

An abstract graphic in the top right corner of the cover. It features a dark, semi-circular shape resembling a globe or a data sphere. Inside this shape, there are numerous thin, glowing lines in shades of purple, pink, and blue. These lines radiate from a central point and connect to various points on the sphere's surface, creating a network-like or data flow visualization. The background of the entire cover is black.

# Azure Data Factory by Example

Practical Implementation for  
Data Engineers

—  
Richard Swinbank

Apress®

## Contenido

<b>5. Parámetros</b>	<b>3</b>
<b>5.1. Configurar una Azure Key Vault</b>	<b>3</b>
5.1.1. Crear un Key Vault	3
5.1.2. Crear un Key Vault Secret	4
5.1.3. Conceder acceso a la Key Vault (Grant Access to the Key Vault)	7
5.1.4. Crear un Key Vault ADF Linked Service	9
5.1.5. Crear un nuevo servicio vinculado a la cuenta de almacenamiento	11
<b>5.2. Utilizar Dataset Parameters</b>	<b>14</b>
5.2.1. Crear un dataset parametrizado	15
5.2.2. Utilizar el dataset parametrizado	18
5.2.3. Reutilizar el dataset parametrizado	20
<b>5.3. Utilizar Linked Service Parameters</b>	<b>22</b>
5.3.1. Crear un linked service parametrizado	22
5.3.2. Aumentar la reutilización del dataset	27
5.3.3. Utilice el nuevo dataset	29
5.3.4. ¿Por qué parametrizar los servicios vinculados?	30
<b>5.4. Utilizar Pipeline Parameters</b>	<b>32</b>
5.4.1. Crear un pipeline parametrizado	32
5.4.2. Ejecutar el pipeline parametrizado	35
5.4.3. Utilice la actividad Execute Pipeline	38
5.4.4. Ejecución paralela	41
<b>5.5. Parámetros globales</b>	<b>42</b>
<b>5.6. Repaso del capítulo</b>	<b>43</b>
5.6.1. Conceptos clave	43
5.6.2. Para los desarrolladores de SSIS	44

## 5. Parámetros

---

El capítulo 4 introdujo las expresiones como una forma de establecer los valores de las propiedades en los recursos de la fábrica en tiempo de ejecución. Los ejemplos presentados utilizaban expresiones para determinar los valores de una variedad de propiedades, todas ellas bajo el control interno del pipeline. Pero a veces es conveniente poder inyectar valores externos en los recursos de fábrica en tiempo de ejecución, ya sea para compartir datos o para crear recursos genéricos que puedan ser reutilizados en múltiples escenarios. La inyección de valores en tiempo de ejecución se consigue mediante el uso de parámetros.

Los parámetros en tiempo de ejecución son soportados no sólo por los pipelines, sino también por los datasets, los linked services y los data flows. Los data flows se introducen en el capítulo 7 - este capítulo examina el uso de parámetros con los otros tipos de recursos de fábrica. Los parámetros también pueden definirse a nivel de una instancia de data factory. A diferencia de otros tipos de parámetros, estos parámetros globales no permiten la sustitución de valores en tiempo de ejecución, sino que representan constantes, compartidas por todos los pipelines.

Utilizarás una instancia de Azure Key Vault para poder parametrizar servicios vinculados de forma segura, por lo que en la primera parte del capítulo, crearás y configurarás tu key vault.

### 5.1. Configurar una Azure Key Vault

Azure Key Vault proporciona un repositorio seguro, basado en la nube, para las claves criptográficas y otros secretos como las cadenas de conexión (connection strings) de la base de datos o de la cuenta de almacenamiento. En esta sección, añadirá una Key Vault a su grupo de recursos, almacenará una cadena de conexión dentro de ella y configurará un servicio vinculado de Azure Data Factory utilizando la cadena de conexión almacenada de forma segura.

#### 5.1.1. Crear un Key Vault

Las instancias de Azure Key Vault se crean y gestionan en el portal de Azure.

1. Abra el portal y cree un nuevo recurso de tipo Key Vault.
2. Seleccione la suscripción y el grupo de recursos que contenga su instancia de ADF y, a continuación, elija un nombre de Key Vault único a nivel mundial.
3. Elija la Región más cercana geográficamente.
4. Asegúrese de que el nivel de precios está configurado como "Standard", luego acepte el resto de los valores por defecto haciendo clic en Revisar + crear. (La Figura 5-1 muestra la pestaña Básica de la hoja de creación de almacén de claves.

Home > New > Key Vault >

## Create key vault

your resources:

Subscription \* Free Trial

Resource group \* adfbexample-rg

Instance details

Key vault name \* adfbexample-kv

Region \* UK South

Pricing tier \* Standard

[Review + create](#) [< Previous](#) [Next: Access policy >](#)

