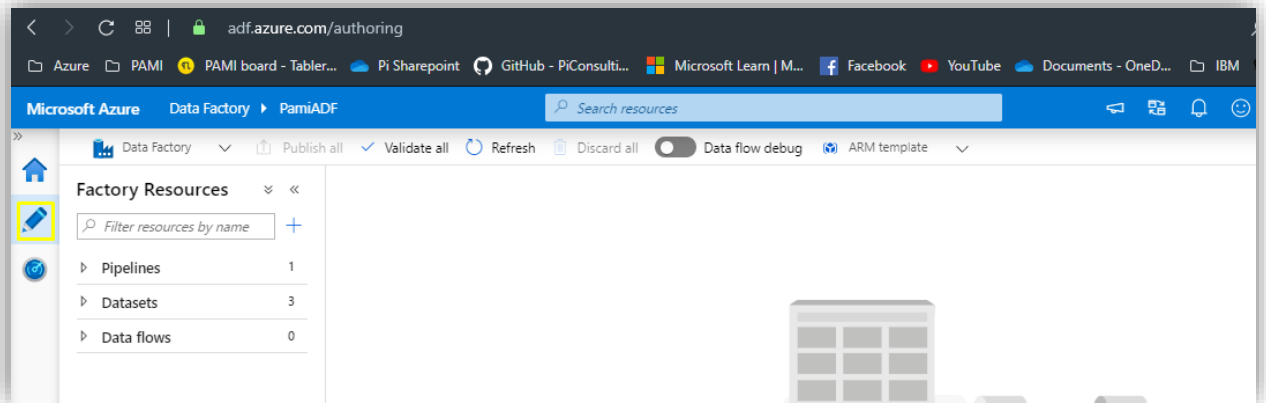


Haciendo click en el Lapiz entraremos al modo edición de este, donde tendremos acceso a los Pipelines, Datasets y DataFlows.

Los **DataFlows** son utilizados cuando se mueven datos y estos se modifican en el proceso, si estos no son modificados, solo se utilizarán actividades, que están en los Pipelines

Pipelines son organizadores de actividades.

Estas actividades usan **Datasets**, que son abstracciones de conjuntos de datos





Integration Runtime

Integration Runtime (IR) es la infraestructura informática utilizada por Azure Data Factory para proporcionar las siguientes capacidades de integración de datos en diferentes entornos de red:

- **Flujo de datos:** ejecute un [Dataflow](#) en un entorno informático administrado de Azure.
- **Movimiento de datos:** copie datos a través de almacenes de datos en redes públicas y almacenes de datos en redes privadas (red privada local o virtual). Proporciona soporte para conectores integrados, conversión de formato, mapeo de columnas y transferencia de datos escalable y de rendimiento.
- Despacho de **actividades:** envíe y supervise las actividades de transformación que se ejecutan en una variedad de servicios informáticos como Azure Databricks, Azure HDInsight, Azure Machine Learning, Azure SQL Database, SQL Server y más.
- **Ejecución del paquete SSIS:** ejecute de forma nativa los paquetes de SQL Server Integration Services (SSIS) en un entorno informático administrado de Azure.

Tipos de Integration Runtime

Data Factory ofrece tres tipos de tiempo de ejecución de integración, y debe elegir el tipo que mejor sirva a las capacidades de integración de datos y las necesidades del entorno de red que está buscando. Estos tres tipos son:

- Azur
- Self-hosted
- Azure-SSIS

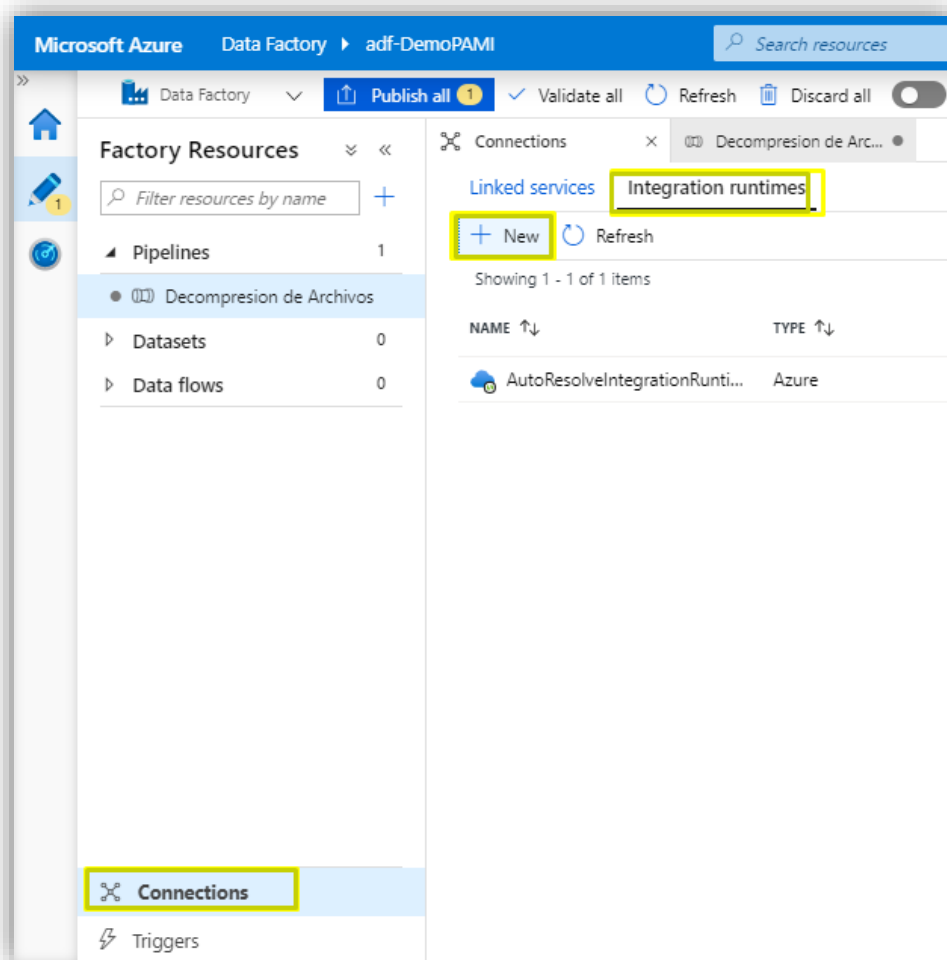
La siguiente tabla describe las capacidades y el soporte de red para cada uno de los tipos de tiempo de ejecución de integración:

Tipo de IR	Red pública	Red privada
Azure	Flujo de datos Movimiento de datos Ejecución de Actividad	
Self-Hosted	Movimiento de datos Ejecución de Actividad	Movimiento de datos Ejecución de Actividad
Azure-SSIS	Ejecución del paquete SSIS	Ejecución del paquete SSIS

Creación de Integration Runtime

Para crear un Integration Runtime, es necesario seguir los siguientes pasos:


- 1- Hacer click en Connections – Integration Runtimes – New
- 2- Seleccionar **Azure, Self-Hosted**
- 3- Seleccionar Azure - Continue






Integration runtime setup

Integration Runtime is the native compute used to execute or [dispatch](#) activities. Choose what integration runtime to create based on required capabilities. [Learn more](#)


**Azure, Self-Hosted**
Perform data flows, data movement and dispatch activities to external compute.


**Azure-SSIS**
Lift-and-shift existing SSIS packages to execute in Azure.

[Continue](#) [Cancel](#)


Integration runtime setup

Network environment:
Choose the network environment of the data source / destination or external compute to which the integration runtime will connect to for data flows, data movement or dispatch activities:

**Azure**
Choose this if you are accessing services with public accessible endpoints.

**Self-Hosted**
Use this for running activities in an on-premise / private network
✓

External Resources:
You can use an existing self-hosted integration runtime that exists in another resource. This way you can reuse your existing infrastructure where self-hosted integration runtime is setup.

**Linked Self-Hosted**
[Learn more](#)

[Continue](#) [Back](#) [Cancel](#)



Luego gestionaremos su Setup

Seleccionamos la Región East US, y en cómputo y cores, seleccionamos General Purpose y 4, ya que estamos trabajando en una prueba de concepto, en la cual no necesitaremos más recursos.

Finalmente hacemos click en Create.

Integration runtime setup

The Data Factory manages the integration runtime in Azure to connect to required data source/destination or external compute in public network. The compute resource is elastic allocated based on performance requirement of activities.

Name *
IntegrationRuntime-Pami

Description
Enter description here...

Type
Azure

Region *
East US

Data flow run time

Compute type *
General Purpose

Core count *
4 (+ 4 Driver cores)

Time to live
0 minutes

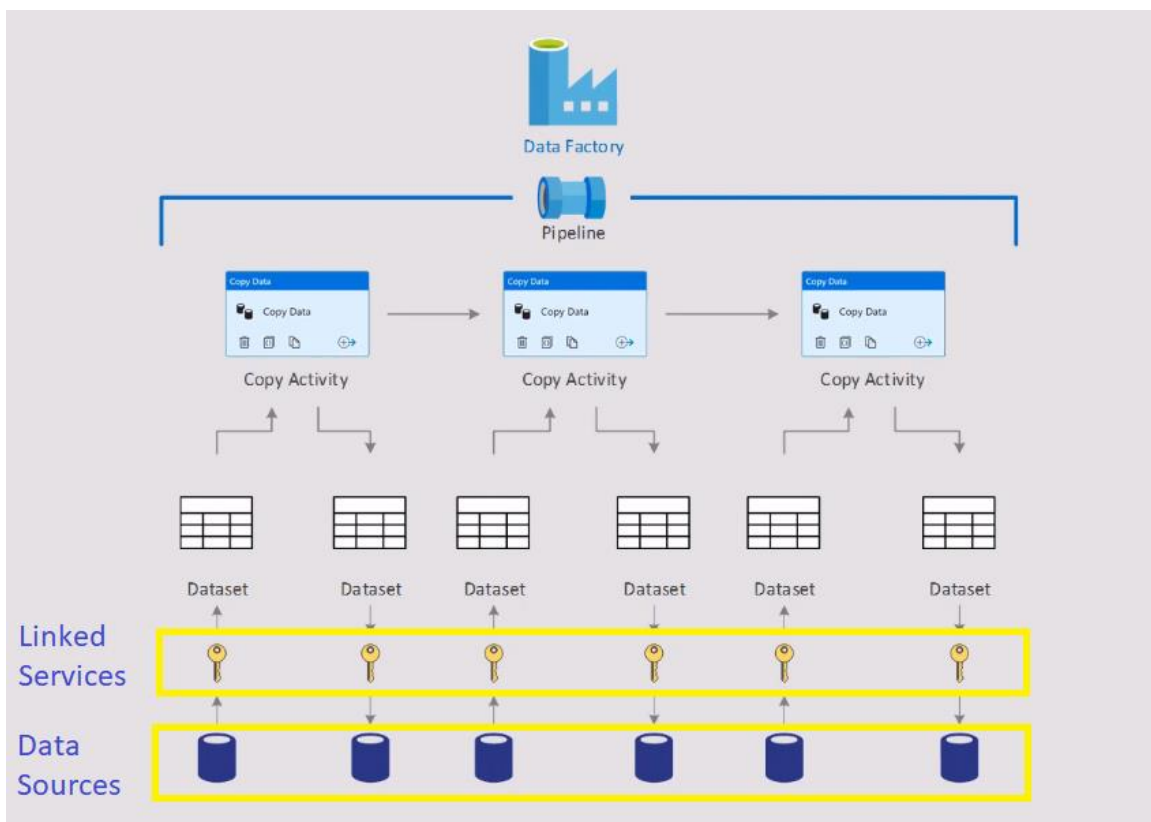
Billing for data flows is based upon the type of compute you select and the number of cores selected per hour. If you set a TTL then the minimum billing time will be that amount of time.

Create **Back** **Cancel**

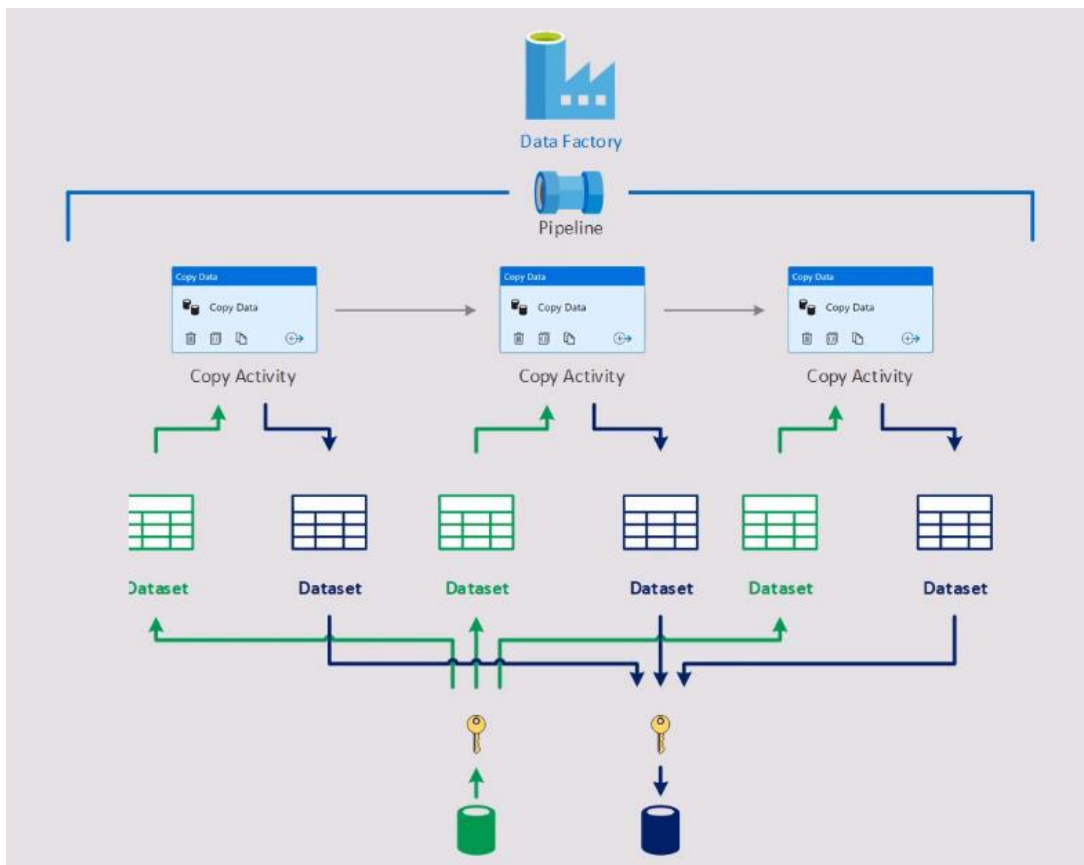


Linked Services [LS]

Con Linked Services gestionamos los servicios con los cuales nos conectaremos para utilizar su contenido.



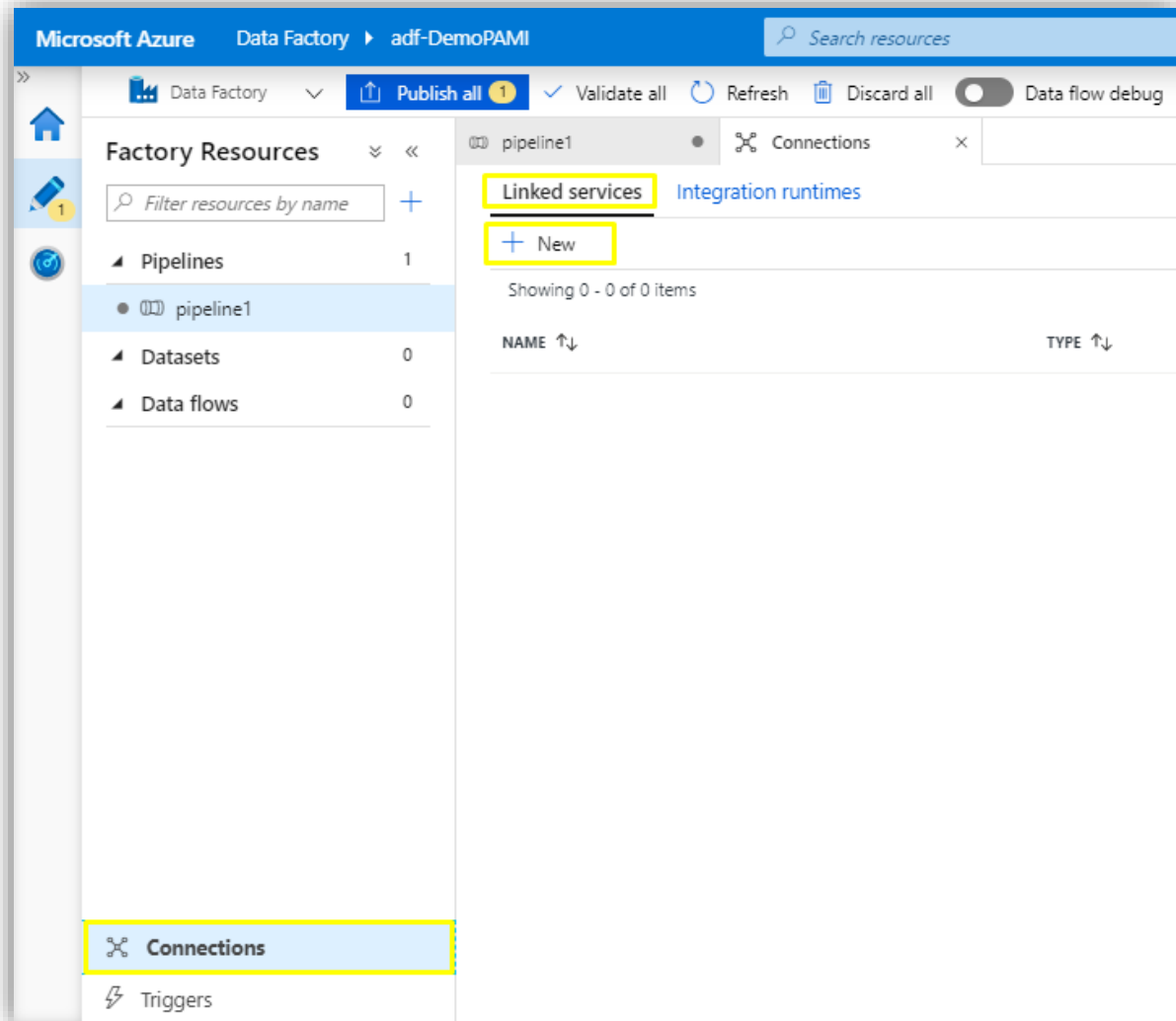
Si estas fuentes de datos se repiten es posible usar el mismo LS



*Imágenes de Azure4Everyone

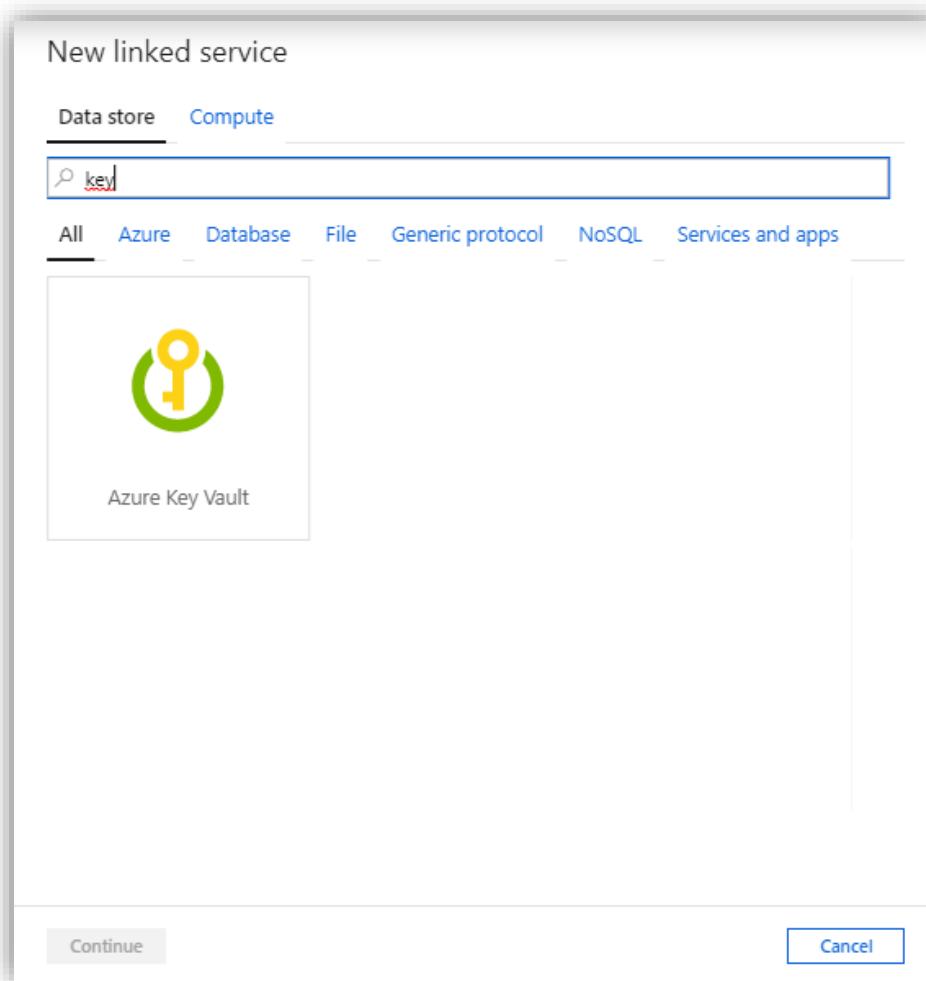


Para crear un Linked Service, hay que hacer click en Connections – Linked Services - New



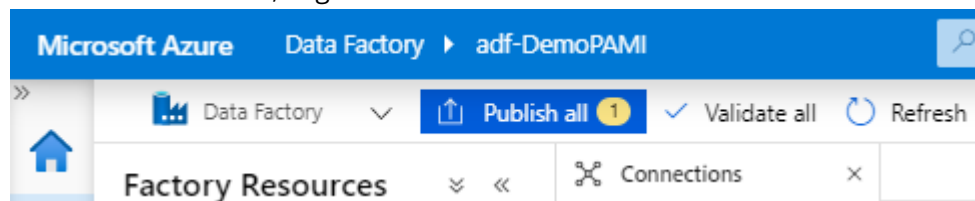


Nuestro primer Linked Service será a nuestra Key Vault



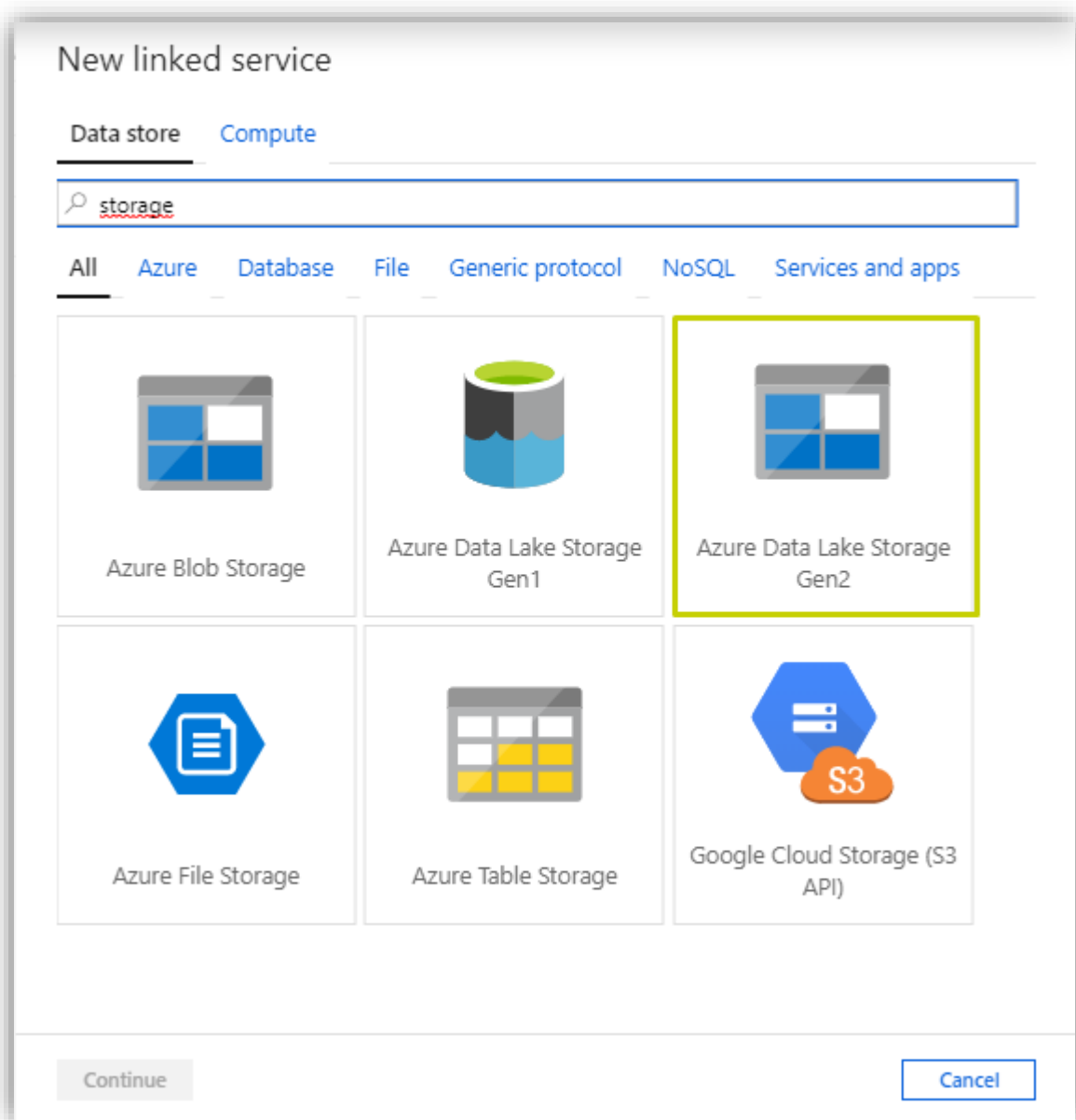
Aquí cargaremos los datos de nuestro Key Vault, primero le daremos un nombre a nuestro LS, seleccionaremos desde cual suscripcion accederemos, y el nombre de nuestro KV. Finalmente haremos click en Create.

Ya creado nuestro LS, lo guardaremos haciendo click en **Publish all**





Con nuestro KV ya enlazado, procederemos a enlazar nuestra fuente de datos.



Aquí le daremos nombre a nuestro LS, seleccionaremos cual Runtime lo va a conectar, elegiremos el metodo de autenticación por SP, seleccionaremos nuestra suscripción, el nombre del Storage Account

New linked service (Azure Data Lake Storage Gen2)

i 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 *

Pamilake

Description

Connect via integration runtime *

IntegrationRuntime-Pami

[Edit integration runtime](#)

Authentication method

Service Principal

Account selection method

☒ From Azure subscription ☐ Enter manually

Azure subscription

Patrocinio de Microsoft Azure (08f96a01-421d-447c-ae35-e96908bc704d)

Storage account name *

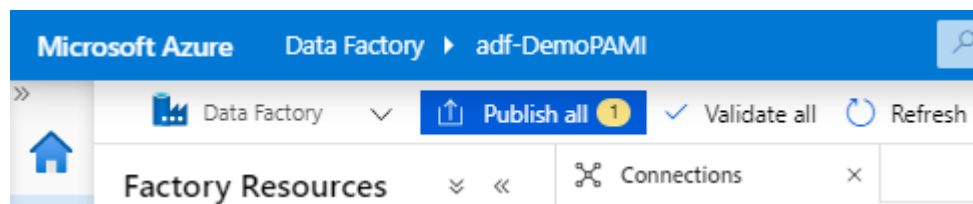
pamilakegen2

Continuando en esta pantalla, seleccionaremos nuestro Tennant, el ID del SP, que se encuentra en Azure Active directory, seleccionaremos Azure Key Vault para acceder al SP, colocando:

- El nombre del LS para acceder a nuestro KV
- El nombre del secreto donde almacenamos el acceso al SP

Finalmente haremos click en Create

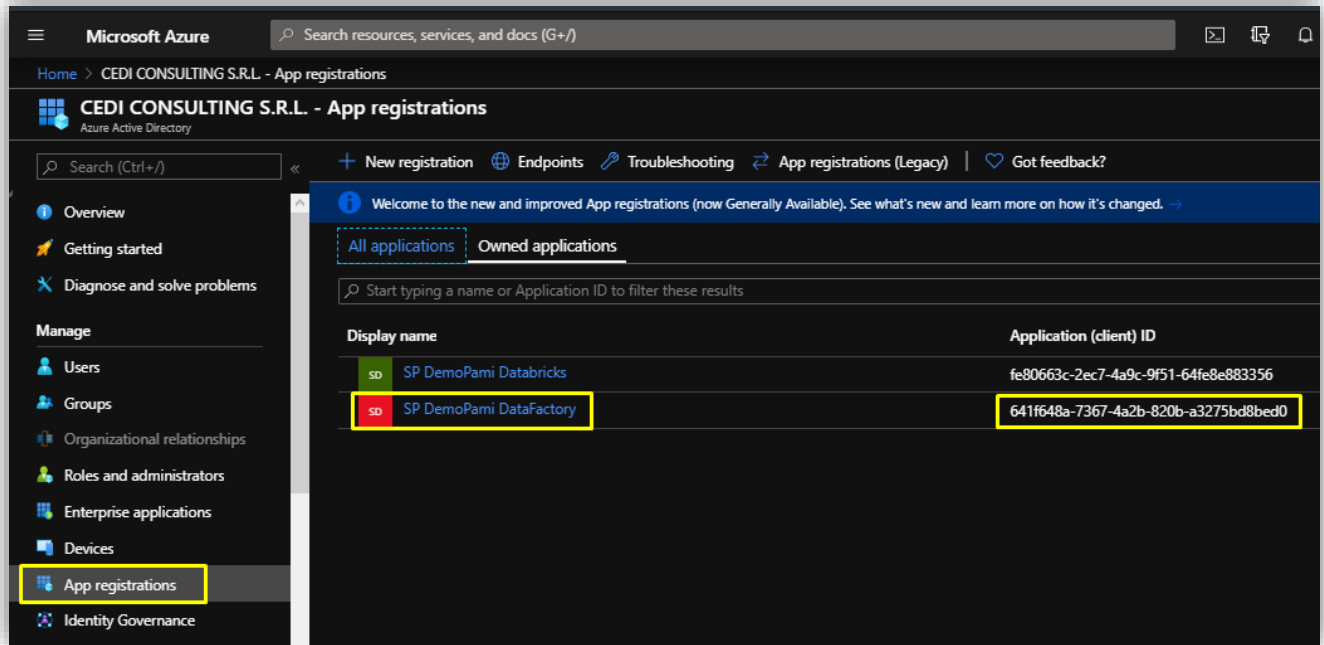
Ya creado nuestro LS, lo guardaremos haciendo click en **Publish all**