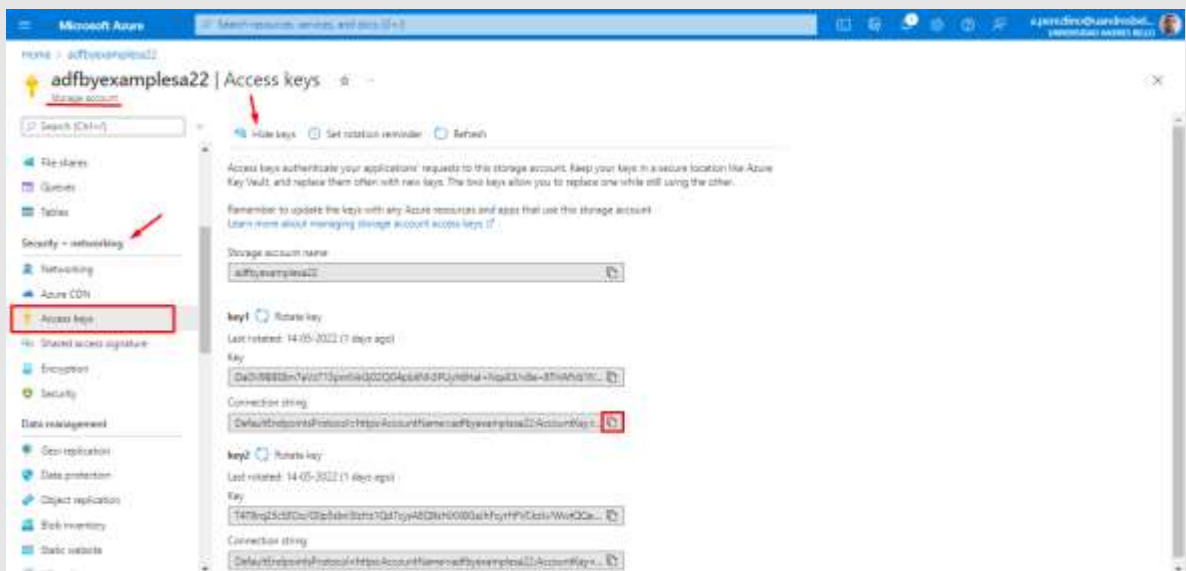
Figura 5-1 Pestaña Crear key vault Aspectos básicos

- Después de la validación, haga clic en Crear. Se mostrará un mensaje de notificación cuando se haya completado el despliegue.

### 5.1.2. Crear un Key Vault Secret

La cadena de conexión de tu storage account se encuentra en el portal de Azure. En esta sección, almacenará su valor de forma segura en la nueva key vault.

- Abre la hoja del portal para tu cuenta de almacenamiento (puedes encontrarla en la lista de recursos contenidos en tu grupo de recursos o utilizando la casilla Buscar recursos, servicios y docs (G + /) en la barra de herramientas del portal).
- Seleccione Access keys en el grupo Settings de la barra lateral. Para copiar los valores de las claves, primero debe hacer clic en Mostrar claves. La Figura 5-2 muestra la hoja después de hacerlo: el botón Mostrar claves ha sido sustituido por Ocultar claves. Los valores secretos han sido truncados en la captura de pantalla.



3. Copie el valor de la cadena de conexión para key1 en el portapapeles utilizando el botón de copia en el extremo derecho del campo (indicado en la Figura 5-2).

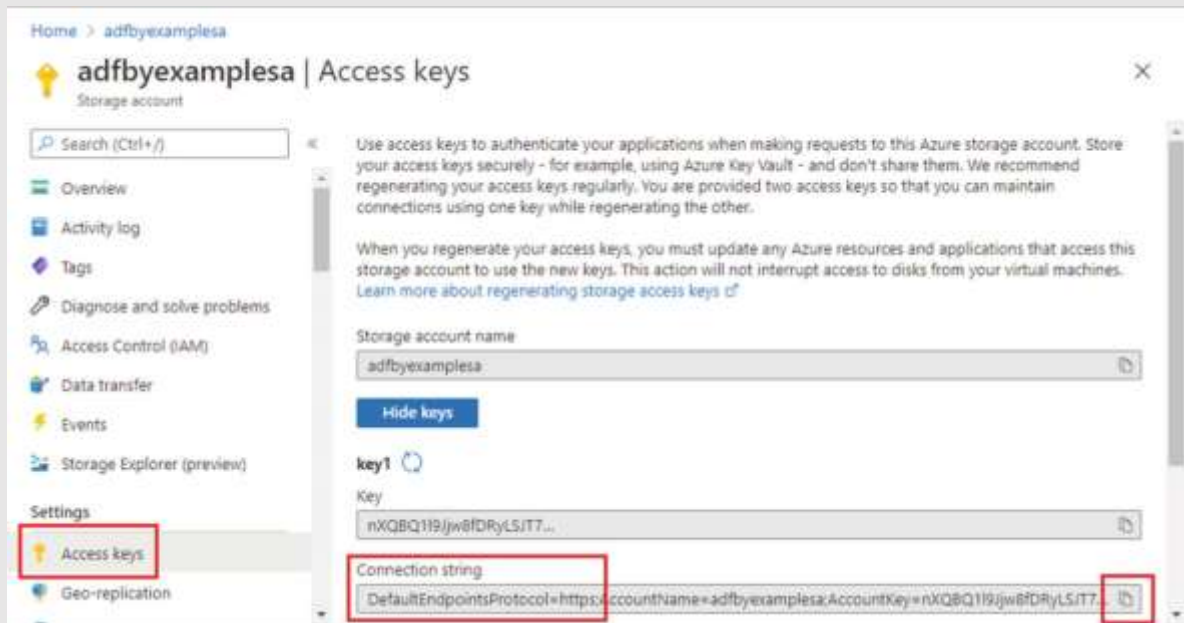
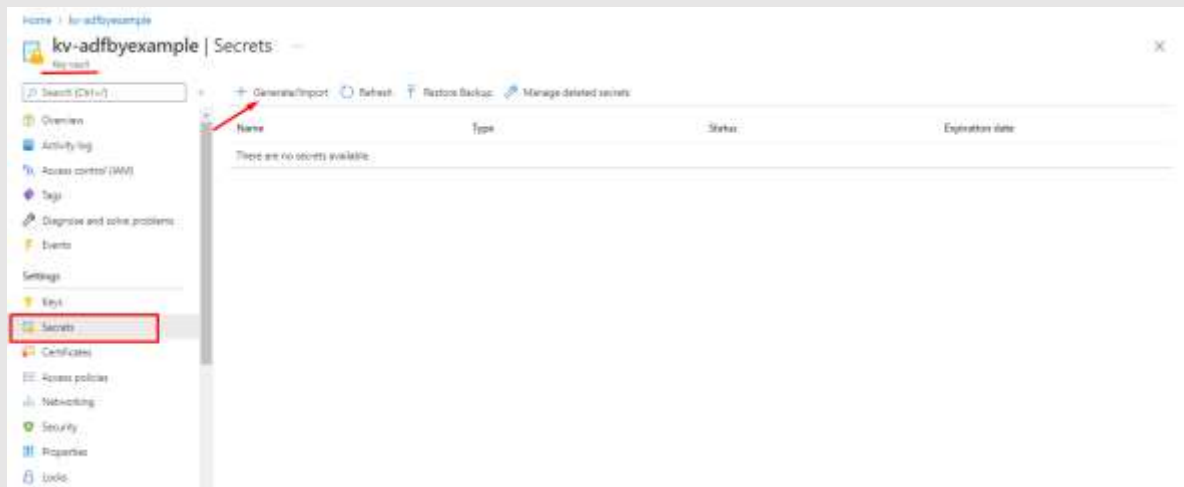