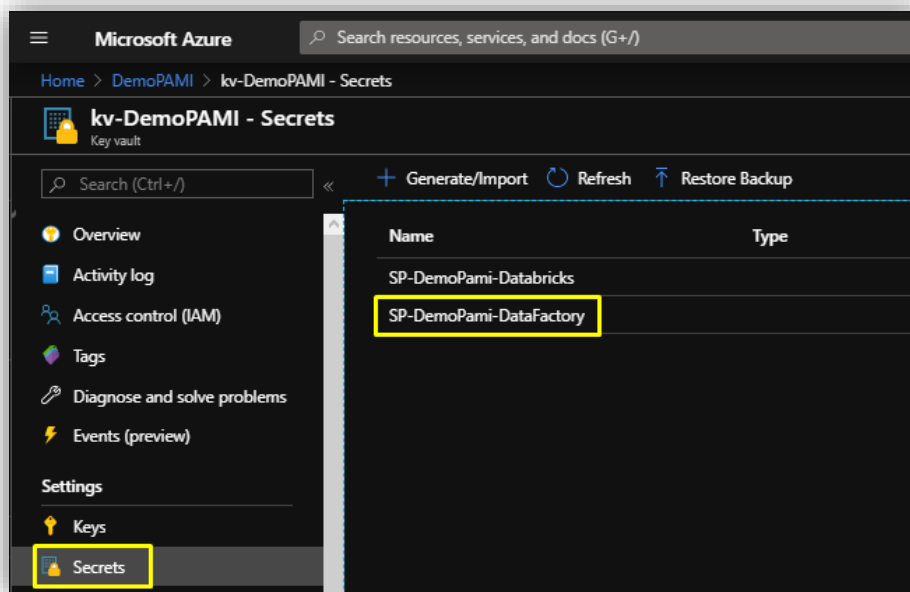


Para facilitar la búsqueda, coloco a continuación donde encontrar algunos recursos

Service Principal ID – Dentro de Azure Active Directory



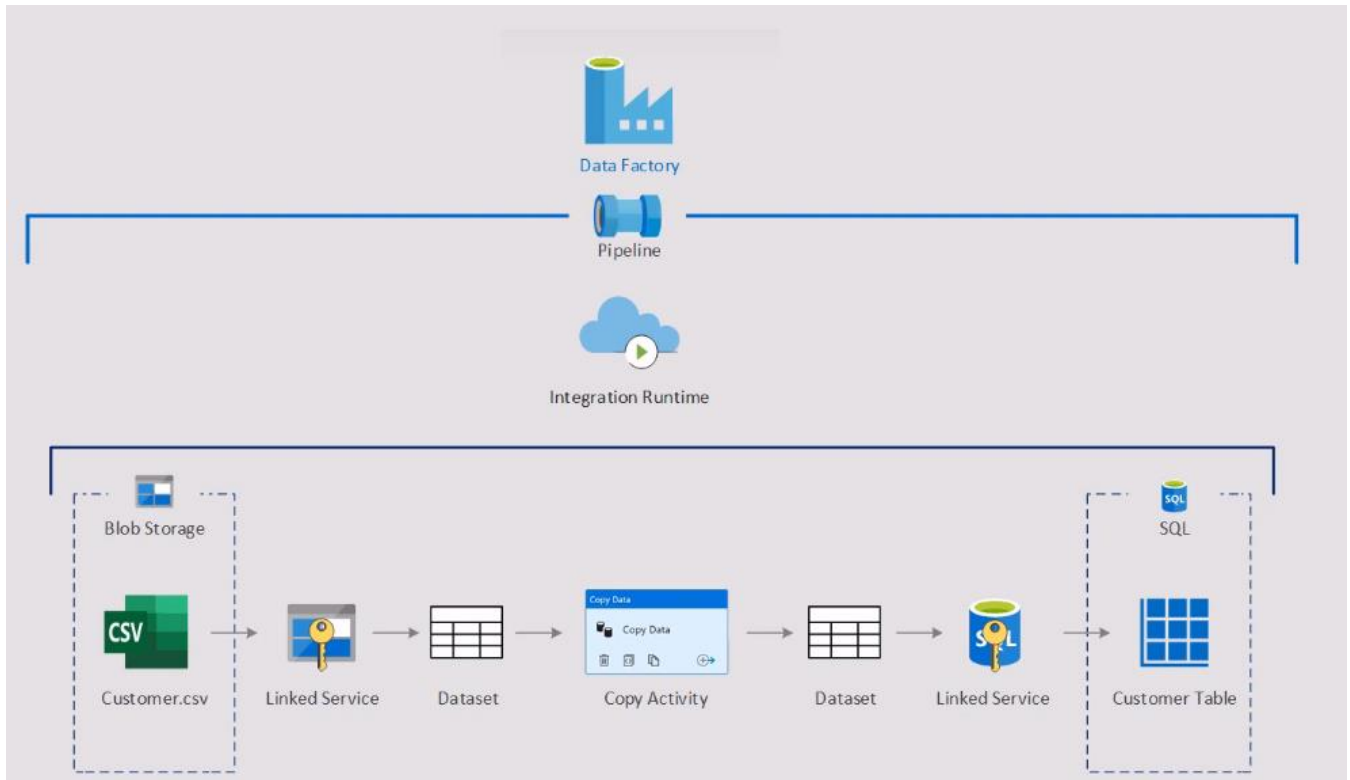
Nombre del secreto – Dentro de nuestro KV



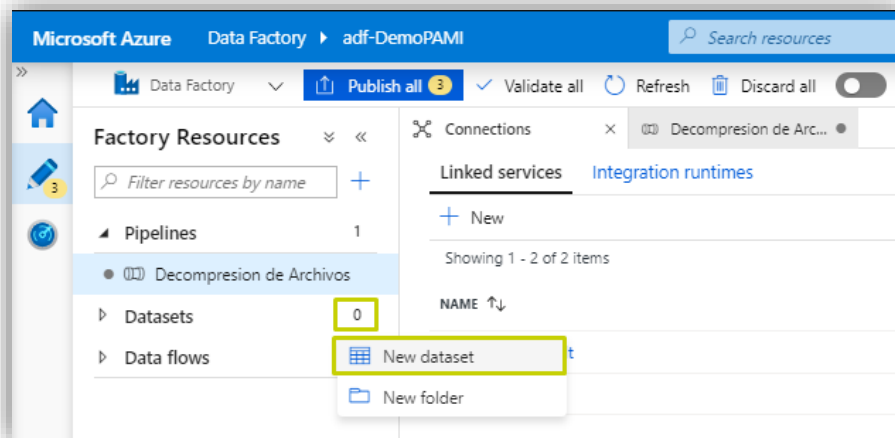


Gestión de Datasets

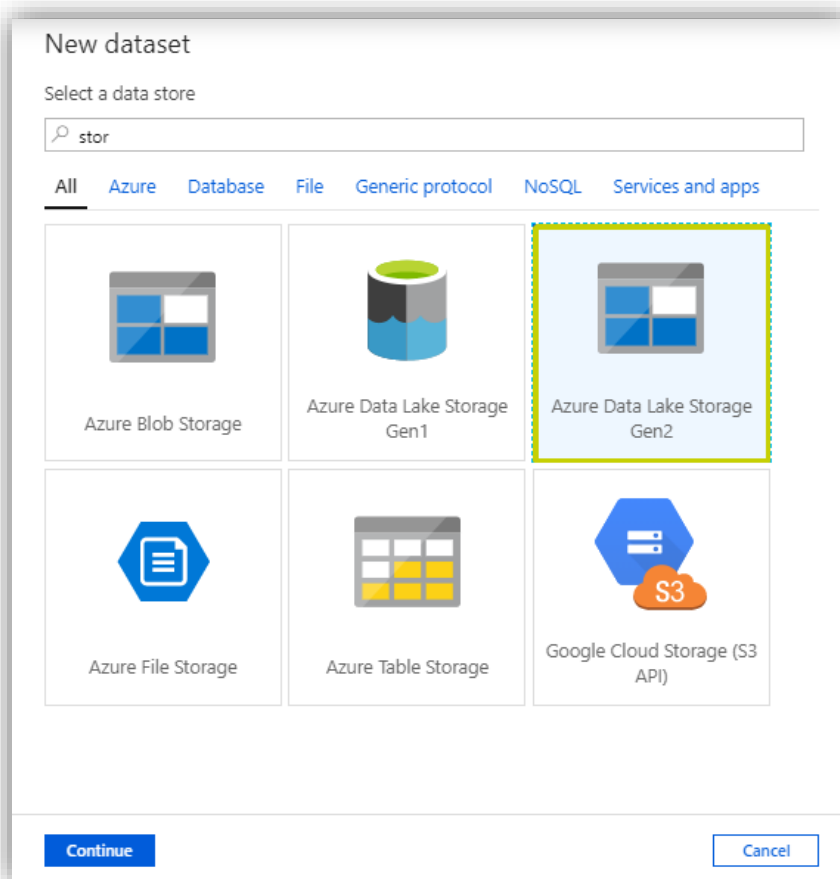
Para trabajar en Datafactory, es necesario importar los Datasets con los que trabajaremos. Es necesario hacerlo tanto el de fuente como con el de destino.



Para crear un nuevo Dataset, dentro de la pestaña de edición, click en la pestaña – New dataset



Aquí seleccionaremos de donde se consumirá, en nuestro caso seleccionamos Azure Data Lake Storage Gen2 y continuar











Aquí seleccionaremos el formato en el cual estará nuestro Dataset y haremos click en Continue

Select format

Choose the format type of your data

 Parquet	 DelimitedText	 Json
 Avro	 ORC	 Binary

Continue Back Cancel



Aquí pondremos el nombre de nuestro dataset, el LS con el cual accederemos, el path hasta el archivo, definimos si la primera fila representa el encabezado y especificaremos de donde tomara la Metadata, en nuestro caso, haremos que lo tome de la misma conexión.

Set properties

Name
Test_churn_input

Linked service *
Pamilake

[Edit connection](#)

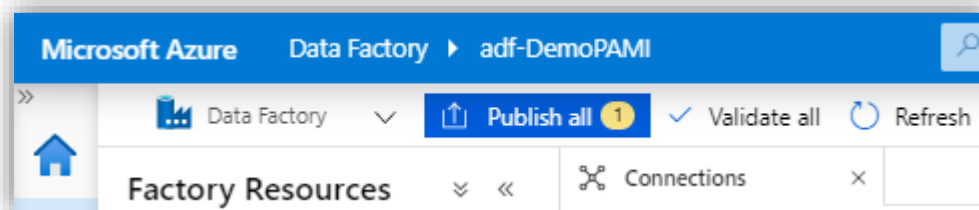
File path
pruebas / Test-Origin / Customer Churn.csv [Browse](#)

First row as header ☒

Import schema
☒ From connection/store ☐ From sample file ☐ None

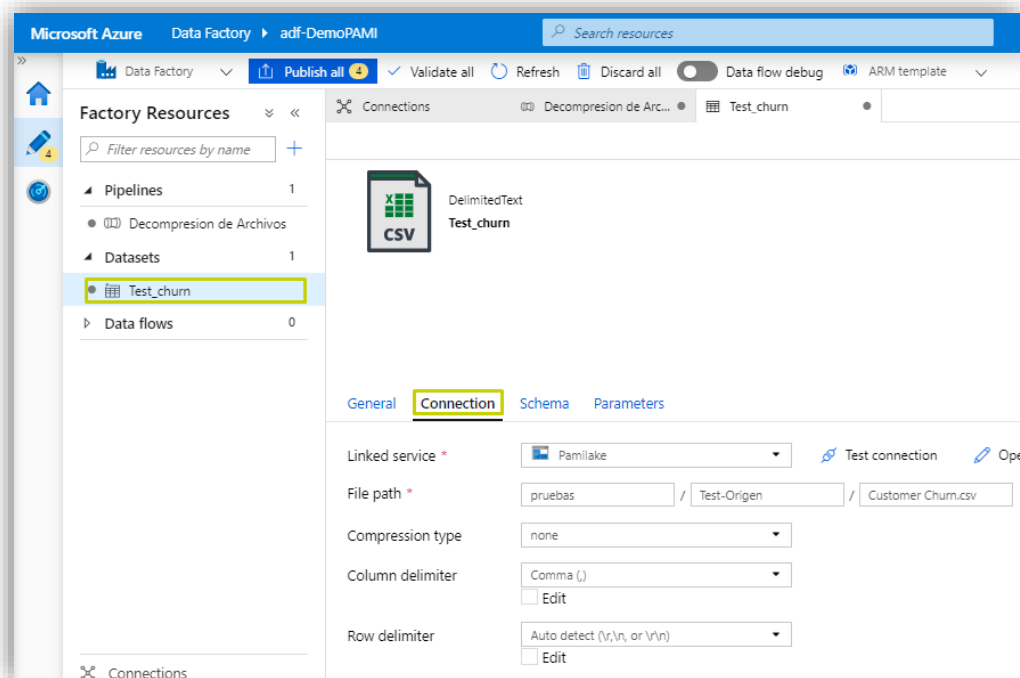
[OK](#) [Back](#) [Cancel](#)