Figura 5-2 Claves de acceso a la cuenta de almacenamiento

4. Abra la hoja del portal de su key vault y seleccione Secrets en el grupo Settings de la barra lateral. En la parte superior de la hoja de Secretos, haga clic en el botón + Generar/Importar para añadir un nuevo secreto.



5. Aparece la hoja Crear un secreto. Pegue el contenido del portapapeles en el campo Value (valor) y, a continuación, introduzca un nombre para el secreto. La Figura 5-3 muestra el formulario completado para mi secreto "StorageAccountConnectionString". Haga clic en Crear.

Microsoft Azure

Home > kv-adfbyexample >

### Create a secret

Upload options: Manual

Name: StorageAccountConnectionString ✓

Value: ..... ✓

Content type (optional):

Set activation date: ☐

Set expiration date: ☐

Enabled: Yes No

Tags: 0 tags

Create

Home > kv-adfbyexample > kv-adfbyexample | Secrets

Search (Ctrl+F)

Generate keypair Refresh Restore Backup Manage stored secrets

The secret 'StorageAccountConnectionString' has been successfully deleted.

Name	Type	Status	Expiration date
StorageAccountConnectionString		✓ Enabled	

Home > adfbyexample-kv >

### Create a secret

Upload options: Manual

Name: StorageAccountConnectionString ✓

Value: ..... ✓

Content type (optional):

Set activation date? ☐

Set expiration date? ☐

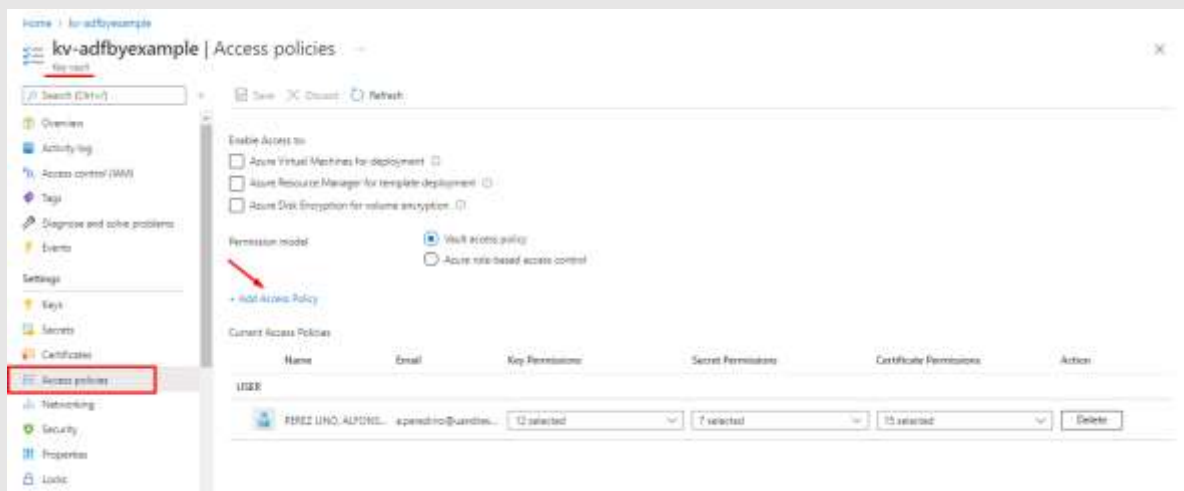
Create

Figura 5-3 Creación de un secreto en el key vault

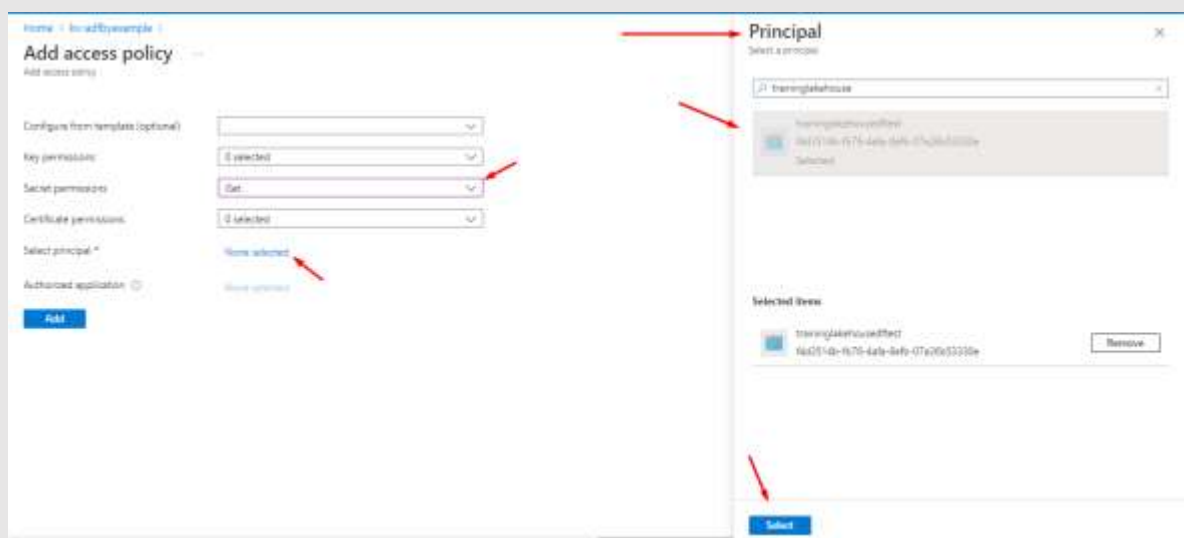
### 5.1.3. Conceder acceso a la Key Vault (Grant Access to the Key Vault)

Tu data factory no puede usar los secretos almacenados en tu key vault hasta que le des permiso para hacerlo. La instancia de la data factory tiene una identidad gestionada asociada - una aplicación gestionada registrada en Azure Active Directory - que se creó automáticamente cuando usted creó la data factory. Debe conceder acceso a esta identidad.

1. Abra la hoja del portal para su key vault y seleccione Access policies en el grupo Settings de la barra lateral. En la hoja de políticas de acceso, localice y haga clic en el botón + Añadir política de acceso.



2. En la hoja de Add access policy, seleccione Get en el menú desplegable Secret permissions, luego en Select principal, haga clic en None selected - esto abre la hoja de selección del security principal.
3. En la parte superior de la hoja hay un campo de entrada de búsqueda. Una ADF managed identity service principal tiene el mismo nombre que la instancia de ADF que representa - introduzca el nombre de su data factory para buscar la service principal. La búsqueda devolverá un elemento coincidente, como se muestra en la Figura 5-4. Haga clic en el elemento para elegirlo y, a continuación, haga clic en el botón Seleccionar de la parte inferior de la hoja.