Guardaremos nuestro Dataset haciendo click en **Publish all**

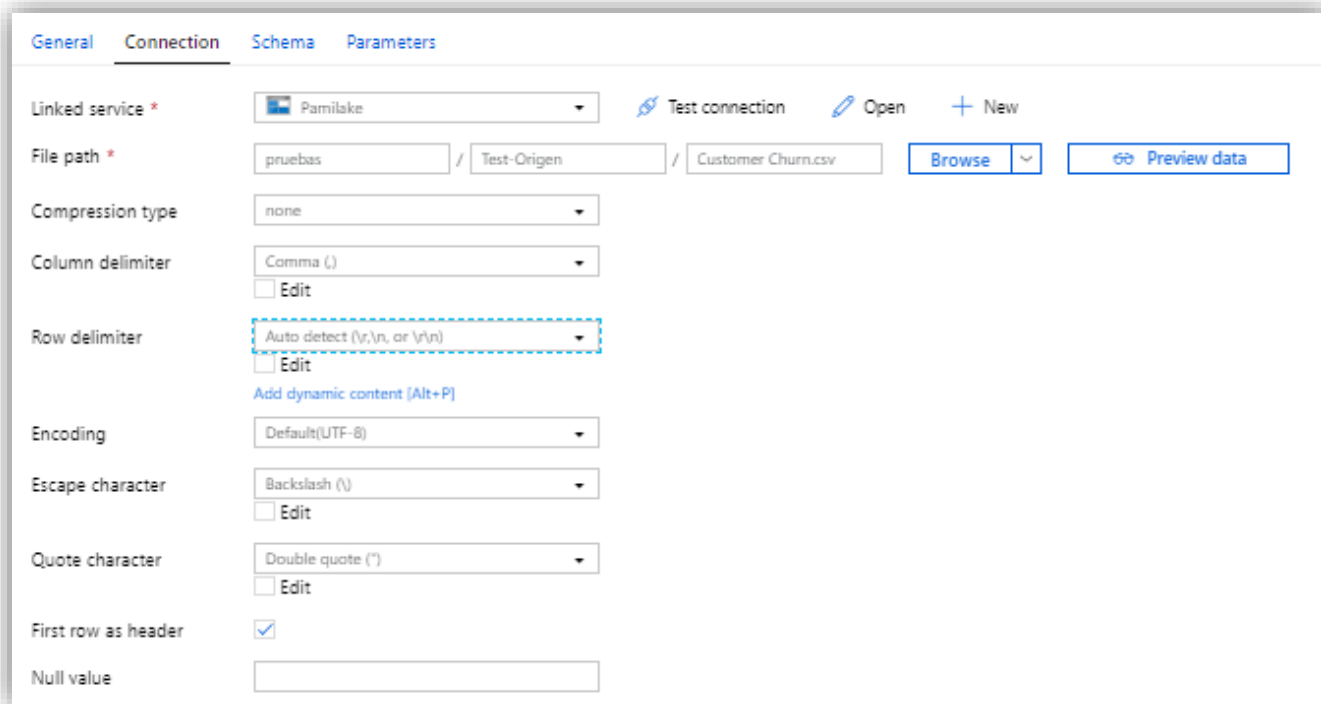




Una vez creado el dataset, se puede hacer click en el mismo – Connection, para gestionar detalles de este



Aquí podremos modificar propiedades como el delimitador de columnas, el tipo de compresión, el delimitador de filas, el path hacia donde se encuentra el archivo, codificación, etc.



También podremos utilizar Preview Data para ver como reconoce la plataforma a nuestro dataset.



Preview data

Linked service: Pamilake
Object: Customer Churn.csv

LoyaltyID	Customer ID	Senior Citizen	Partner	Dependents	Tenure	Phone Service	Multiple Lines	Internet Service	Online Security	Online Backup	Device Protection
318537	7590-VHVEG	No	Yes	No	1	No	No phone service	DSL	No	Yes	No
152148	5575-GNVDE	No	No	No	34	Yes	No	DSL	Yes	No	Yes
326527	3668-QPYBK	No	No	No	2	Yes	No	DSL	Yes	Yes	No
845894	7795-CFOCW	No	No	No	45	No	No phone service	DSL	Yes	No	Yes
503388	9237-HQITU	No	No	No	2	Yes	No	Fiber optic	No	No	No
160192	9305-CDSKC	No	No	No	8	Yes	Yes	Fiber optic	No	No	Yes
608533	1452-	No	No	Yes	33	Yes	Yes	Fiber	No	Yes	No

Por otro lado, hay que gestionar el destino de nuestra tabla, aquí dejaremos el dónde se cargara, y en caso de ser un archivo, el path y formato de este.

El método es similar a crear el archivo fuente, igualmente mostraremos todos los pasos para facilitar comprensión.

En nuestro caso crearemos un csv.



New dataset

Select a data store

🔍 stora

All Azure Database File Generic protocol NoSQL Services and apps



Azure Blob Storage



Azure Data Lake Storage
Gen1



Azure Data Lake Storage
Gen2



Azure File Storage



Azure Table Storage



Google Cloud Storage (S3
API)

Select format

Choose the format type of your data



Parquet



DelimitedText



Json



Avro



ORC



Binary



Set properties

Name
Test_churn_output

Linked service *
Pamilaake

[Edit connection](#)

File path
pruebas / Test-Destino / File [Browse](#)

First row as header ☒

Import schema
☒ From connection/store ☐ From sample file ☐ None

[OK](#) [Back](#) [Cancel](#)

En nuestro caso, dejaremos el path hacia una carpeta, por lo tanto, al realizar la carga de datos, se cargará un archivo con el mismo nombre que el archivo fuente.

Set properties

Name
Test_churn_output

Linked service *
Pamilaake

[Edit connection](#)

File path
pruebas / Test-Destino / File [Browse](#)

First row as header ☒

Import schema
☒ From connection/store
 ☐ From sample file
 ☐ None

OK Back Cancel

Guardaremos nuestro Dataset haciendo click en **Publish all**

Microsoft Azure Data Factory ► adf-DemoPAMI

» Data Factory ▼ **Publish all** 1 ✓ Validate all ↺ Refresh

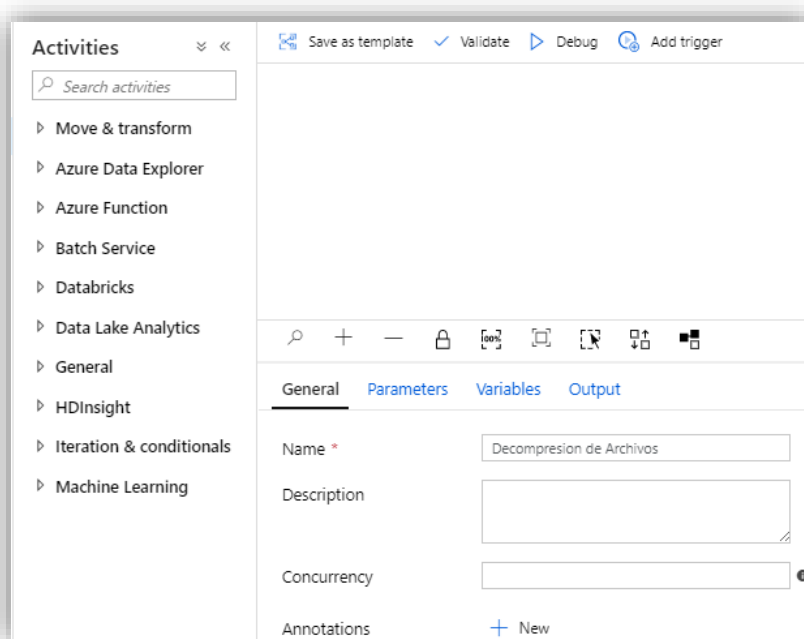
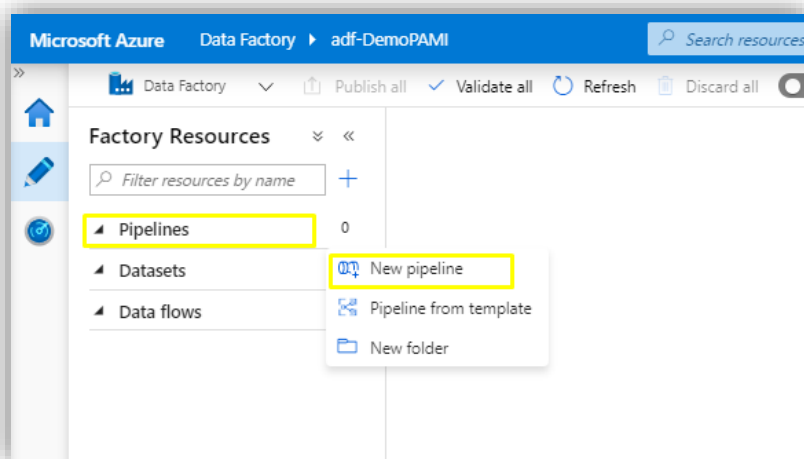
Home Factory Resources ≡ << Connections ×

Creación de un Pipeline

Con Azure Data Factory, puede crear y programar flujos de trabajo basados en datos (llamados Pipelines) que pueden ingerir datos de varios almacenes de datos.

Para crear un nuevo Pipeline, se proceden los siguientes pasos:

- 1- Click en Pipelines – New Pipeline
- 2- Gestionamos las características de nuestro pipeline
 - a. Nombre
 - b. Descripción
 - c. Cantidad Simultanea de Pipelines ejecutados



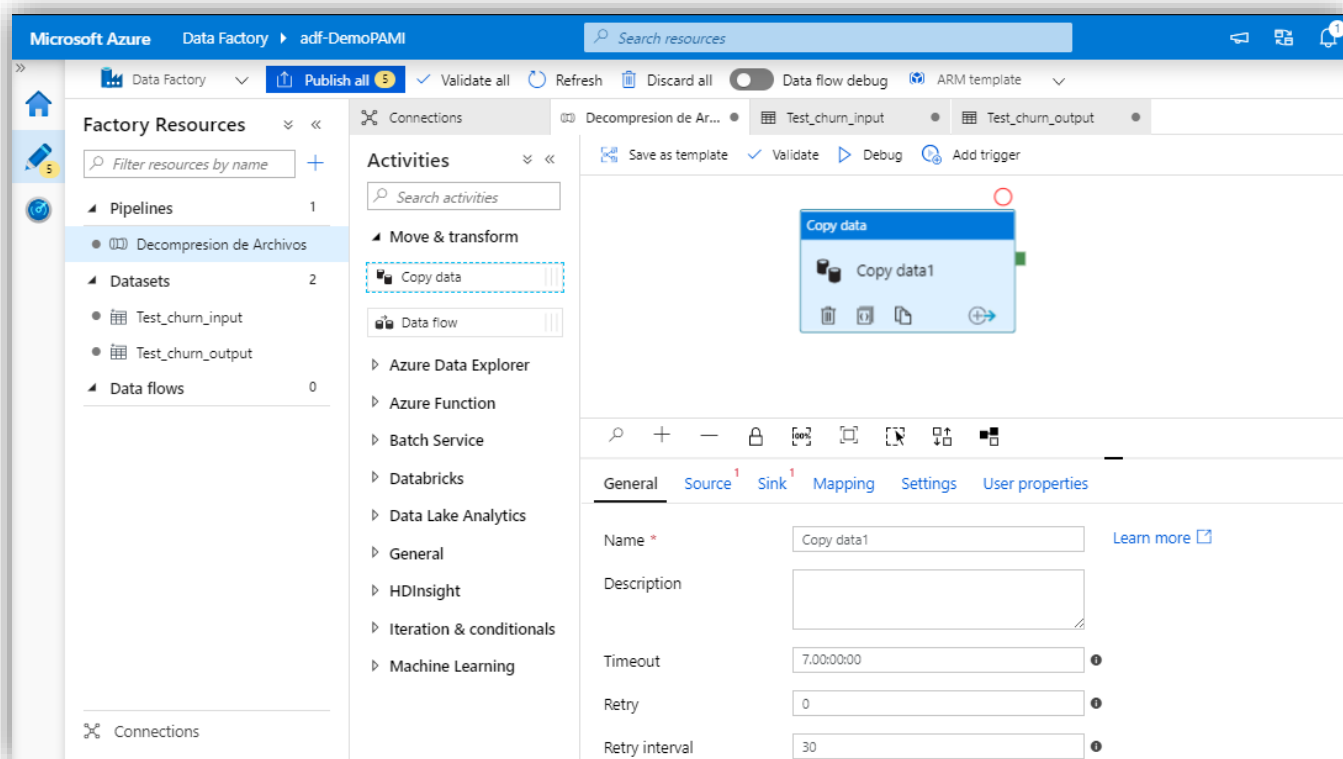


Gestión de Actividades

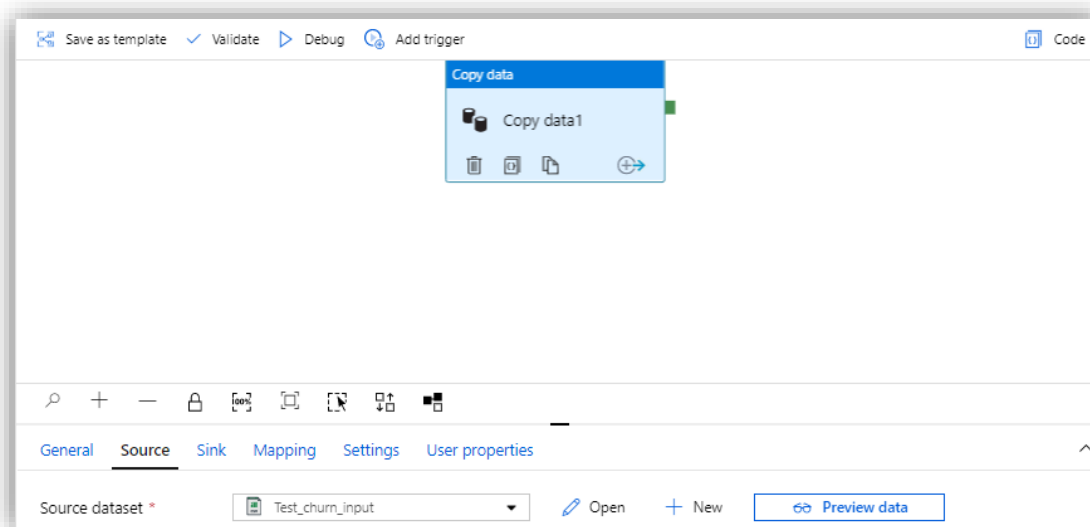
Actividades son todas las tareas que se pueden realizar dentro del Pipeline.

Dentro de cada Pipeline, en la columna correspondiente a Activities, hay un listado de todas las tareas posibles.

Como ejemplo, nosotros crearemos una Actividad llamada Copy Data. Con esta Actividad, nosotros podemos copiar un dataset desde una fuente hacia un destino determinado.



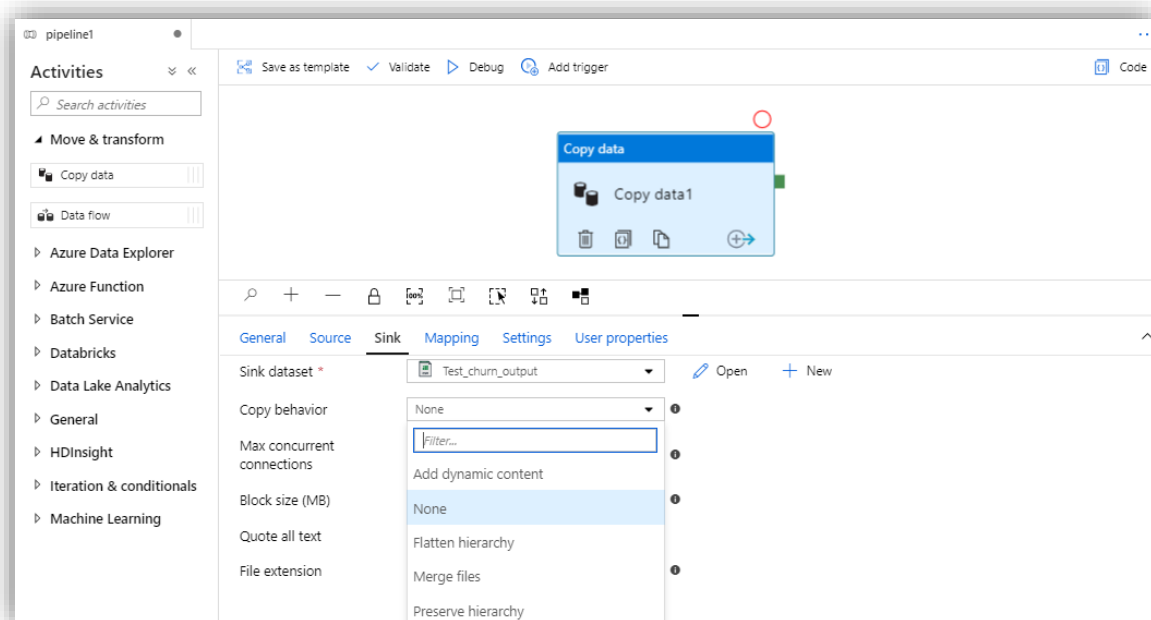
Dentro de esta actividad veremos la pestaña Source, en esta seleccionaremos el Dataset fuente, entre los que ya se han determinado,



Luego seleccionaremos en Sink donde copiaremos nuestra información.

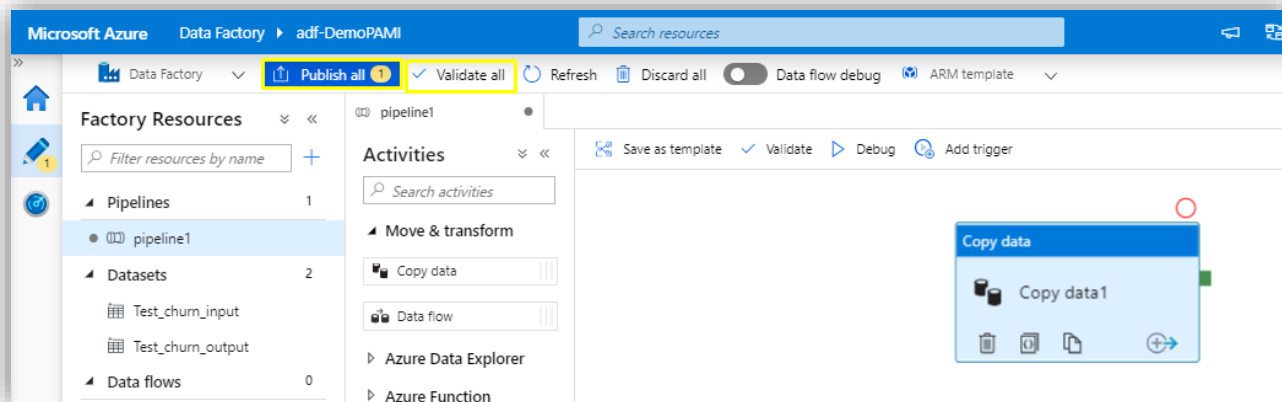
En Copy behavior definiremos el comportamiento de la copia, ya que es de un storage a otro, la jerarquía dentro del File system, y si se pasan varios archivos de manera simultánea, que esos archivos se unan en uno solo.

También es necesario especificar la extensión del archivo.



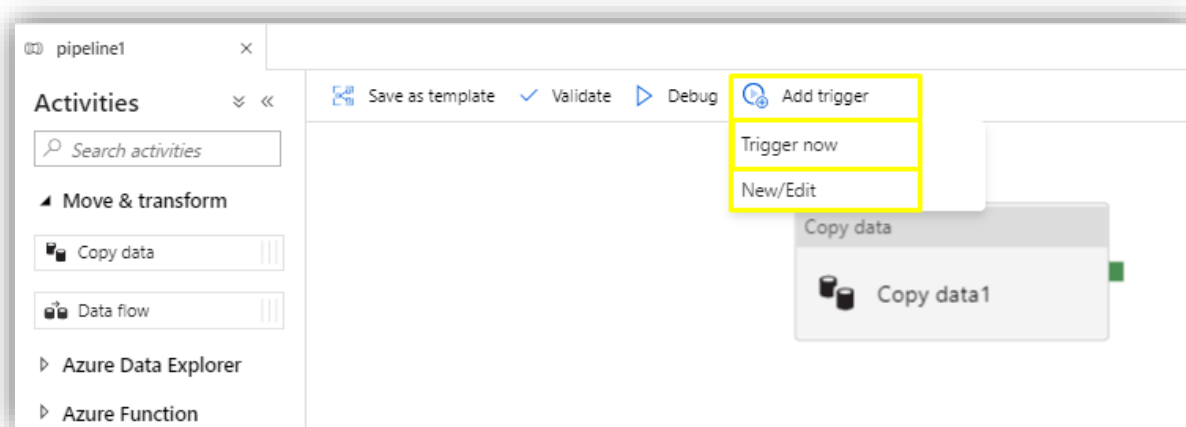


Para guardar el Pipeline, haremos click en validar y luego en Publicar



Gestión de Ejecución - Triggers

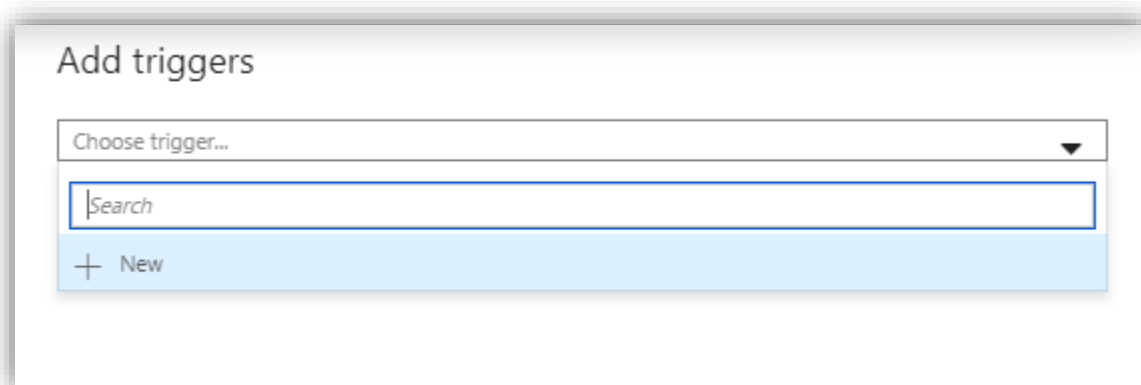
Para que se ejecute el pipeline, es necesario la utilización de Triggers. Crearemos uno haciendo click en Add trigger.



Aquí nos aparecerán las opciones:

- Trigger now, que ejecutara el Pipeline en el momento,
- New/Edit que permitirá la gestión de ejecuciones ante la presencia de ciertos eventos.

Creando un nuevo Trigger nos aparecerá la siguiente ventana



Clickeando en New, crearemos un nuevo Trigger, del cual tendremos que definir el tipo.



Schedule

El primero con el cual trabajaremos será Schedule, que gestiona la ejecución según la fecha.

En el ejemplo que muestro a continuación, el pipeline se ejecutará todos los lunes a las 4AM

Edit trigger

Name *
Lunes madrugada

Description

Type *
ScheduleTrigger

Start Date (UTC) *
03/02/2020 4:00 AM

Recurrence *
Every 1 Week(s)

Advanced recurrence options

Run on these days

Sun	Mon	Tue	Wed	Thu	Fri	Sat
-----	-----	-----	-----	-----	-----	-----

Execute at these times

Hours (UTC) 4

Minutes (UTC) 0

Schedule execution times (UTC)

Dentro de la configuración de este, también podremos definir si esta programación se detendrá en algún momento, y si está activo.

Hours (UTC) 4

Minutes (UTC) 0

Schedule execution times (UTC)
04:00

End *
☒ No End ☐ On Date

Annotations
+ New

Activated *
☒ Yes ☐ No

OK Cancel

Tumbling window

Este será el tipo de Trigger utilizado para ejecutar el Pipeline cada cierto tiempo a partir de un momento determinado. Como ejemplo, les comparto la siguiente imagen.

New trigger

Name *
Cada una hora

Description

Type *
☐ Schedule ☒ Tumbling window ☐ Event

Start Date (UTC) *
02/27/2020 6:18 PM

Recurrence *
Every 1 Hour(s)

End *
☒ No End ☐ On Date

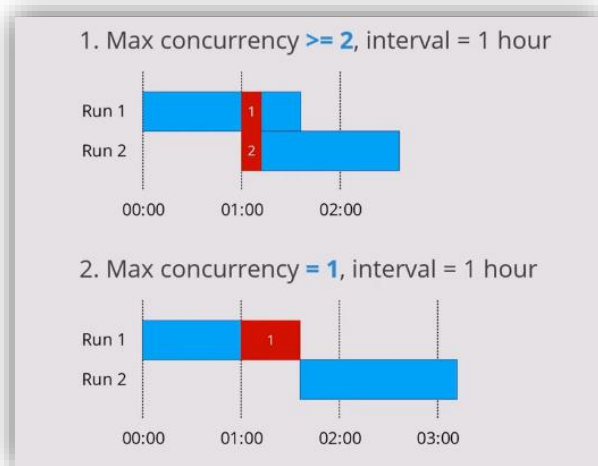
Advanced

Annotations
+ New

Activated *
☐ Yes ☒ No

OK Cancel

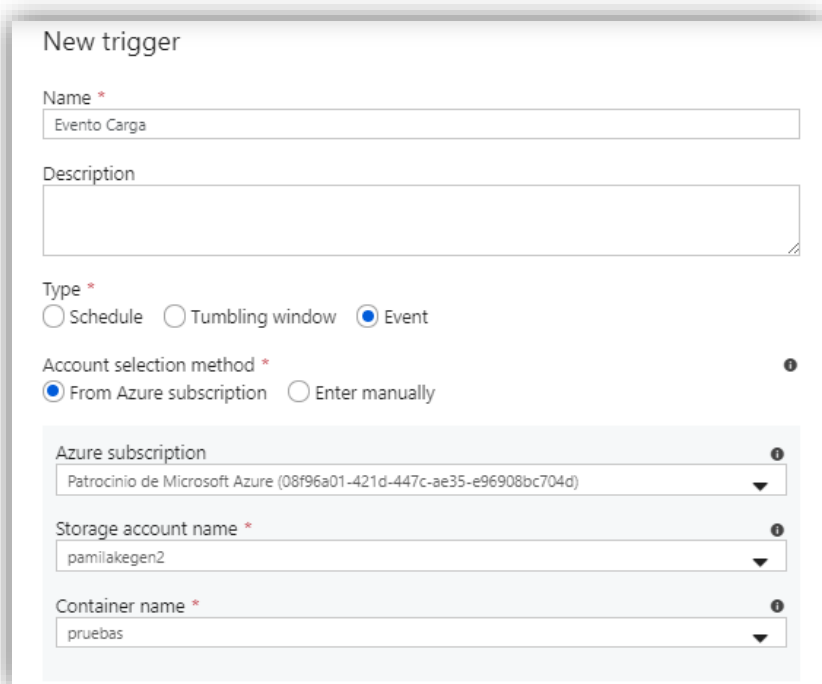
Este Trigger que gestiona la superposición de pipelines por lo cual se podrá gestionar en qué momento comenzará a contar el tiempo



Event

También se puede programar la ejecución del Pipeline cuando ocurre determinado evento.

En las imágenes mostradas a continuación, el evento que determina la ejecución del Pipeline es la creación de un blob dentro del container Pruebas, que a su vez este dentro de la carpeta "Test-Origin" y cuyo formato sea ".csv"



New trigger

Name *
Evento Carga

Description

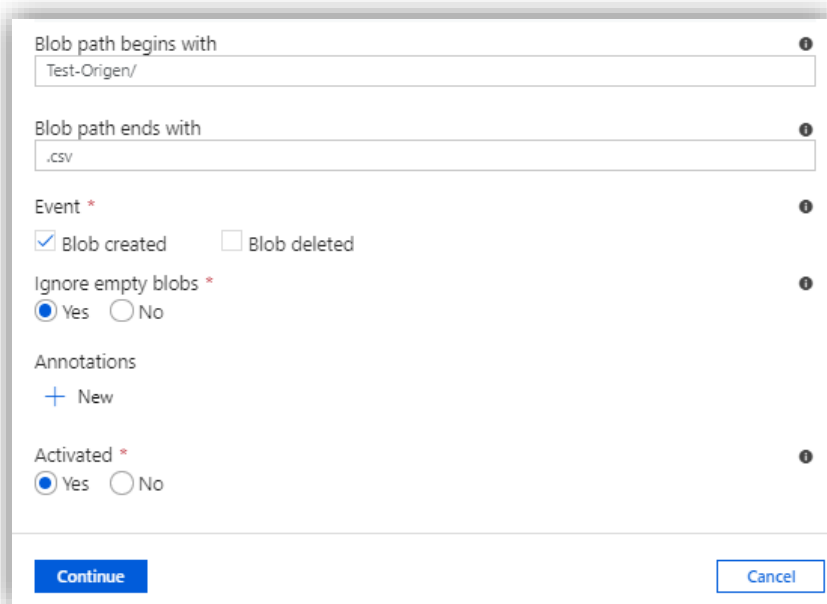
Type *
☐ Schedule
 ☐ Tumbling window
 ☒ Event

Account selection method *
☒ From Azure subscription
 ☐ Enter manually

Azure subscription
 Patrocinio de Microsoft Azure (08f96a01-421d-447c-ae35-e96908bc704d)

Storage account name *
 pamilaakegen2

Container name *
 pruebas



Blob path begins with
 Test-Origin/

Blob path ends with
 .csv

Event *
☒ Blob created
 ☐ Blob deleted

Ignore empty blobs *
☒ Yes
 ☐ No

Annotations
 + New

Activated *
☒ Yes
 ☐ No

Continue Cancel



Autorización del acceso a datos en Azure Storage

Cada vez que accede a datos de la cuenta de almacenamiento, el cliente realiza una solicitud a través de HTTP/HTTPS a Azure Storage. Todas las solicitudes a un recurso seguro deben estar autorizadas para que el servicio garantice que el cliente tiene los permisos necesarios para acceder a los datos.