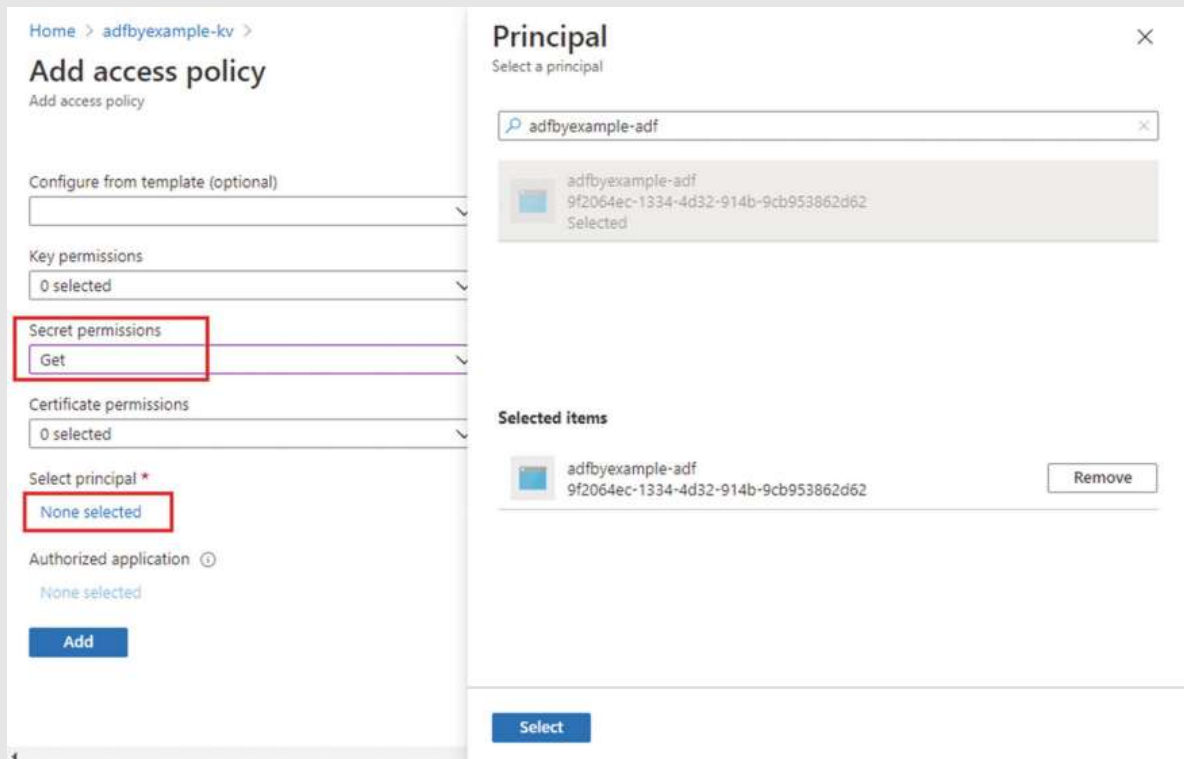
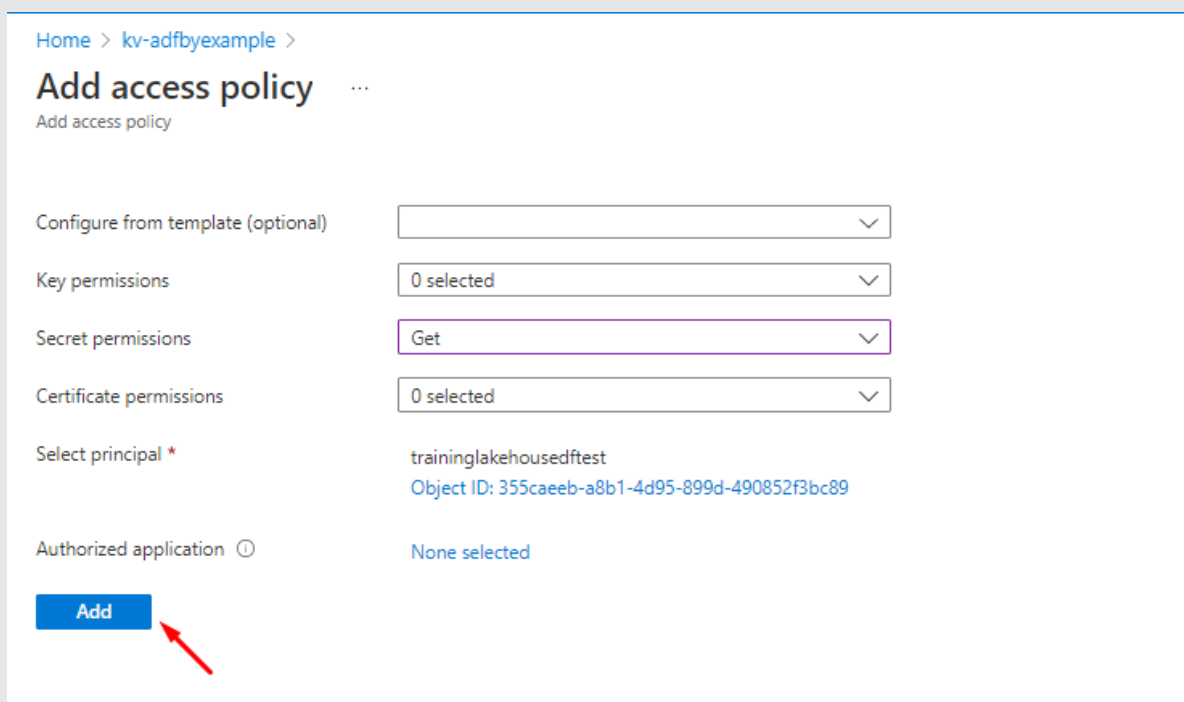
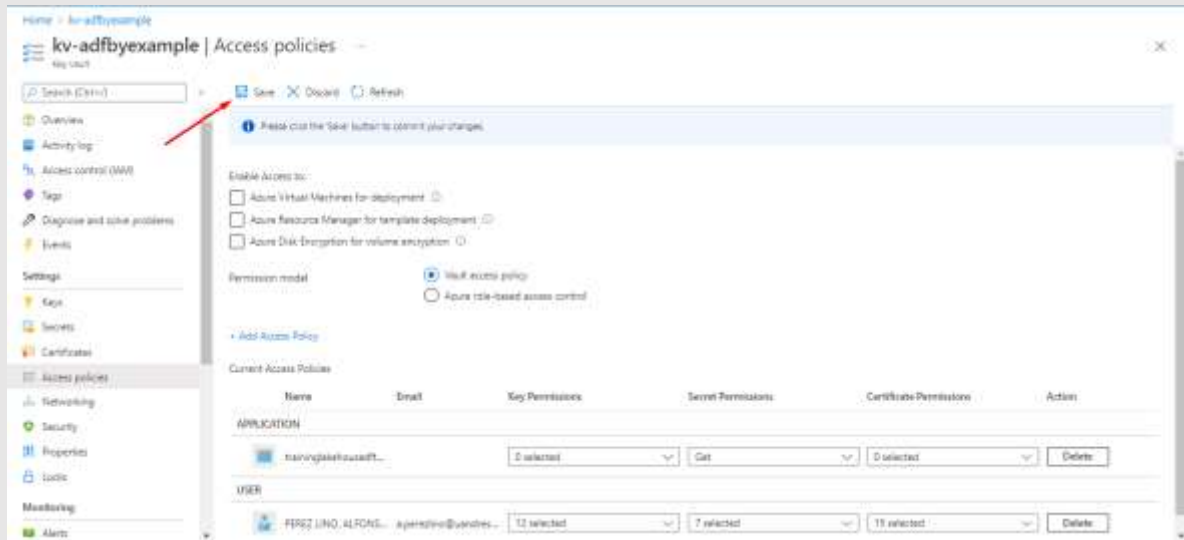


Figura 5-4 Seleccionar la entidad de identidad gestionada por el ADF para la política de acceso al depósito de claves

4. En la hoja de Add access policy, haga clic en Add, y luego en la parte superior de la hoja de políticas de acceso, haga clic en Save.





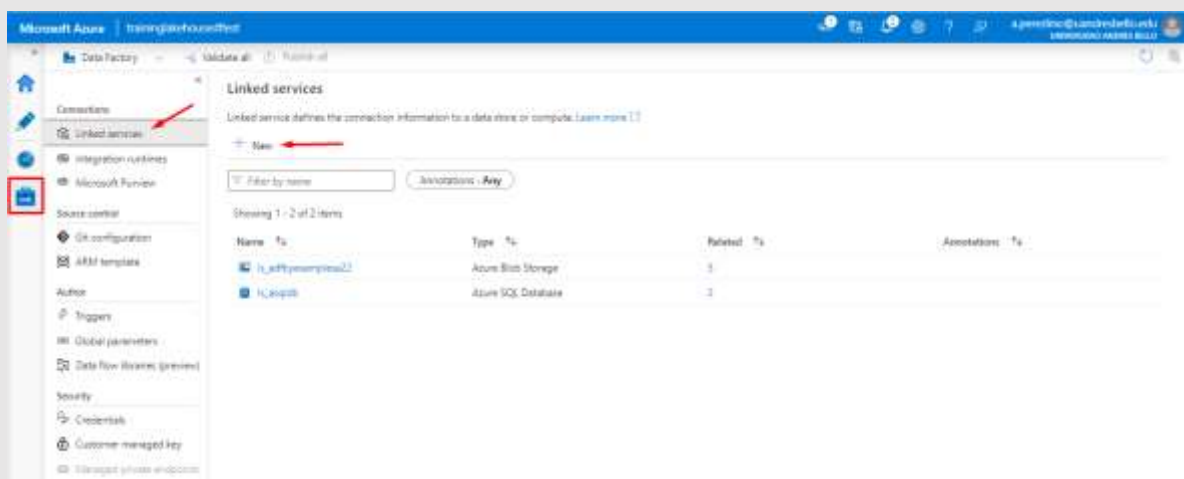


**Nota** Las políticas de acceso también pueden configurarse durante la creación del key vault - el botón Next: Access policy visible en la Figura 5-1 lleva a este paso después de completar los detalles básicos de la bóveda.

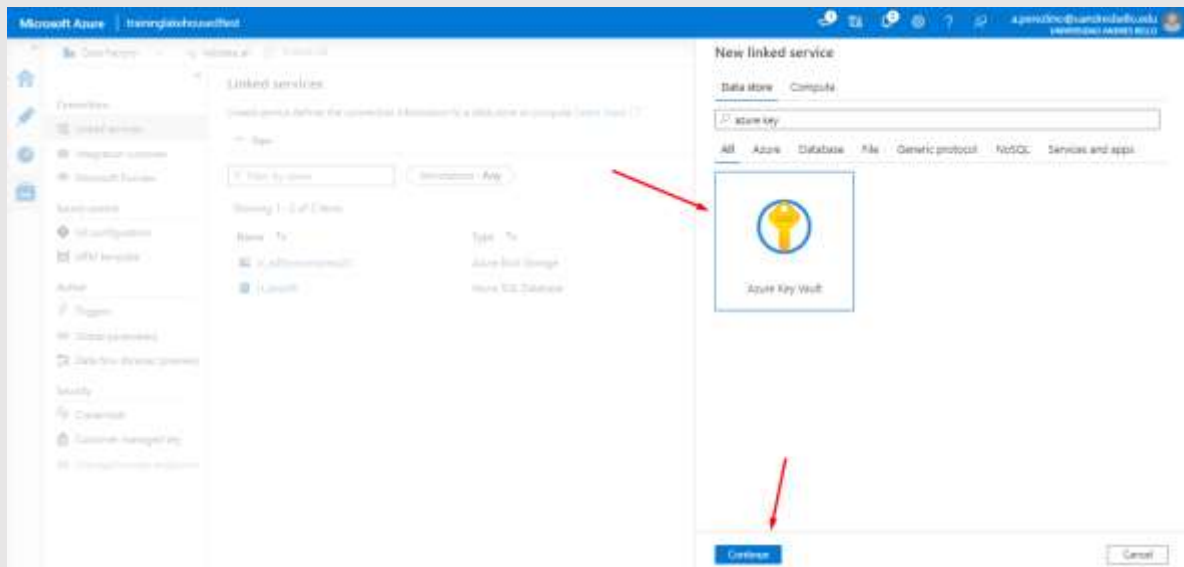
#### 5.1.4. Crear un Key Vault ADF Linked Service

Azure Data Factory accede a un key vault exactamente de la misma manera que a otros tipos de recursos externos: utilizando un servicio vinculado. Para referirse a una key vault desde su data factory, debe crear un servicio vinculado que la represente.