En la tabla siguiente se describen las opciones que ofrece Azure Storage para autorizar el acceso a los recursos:

	Clave compartida (clave de cuenta de almacenamiento)	Firma de acceso compartido (SAS)	Azure Active Directory (Azure AD)	Active Directory (versión preliminar)	Acceso de lectura anónimo
Azure Blobs	Compatible	Compatible	Compatible	No compatible	Compatible
Azure Files (SMB)	Compatible	No compatible	Admitido, solo con AAD Domain Services	Admitido, las credenciales deben sincronizarse con Azure AD	No compatible
Azure Files (REST)	Compatible	Compatible	No compatible	No compatible	No compatible
Colas de Azure	Compatible	Compatible	Compatible	No compatible	No compatible
Azure Tables	Compatible	Compatible	No compatible	No compatible	No compatible



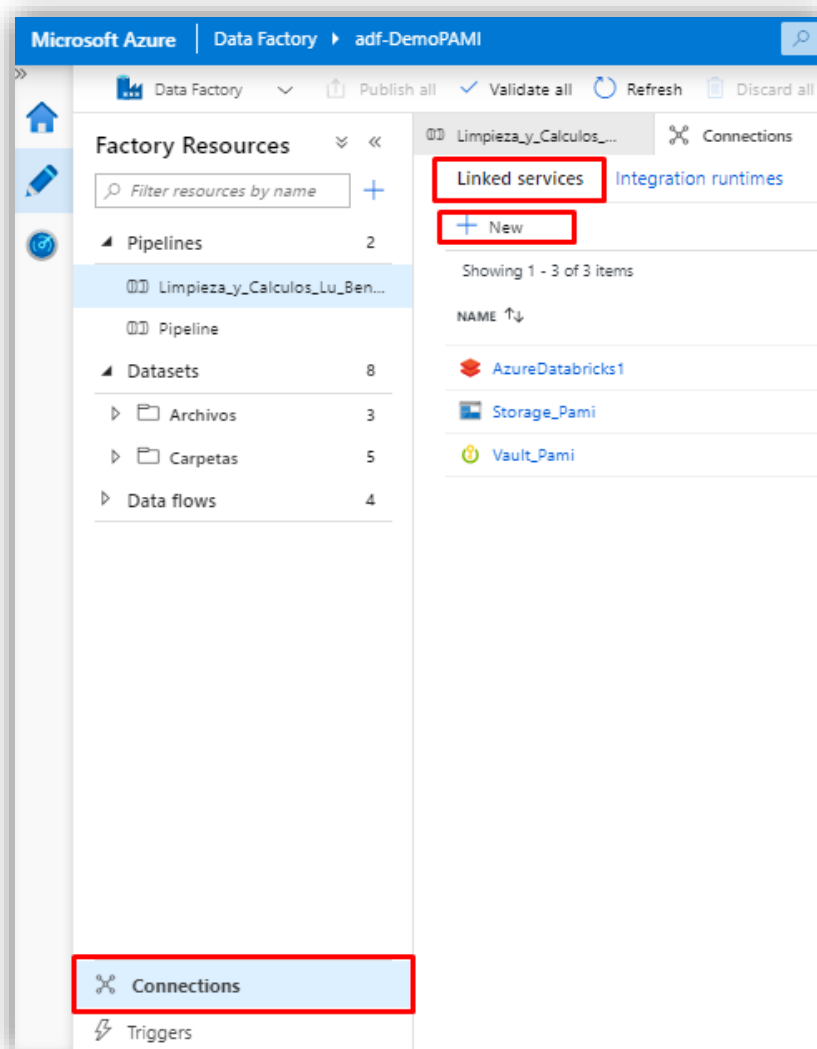
Anexo 3: Ejecución de una Notebook Databricks desde Data Factory

Dado el caso en que deseemos programar la ejecución de nuestra notebook, podemos realizar dicha actividad desde Data Factory.

Linked Service de Databricks

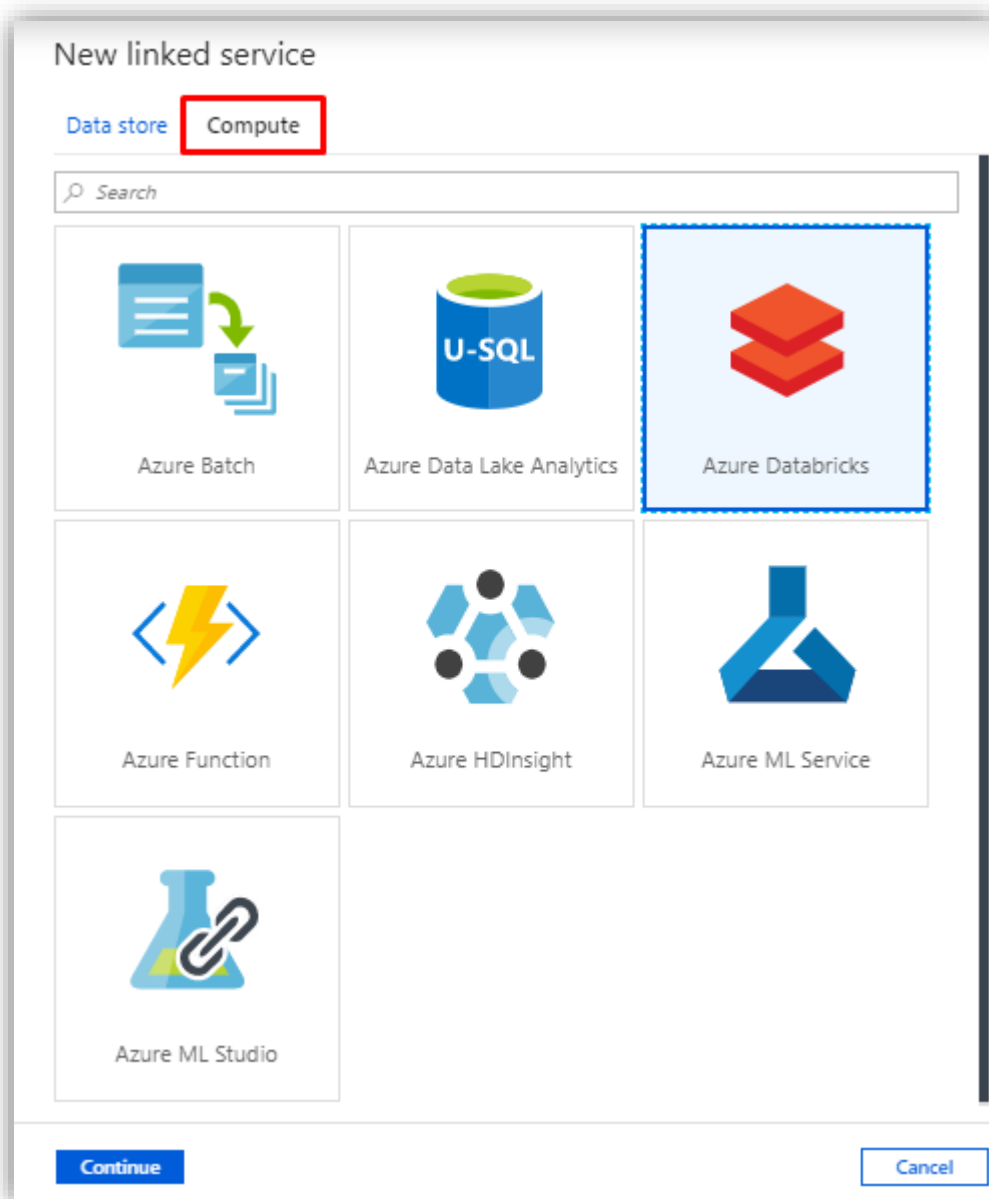
Nuestro primer paso será crear un nuevo Linked Service

Recordando lo explicado previamente, procederemos desde nuestro Data Factory, en la parte de edición, haciendo click en Connections – Linked Services - New





Aquí procederemos en Compute – Azure Databricks - Continue





Aquí pondremos el nombre del Linked Service a crear, utilizamos el Integration Runtime [IR] que creamos previamente, seleccionamos la cuenta desde la Suscripción de Azure, como así también nuestro Workspace.

Al seleccionar cluster, utilizaremos la primera opción, y la región se inferirá a partir de nuestro IR

New linked service (Azure Databricks)

Name *

LS_Databricks

Description

Connect via integration runtime *

IntegrationRuntime-Pami

[Edit integration runtime](#)

Account selection method *

From Azure subscription

Azure subscription *

Patrocinio de Microsoft Azure (08f96a01-421d-447c-ae35-e96908bc704d)

Databricks workspace *

adbr-DemoPAMI

Select cluster

☒ New job cluster ☐ Existing interactive cluster ☐ Existing instance pool

Domain/Region

https://eastus.azuredatabricks.net

Access token

Azure Key Vault

Access token *



Colocaremos el token de acceso de nuestro Workspace, la versión de cluster y Python a utilizar, y las opciones del Worker.

Access token Azure Key Vault

Access token *

Select cluster

☒ New job cluster ☐ Existing interactive cluster ☐ Existing instance pool

Cluster version *

5.5.x-conda-scala2.11

Cluster node type *

Standard_D3_v2

Python Version *

3

Worker options

☐ Fixed ☒ Autoscaling

Min Workers *

1

Max Workers *

2

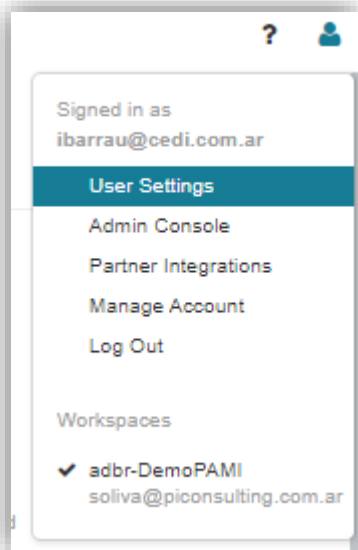
▶ Additional cluster settings

Annotations

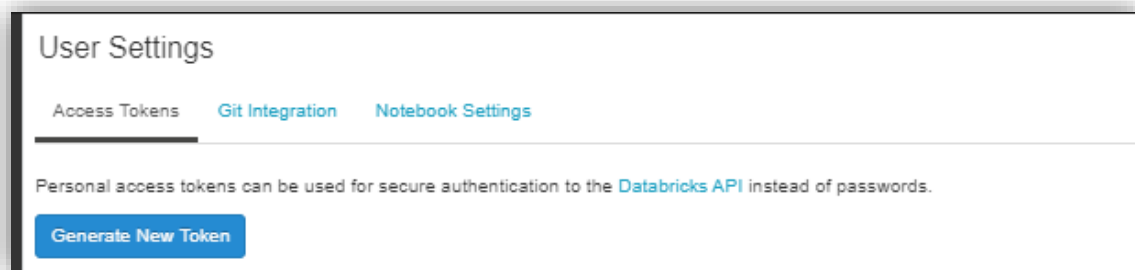


Para obtener el Access Token del Workspace, procederemos de la siguiente manera:

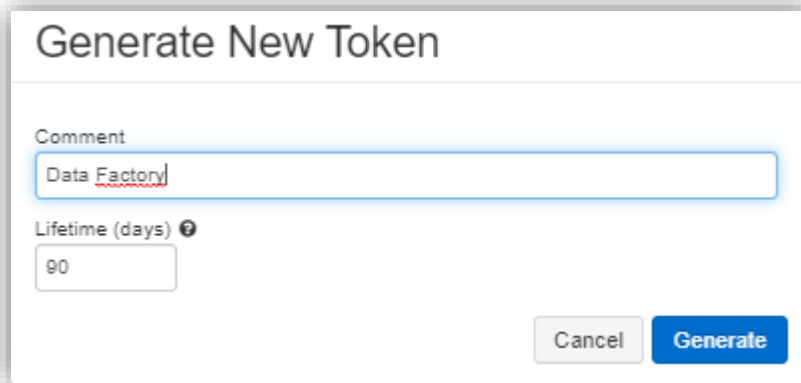
En nuestro Workspace de Databricks, haremos click en el Icono de Cuenta – User Settings



Haremos click en Generate New Token

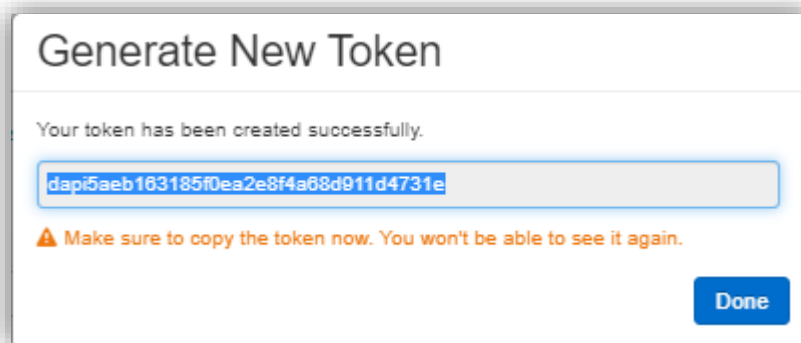


Aquí haremos un comentario para identificar para que es el token, y le daremos el tiempo de vida. Si este último espacio se deja en blanco, este token no caducará.



The dialog box is titled "Generate New Token". It contains a "Comment" field with the text "Data Factory" and a "Lifetime (days)" field with the value "90". There are "Cancel" and "Generate" buttons at the bottom right.

Finalmente se mostrará el token por única vez, este se mantendrá útil por el tiempo establecido.



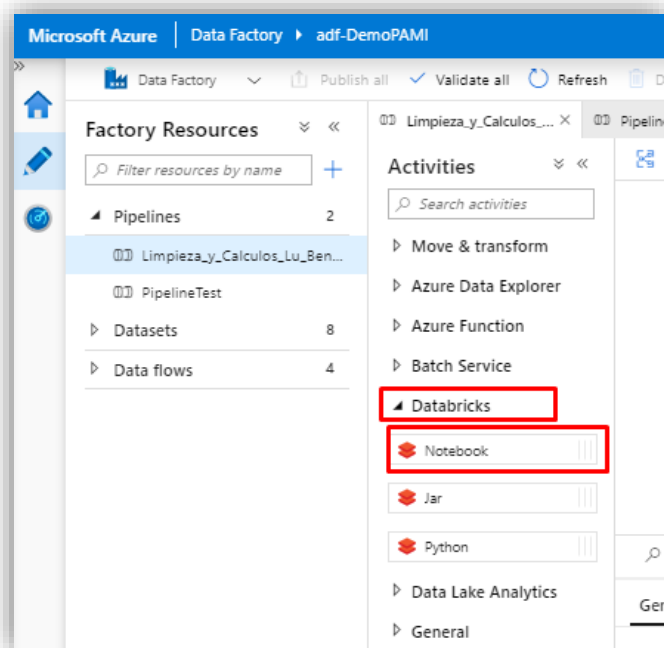
The dialog box is titled "Generate New Token". It displays the message "Your token has been created successfully." followed by a text box containing the token: "dapi5aeb163185f0ea2e8f4a68d911d4731e". Below the token, there is a warning message: "⚠ Make sure to copy the token now. You won't be able to see it again." and a "Done" button.