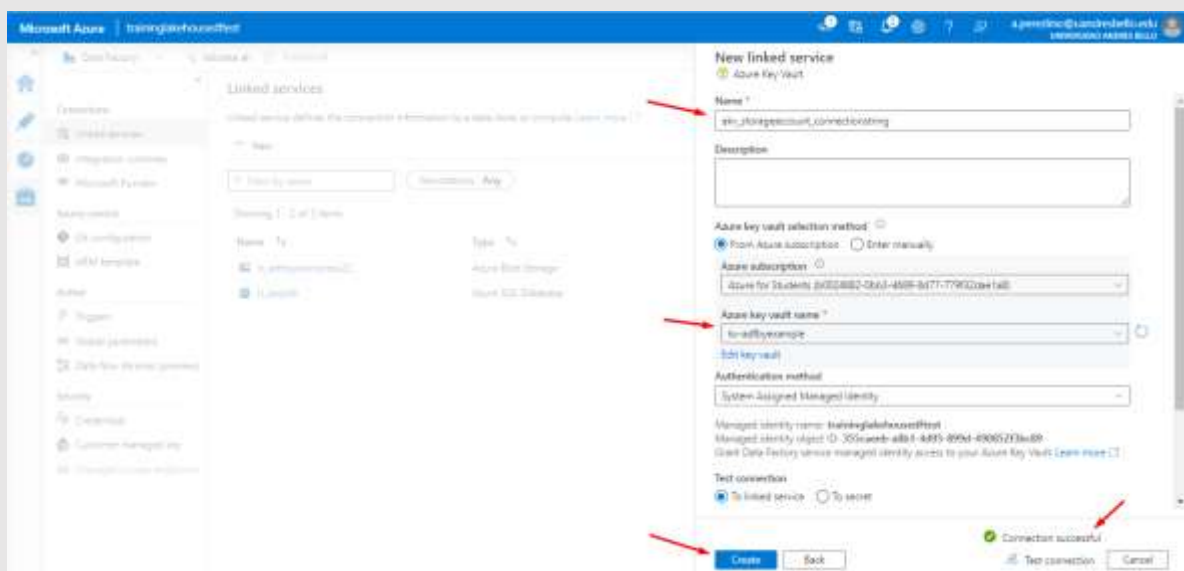
1. Abra el hub de gestión de ADF UX y seleccione Servicios vinculados en la sección Conexiones de su barra lateral.



2. Haga clic en el botón + Nuevo para añadir un nuevo servicio vinculado, luego busque y seleccione el almacén de datos Azure Key Vault. Haz clic en Continuar.



3. En la hoja *New linked service* (Azure Key Vault), proporcione un Nombre para el servicio vinculado de key vault y, a continuación, seleccione su key vault en el menú desplegable Nombre de Azure key vault.



4. La Figura 5-5 muestra la hoja completada. Utilice el botón Probar conexión para comprobar la configuración del servicio vinculado y, cuando lo haga correctamente, haga clic en Crear.

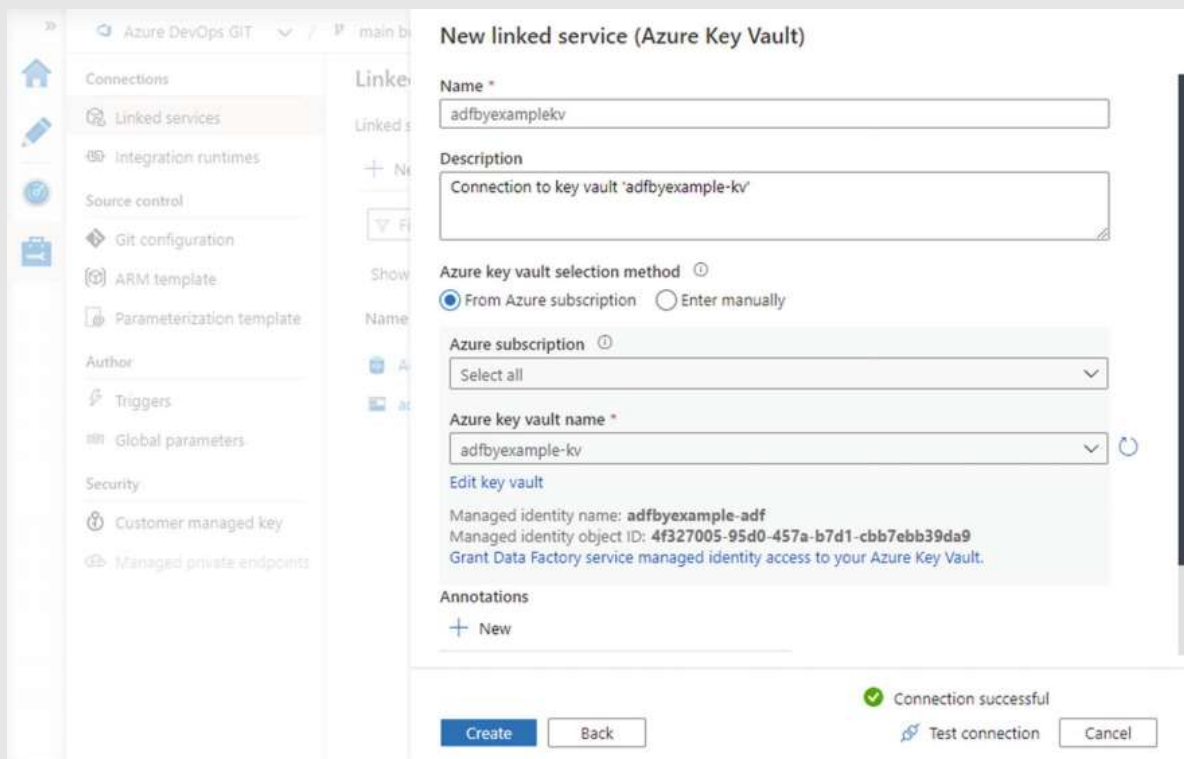
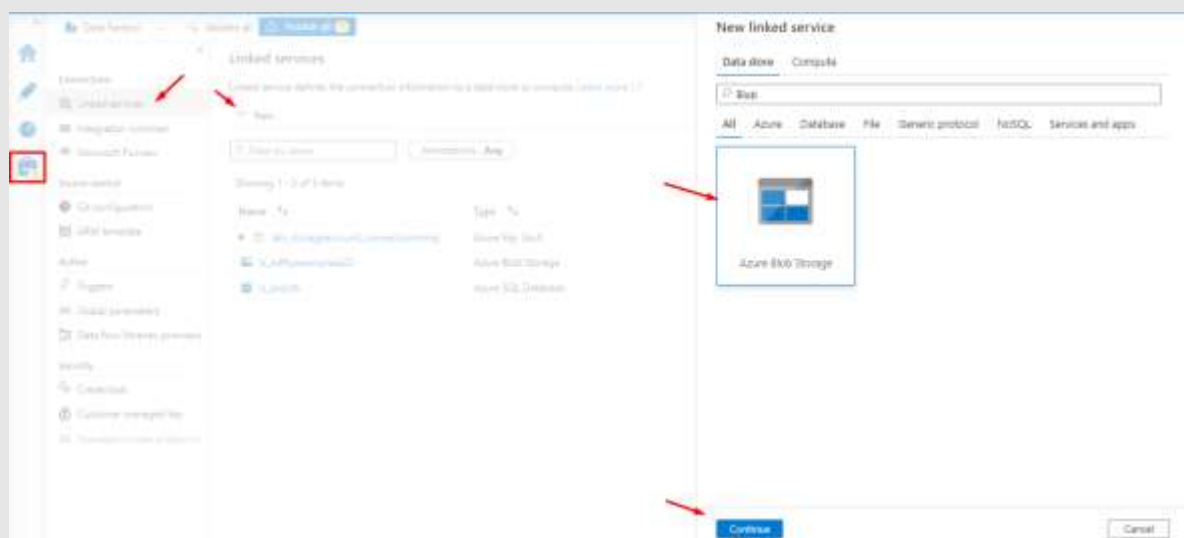