Este será el token que aplicaremos en nuestro Linked Service

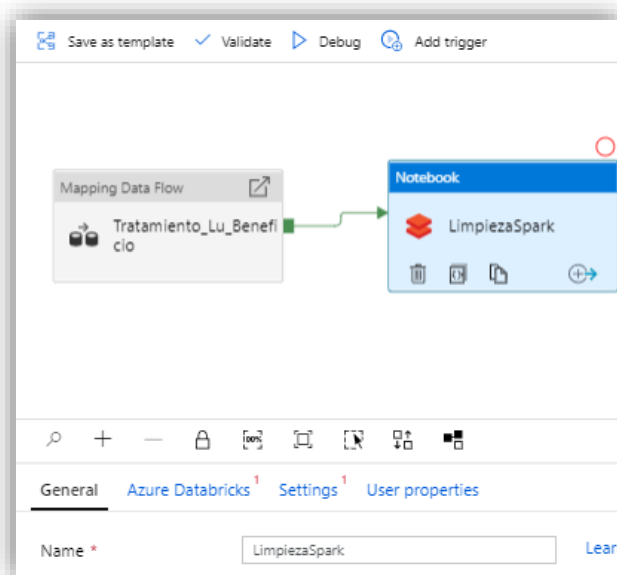
Uso desde Data Factory

Para generar una actividad de Databricks en Data factory se procede de la siguiente manera:

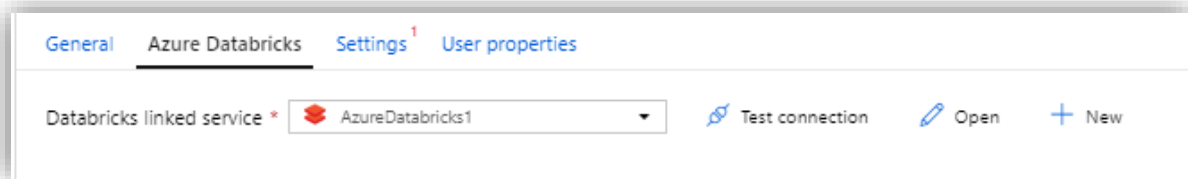
- En nuestro Data Factory, seleccionamos el pipeline donde programaremos la ejecución de la notebook en cuestión,
- Seleccionamos en la columna Activities Databricks - Notebook



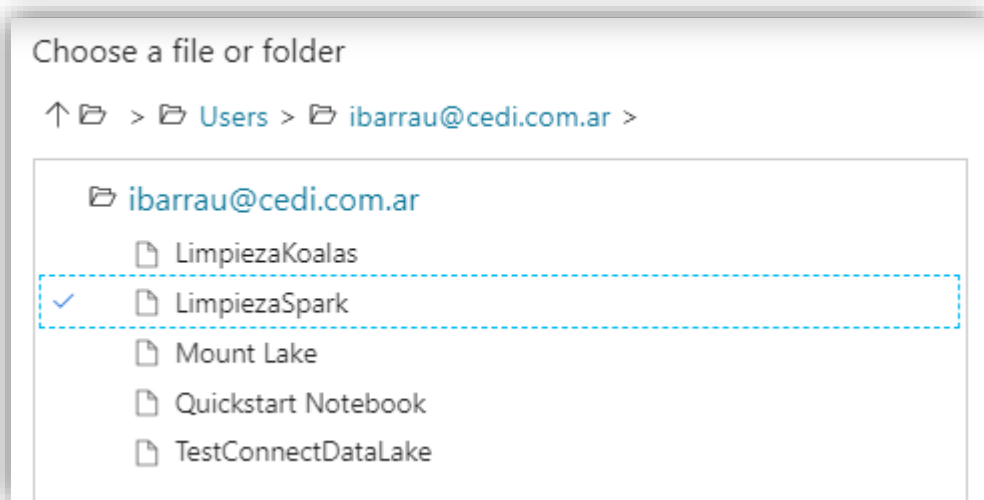
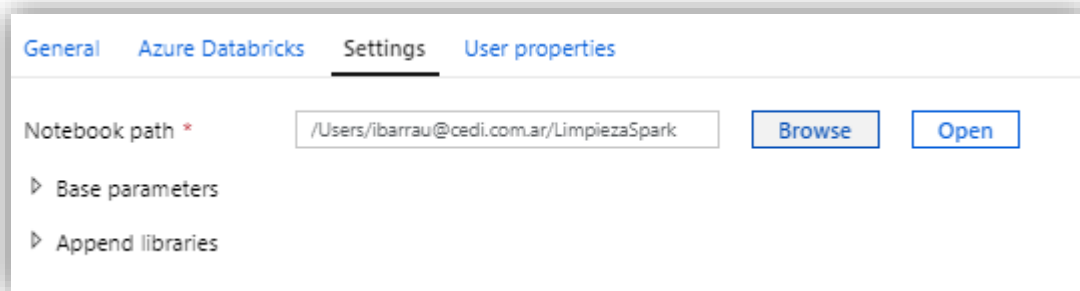
- Arrastramos la opción notebooks y configuraremos los siguientes ítems
- Primero le damos un nombre para identificar dicha actividad en Data factory



- En la siguiente pestaña, seleccionamos el Linked Service creado para nuestra notebook



- En Settings, daremos el path de donde se encuentra la notebook que queremos trabajar, utilizaremos el Browser para encontrar la indicada



- Finalmente haremos click en Publish All para guardar nuestros cambios

Anexo 1 : Permisos de Acceso a DataLake



Integración de Azure Active Directory (Azure AD) para blobs y colas.

Cuando una entidad de seguridad (un usuario, un grupo o una aplicación) intenta acceder a un recurso de blob o cola, la solicitud debe autorizarse, a menos que sea un blob disponible para acceso anónimo. Con Azure AD, el acceso a un recurso es un proceso de dos pasos. En primer lugar, se autentica la identidad de la entidad de seguridad y se devuelve un token de OAuth 2.0. Luego el token se pasa como parte de una solicitud a Blob Service o Queue Service y el servicio lo usa para autorizar el acceso al recurso especificado.

El paso de autenticación exige que una aplicación solicite un token de acceso de OAuth 2.0 en tiempo de ejecución.

El paso de autorización exige que se asignen uno o varios roles RBAC a la entidad de seguridad. Azure Storage proporciona roles RBAC que abarcan conjuntos comunes de permisos para datos de blobs y colas. Los roles que se asignan a una entidad de seguridad determinan los permisos que tiene esa entidad de seguridad. Para obtener más información sobre la asignación de roles RBAC en Azure Storage, vea [Administración de los derechos de acceso a los datos de almacenamiento con RBAC](#).



Acceso a datos con una cuenta de Azure AD

El acceso a los datos de blobs o colas a través de Azure Portal, PowerShell o la CLI de Azure se puede autorizar mediante la cuenta de Azure AD del usuario o las claves de acceso de cuenta (autorización de clave compartida).

Acceso a datos desde Azure Portal

Azure Portal puede usar la cuenta de Azure AD o las claves de acceso de cuenta para acceder a datos de blobs y colas en una cuenta de Azure Storage. El esquema de autorización que use Azure Portal depende de los roles RBAC que tenga asignados.

Si intenta acceder a datos de blobs o colas, Azure Portal primero comprueba si tiene asignado un rol RBAC con

Microsoft.Storage/storageAccounts/listkeys/action. Si tiene un rol asignado con esta acción, Azure Portal usa la clave de cuenta para acceder a los datos de blobs y colas mediante autorización de clave compartida. Si no tiene un rol asignado con esta acción, Azure Portal intenta acceder a los datos mediante la cuenta de Azure AD.

Para acceder a datos de blobs o colas desde Azure Portal con la cuenta de Azure AD, necesita permisos para acceder a datos de blobs y colas y, además, permisos para examinar los recursos de la cuenta de almacenamiento en Azure Portal. Los roles integrados que proporciona Azure Storage conceden acceso a recursos de blobs y colas, pero no conceden permisos a los recursos de la cuenta de almacenamiento.

Por este motivo, el acceso al portal también requiere la asignación de un rol de Azure Resource Manager, como el rol [Lector](#), con ámbito limitado al nivel de la cuenta de almacenamiento o superior. El rol **Lector** concede los permisos más restringidos, pero otro rol de Azure Resource Manager que conceda acceso a los recursos de administración de la cuenta de almacenamiento también es aceptable. Para obtener más información sobre cómo asignar permisos a los usuarios para el acceso a los datos de Azure Portal con una cuenta de Azure AD, vea [Concesión de acceso a datos de blob y cola de Azure con RBAC en Azure Portal](#).

Azure Portal indica qué esquema de autorización se está usando al examinar un contenedor o una cola. Para obtener más información sobre el acceso a datos en el portal, vea [Usar Azure Portal para acceder a datos de blob o cola](#).



Acceso a datos desde PowerShell o la CLI de Azure

PowerShell y la CLI de Azure admiten el inicio de sesión con credenciales de Azure AD. Después de iniciar sesión, la sesión se ejecuta con esas credenciales. Para obtener más información, vea [Ejecución de comandos de la CLI de Azure o PowerShell con credenciales de Azure AD para acceder a datos de blob o cola](#).



Roles RBAC integrados para blobs y colas

Azure proporciona los siguientes roles RBAC integrados para autorizar el acceso a datos de blob y cola con Azure AD y OAuth:

- [Propietario de datos de Storage Blob](#): se usa para establecer la propiedad y administrar el control de acceso POSIX en Azure Data Lake Storage Gen2. Para más información, consulte [Control de acceso en Azure Data Lake Storage Gen2](#).
- [Colaborador de datos de Storage Blob](#): se usa para conceder permisos de lectura, escritura y eliminación a los recursos de almacenamiento de blobs.
- [Lector de datos de Storage Blob](#): se usa para conceder permisos de solo lectura a los recursos de almacenamiento de blobs.
- [Colaborador de datos de la cola de Storage](#): se usa para conceder permisos de lectura, escritura y eliminación a las colas de Azure.
- [Lector de datos de la cola de Storage](#): se usa para conceder permisos de solo lectura a las colas de Azure.
- [Procesador de mensajes de datos de la cola de Storage](#): se usa para conceder permisos de inspección, recuperación y eliminación a los mensajes de las colas de Azure Storage.
- [Emisor de mensajes de datos de la cola de Storage](#): se usa para conceder permisos de adición a los mensajes de las colas de Azure Storage.

Glosario

Es útil entender algunos términos clave relacionados con la autenticación de Azure AD Domain Service sobre SMB para recursos compartidos de archivos de Azure:

- **Autenticación Kerberos**

Kerberos es un protocolo de autenticación que se utiliza para comprobar la identidad de un usuario o un host. Para más información sobre Kerberos, consulte [Introducción a la autenticación Kerberos](#).

- **Protocolo Bloque de mensajes del servidor (SMB)**

SMB es un protocolo de uso compartido de archivos de red estándar del sector. SMB también se conoce como sistema de archivos de Internet común o CIFS. Para más información sobre SMB, consulte [Microsoft SMB Protocol and CIFS Protocol Overview](#) (Introducción a los protocolos CIFS y SMB de Microsoft).

- **Azure Active Directory (Azure AD)**

Azure Active Directory (Azure AD) es el directorio multiinquilino basado en la nube y el servicio de administración de identidades de Microsoft. Azure AD combina servicios de directorio fundamentales, administración del acceso a las aplicaciones y protección de identidades en una única solución. Azure AD permite que las máquinas virtuales (VM) Windows unidas a un dominio tengan acceso a los recursos compartidos de archivos de Azure con sus credenciales de Azure AD. Para más información, consulte [¿Qué es Azure Active Directory?](#)



- **Azure AD Domain Services (Azure AD DS)**

Azure AD Domain Services (disponibilidad general) proporciona servicios de dominio administrados como, por ejemplo, unión a un dominio, directivas de grupo, LDAP y autenticación Kerberos/NTLM. Estos servicios son totalmente compatibles con Windows Server Active Directory. Para más información, consulte [Azure Active Directory \(AD\) Domain Services](#).

- **Active Directory Domain Services (AD DS, también denominado AD)**

Active Directory (AD) (versión preliminar) proporciona los métodos para almacenar datos de directorio y poner dichos datos a disposición de los usuarios y administradores de la red. La seguridad se integra en Active Directory mediante la autenticación de inicio de sesión y el control de acceso a los objetos del directorio. Con un único inicio de sesión de red, los administradores pueden administrar los datos del directorio y la organización a través de su red, y los usuarios de red autorizados pueden tener acceso a los recursos en cualquier parte de la red. Normalmente son empresas en entornos locales las que adoptan AD, y usan las credenciales de AD como identidad para el control de acceso. Para obtener más información, consulte [Introducción a Active Directory Domain Services](#).

- **Control de acceso basado en rol (RBAC) de Azure**

El control de acceso basado en roles (RBAC) de Azure permite realizar una administración detallada del acceso para Azure. Con RBAC, puede administrar el acceso a los recursos mediante la concesión a los usuarios del menor número de permisos necesarios para realizar su trabajo. Para obtener más información sobre RBAC, consulte [¿Qué es el control de acceso basado en rol \(RBAC\) en Azure?](#)



Autorizar con clave compartida [SAS]

Toda solicitud realizada contra un servicio de almacenamiento debe estar autorizada, a menos que la solicitud sea para un recurso blob o contenedor que se haya puesto a disposición para acceso público o firmado. Una opción para autorizar una solicitud es mediante el uso de clave compartida, descrita en este artículo.

Especificando el encabezado de Autorización

Una solicitud autorizada debe incluir el encabezado de Autorización.

Si este encabezado no está incluido, el pedido es anónimo y solo puede ser efectivo ante un container o Blob que haya sido configurado para acceso público, o cuya firma de acceso compartido (SAS) sea provista por acceso delegado.

Para autorizar un pedido, es necesario firmarlo con la Key para la cuenta que está realizando el pedido, y pasar esta firma como parte del pedido.

El formato del encabezado de la Autorización es el siguiente:

```
Authorization="[SharedKey|SharedKeyLite] <AccountName>:<Signature>"
```

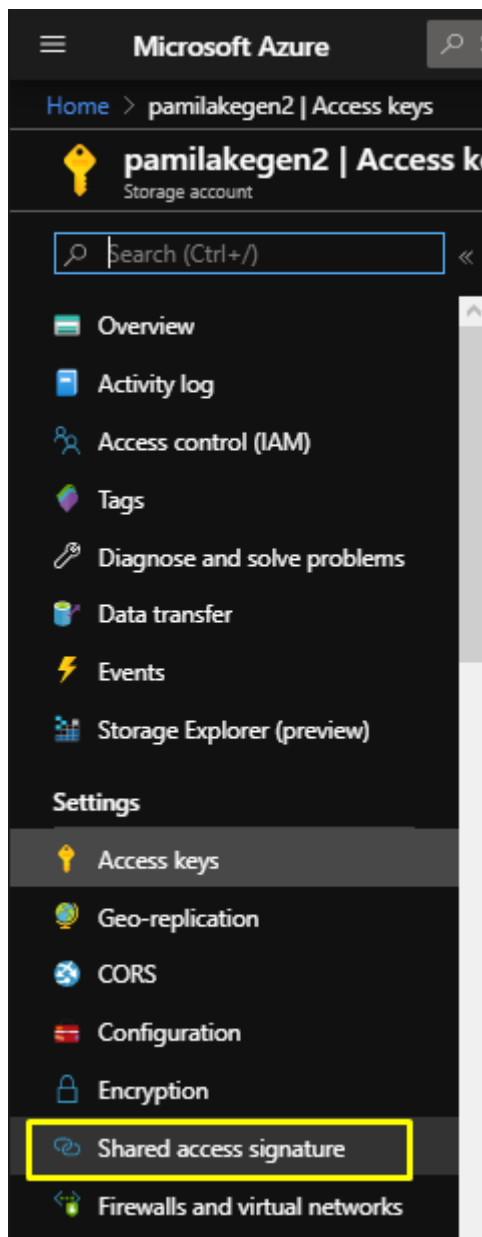
Donde:

- SharedKey o SharedKeyLite es el nombre del esquema de autorización
- AccountName es el nombre de la cuenta que realiza el pedido al recurso
- Signature es un Código de Autenticación de Mensaje basado en Hash (HMAC)



Construyendo el string de firma

Accediendo a la configuración del DataLake, iremos a la opción de Shared Access Signature





A shared access signature (SAS) is a URI that grants restricted access rights to Azure Storage resources. You can provide a shared access signature to clients who should not be trusted with your storage account key but whom you wish to delegate access to certain storage account resources. By distributing a shared access signature URI to these clients, you grant them access to a resource for a specified period of time.

An account-level SAS can delegate access to multiple storage services (i.e. blob, file, queue, table). Note that stored access policies are currently not supported for an account-level SAS.

[Learn more](#)

Allowed services ⓘ

☒ Blob ☒ File ☒ Queue ☒ Table

Allowed resource types ⓘ

☒ Service ☒ Container ☒ Object

Allowed permissions ⓘ

☒ Read ☒ Write ☒ Delete ☒ List ☐ Add ☐ Create ☐ Update ☒ Process

Start and expiry date/time ⓘ

Start

03/19/2020 2:56:46 PM

End

03/19/2020 10:56:46 PM

(UTC-03:00) --- Current Time Zone ---

Allowed IP addresses ⓘ

for example, 168.1.5.65 or 168.1.5.65-168.1.5.70

Allowed protocols ⓘ

☒ HTTPS only ☐ HTTPS and HTTP

Signing key ⓘ

key1

En esta pantalla seleccionaremos que servicios y recursos queremos permitir, como así también que permisos otorgaremos.

Luego seleccionaremos la fecha de inicio y de fin de validez del SAS, los IP permitidos, los protocolos permitidos y la key del Lake utilizada

Clickeando Generar String, obtendremos un conjunto de strings disponible

03/19/2020 10:56:46 PM

(UTC-03:00) --- Current Time Zone ---

Allowed IP addresses ⓘ

for example, 168.1.5.65 or 168.1.5.65-168.1.5.70

Allowed protocols ⓘ

☒ HTTPS only ☐ HTTPS and HTTP

Signing key ⓘ

key1

[Generate SAS and connection string](#)

Connection string

BlobEndpoint=https://pamilakegen2.blob.core.windows.net/;QueueEndpoint=https://pamilakegen2.queue.core.windows.net/;FileEndpoint=https://pamilakegen2.file.core.windows.net/;TableEndpoint=...

SAS token ⓘ

?sv=2019-02-02&ss=bfqt&srt=sco&sp=rwdlp&se=2020-03-20T01:56:46Z&st=2020-03-19T17:56:46Z&spr=https&sig=0iwDIN697oJVjIPuuU9LPEocf5w4kE...

Blob service SAS URL

https://pamilakegen2.blob.core.windows.net/?sv=2019-02-02&ss=bfqt&srt=sco&sp=rwdlp&se=2020-03-20T01:56:46Z&st=2020-03-19T17:56:46Z&spr=https&sig=0iwDIN697oJVjIPuuU9LPEocf5w4kE...

File service SAS URL

https://pamilakegen2.file.core.windows.net/?sv=2019-02-02&ss=bfqt&srt=sco&sp=rwdlp&se=2020-03-20T01:56:46Z&st=2020-03-19T17:56:46Z&spr=https&sig=0iwDIN697oJVjIPuuU9LPEocf5w4kE...

Queue service SAS URL

https://pamilakegen2.queue.core.windows.net/?sv=2019-02-02&ss=bfqt&srt=sco&sp=rwdlp&se=2020-03-20T01:56:46Z&st=2020-03-19T17:56:46Z&spr=https&sig=0iwDIN697oJVjIPuuU9LPEocf5w4kE...

Table service SAS URL

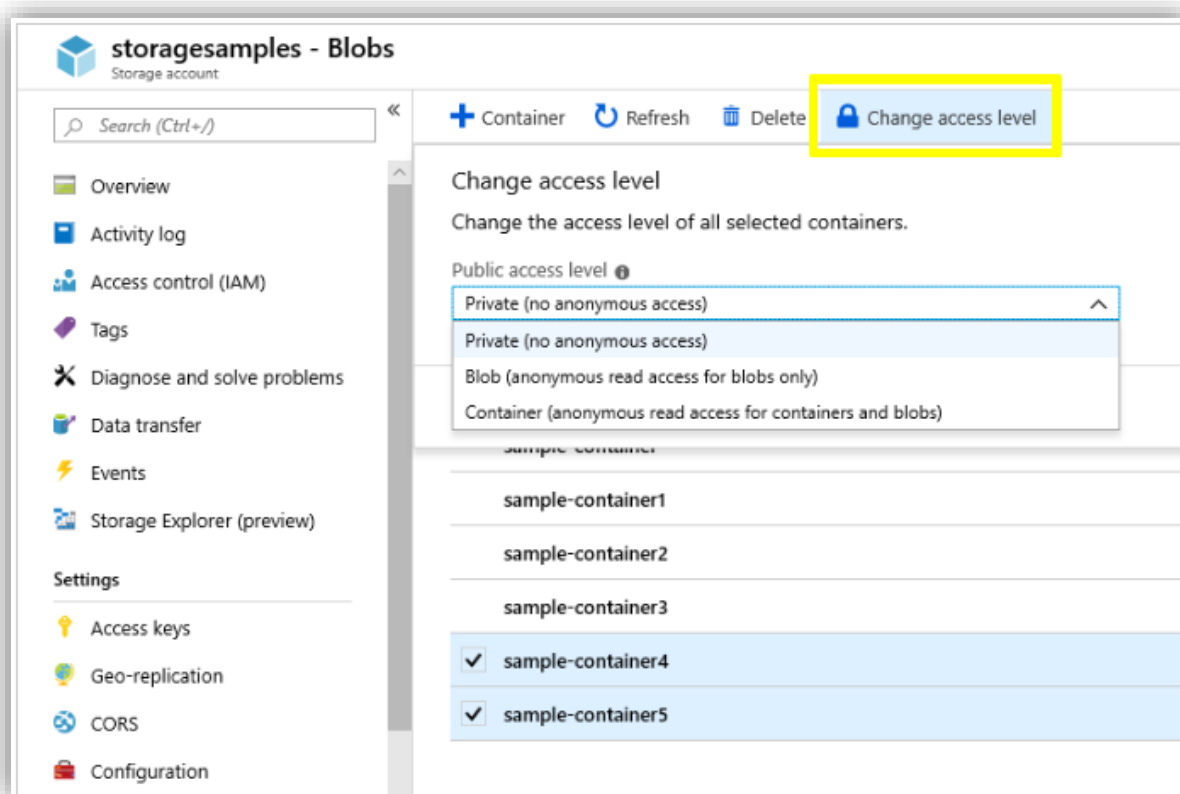
https://pamilakegen2.table.core.windows.net/?sv=2019-02-02&ss=bfqt&srt=sco&sp=rwdlp&se=2020-03-20T01:56:46Z&st=2020-03-19T17:56:46Z&spr=https&sig=0iwDIN697oJVjIPuuU9LPEocf5w4kE...

Administración del acceso de lectura anónimo a contenedores y blobs

Finalmente, se puede habilitar el acceso de lectura anónimo y público a un contenedor y sus blobs en Azure Blob Storage. Al hacerlo, puede conceder acceso de solo lectura a estos recursos sin compartir la clave de cuenta y sin necesidad de una firma de acceso compartido (SAS).

El acceso de lectura público es mejor para escenarios donde desea que ciertos blobs estén siempre disponibles para el acceso de lectura anónimo. Para un control más minucioso, puede crear una firma de acceso compartido. Las firmas de acceso compartido le permiten proporcionar acceso restringido con distintos permisos para un período específico.

En un Blob Storage, esto se implementaría de la siguiente manera





Anexo 2: Implementación en Databricks

Para trabajar en Databricks, previo a todo código, montaremos el Data Lake, cosa que se ha explicado previamente.

Nuestro Lake esta ordenado de la siguiente manera:

- Raw: aquí se guardan los archivos sin tratamiento, este Blob contiene las siguientes carpetas:
 - o Zip: donde se cargan los archivos en primera instancia, estos se encuentran comprimidos en formato .zip
 - o Extract: aquí se guardan los archivos que descomprimimos cuya fuente se ubicaba en la carpeta ZIP.
- TransientData: aquí se guardarán los archivos en tratamiento
- Trusted: aquí se guardarán los archivos listos para ser consumidos, ya limpios.

Utilizaremos los comandos para navegar el Data Lake para encontrar el camino o path de los archivos que queremos trabajar

```
1 %fs ls /mnt/
```

path	name	size
dbfs:/mnt/pruebas/	pruebas/	0
dbfs:/mnt/raw/	raw/	0
dbfs:/mnt/transientData/	transientData/	0
dbfs:/mnt/trusted/	trusted/	0

Haciendo uso de los avances realizados en nuestro pipeline en cuestiones de limpieza, realizado por DataFlow, buscaremos nuestro archivo en transientData.

```
1 %fs ls /mnt/transientData
```

path	name	size
dbfs:/mnt/transientData/Lu_Beneficio_Limpio.Csv	Lu_Beneficio_Limpio.Csv	633826628

Utilizando el path identificado previamente leeremos el csv

```
1 lb_clean= "/mnt/transientData/Lu_Beneficio_Limpio.Csv"
2 df_bene = spark.read.csv( lb_clean ,header=True , sep = ",")
```

Podremos ver la metadata del mismo con el siguiente comando

```
1 print(df_bene.printSchema())

root
 |-- N_BENE_PERSONA: string (nullable = true)
 |-- C_BENE_DET_PAREN: string (nullable = true)
 |-- C_BENE_ESTADO_CIVIL: string (nullable = true)
 |-- C_BENE_TIPO_BENEFICIO: string (nullable = true)
 |-- C_BENE_DET_AFJP: string (nullable = true)
 |-- C_BENE_ESTADO_AFIL: string (nullable = true)
 |-- C_GEOG_LOCALIDAD: string (nullable = true)
 |-- F_BENE_NACIMIENTO: string (nullable = true)
 |-- C_BENE_NACION: string (nullable = true)
 |-- N_RUN_ID: string (nullable = true)
 |-- C_GEOG_AGENCIA: string (nullable = true)
 |-- ID_BENE_BENEFICIO: string (nullable = true)
 |-- F_BENE_ALTA: string (nullable = true)
 |-- F_BENE_BAJA: string (nullable = true)
 |-- C_GEOG_UGL: string (nullable = true)
 |-- ID_BENE_SEXO: string (nullable = true)
 |-- ID_BENE_EDAD: string (nullable = true)
 |-- F_DWH_CREA: string (nullable = true)
 |-- F_DWH_ACTU: string (nullable = true)
 |-- ID_BENE_OBRA_SOCIAL: string (nullable = true)
```

Con la función display podremos ver el contenido del archivo en cuestión

```
1 display(df_bene)
```

► (1) Spark Jobs

N_BENE_PERSONA ▼	C_BENE_DET_PAREN ▼	C_BENE_ESTADO_CIVIL ▼	C_BENE_TIPO_BENEFICIO ▼	C_BENE_DET_AFJP ▼	C_BENE_ESTADO ▼
5807258	1	2	10	REPA	true
5807272	0	2	30	DESC	true

< Showing the first 1000 rows. >

Utilizo Spark para hacer una pequeña operación y crear un nuevo dataframe.

```
1 from pyspark.sql.functions import col, sum, mean, count
2 lb_añoCantidad = (df_bene
3     .select("Anio_nacimiento_bene")
4     .groupBy(col("Anio_nacimiento_bene"))
5     .count()
6     .orderBy(col("Anio_nacimiento_bene"))
```

```
1 display(lb_añoCantidad)
```

► (1) Spark Jobs

Anio_nacimiento_bene ▲	count ▼
1991	19435
1990	18934
1989	18621
1988	19297
1987	18185
1986	19336

Un ejemplo sobre como guardar dataframes en diferentes formatos

```
1 lb_añoCantidad.write.format("avro").mode("overwrite").save("/mnt/trusted/Lu_Bene_Anio_Cantidad.avro")
2 df_bene.write.format("parquet").mode("overwrite").save("/mnt/trusted/Lu_bene.parquet")
```

► (3) Spark Jobs

Command took 47.32 seconds -- by ibarrau@cedi.com.ar at 2/4/2020 11:34:59 on Cluster1

Utilizamos el modo "overwrite" para asegurar la correcta ejecución del script cuando lo programemos en ejecuciones sucesivas.



Chequeando que se hayan guardado nuestros dataframes, veremos que el único archivo que muestra su tamaño es el .csv, esto es porque en los formatos AVRO y PARQUET, los archivos se guardan en varios nodos simultáneamente.

```
1 %fs ls /mnt/trusted/
```

path	name	size
dbfs:/mnt/trusted/Lu_Bene_Anio_Cantidad.avro/	Lu_Bene_Anio_Cantidad.avro/	0
dbfs:/mnt/trusted/Lu_Bene_Anio_Cantidad.parquet/	Lu_Bene_Anio_Cantidad.parquet/	0
dbfs:/mnt/trusted/Lu_bene.parquet/	Lu_bene.parquet/	0
dbfs:/mnt/trusted/Promedio_Nacimiento.csv	Promedio_Nacimiento.csv	8768

Observemos el tiempo de carga en distintos formatos

```
1 lb_clean= "/mnt/transientData/Lu_Beneficio_Limpio.Csv"|
2 df_bene = spark.read.csv( lb_clean ,header=True , sep = ",")
```

► (1) Spark Jobs
► df_bene: pyspark.sql.dataframe.DataFrame = [N_BENE_PERSONA: string, C_BENE_DET_PAREN: string ... 32 more fields]
Command took 5.51 seconds -- by ibarrau@cedi.com.ar at 2/4/2020 11:34:56 on Cluster1

```
3 lb_pq_path = "/mnt/trusted/Lu_bene.parquet"|
4 lb_pq = spark.read.parquet(lb_pq_path)
```

► (1) Spark Jobs
► lb_pq: pyspark.sql.dataframe.DataFrame = [N_BENE_PERSONA: string, C_BENE_DET_PAREN: string ... 32 more fields]
Command took 0.56 seconds -- by ibarrau@cedi.com.ar at 2/4/2020 11:35:03 on Cluster1

```
1 lb_av_path = "/mnt/trusted/Lu_bene.avro"
2 lb_avro = spark.read.format("avro").load(lb_av_path)
```

► lb_avro: pyspark.sql.dataframe.DataFrame = [N_BENE_PERSONA: string, C_BENE_DET_PAREN: string ... 32 more fields]
Command took 0.98 seconds -- by ibarrau@cedi.com.ar at 2/4/2020 12:33:01 on Cluster1

Comparando estos tiempos, podemos observar que utilizando esta metodología, la lectura de archivos se acelera 5 veces en el caso del formato AVRO y 11 veces en el formato PARQUET.