Figura 5-5 Crear un servicio vinculado para Azure Key Vault

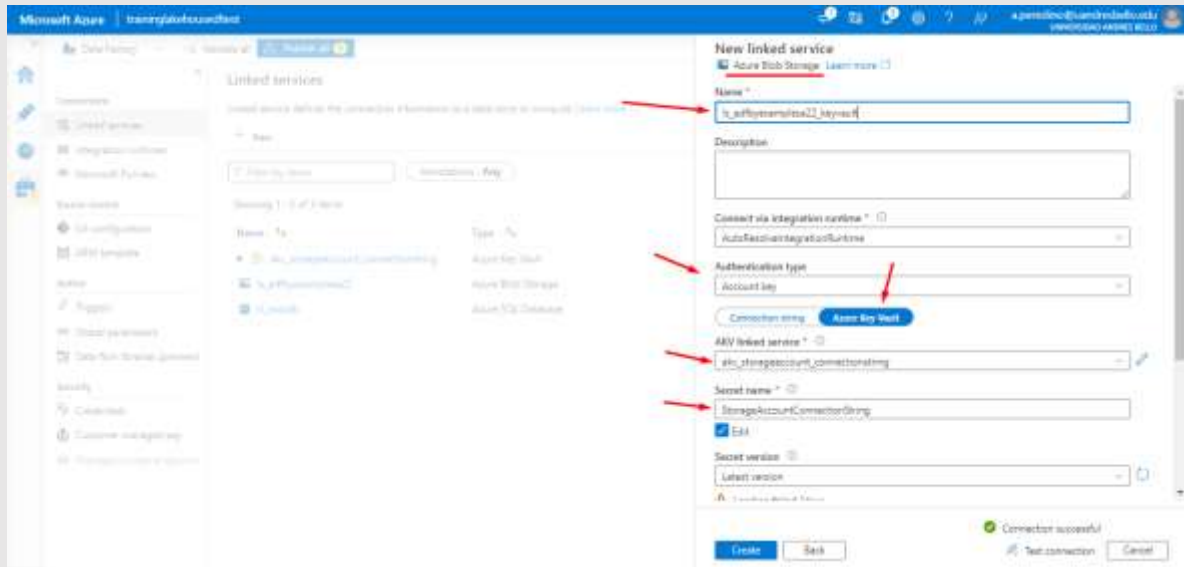
#### 5.1.5. Crear un nuevo servicio vinculado a la cuenta de almacenamiento

Los servicios vinculados de data factory que creaste en los capítulos 2 y 3 se publicaron en ADF inmediatamente, para permitir que las credenciales de conexión asociadas se almacenaran de forma segura. El uso de secretos de key vault es una práctica mucho mejor - **en esta sección, crearás un nuevo servicio vinculado a una cuenta de almacenamiento que obtenga el valor de su cadena de conexión en tiempo de ejecución, haciendo referencia a tu secreto de key vault.**

1. En la misma página de servicios vinculados del hub de gestión, crea otro nuevo servicio vinculado, esta vez utilizando el almacén de datos Azure Blob Storage.



2. Asegúrate de que el método de autenticación esté configurado como "Account key", y luego utiliza el conmutador debajo de ese campo para cambiar el tipo de conexión de "Connection string" a "Azure Key Vault".
3. Seleccione su key vault linked service en el menú desplegable AKV linked service y, a continuación, introduzca el nombre del secreto de la cadena de conexión de su cuenta de almacenamiento.



4. La Figura 5-6 muestra la hoja completada. Utilice el botón de conexión de prueba para comprobar la configuración del servicio vinculado, y cuando tenga éxito, haga clic en Guardar.

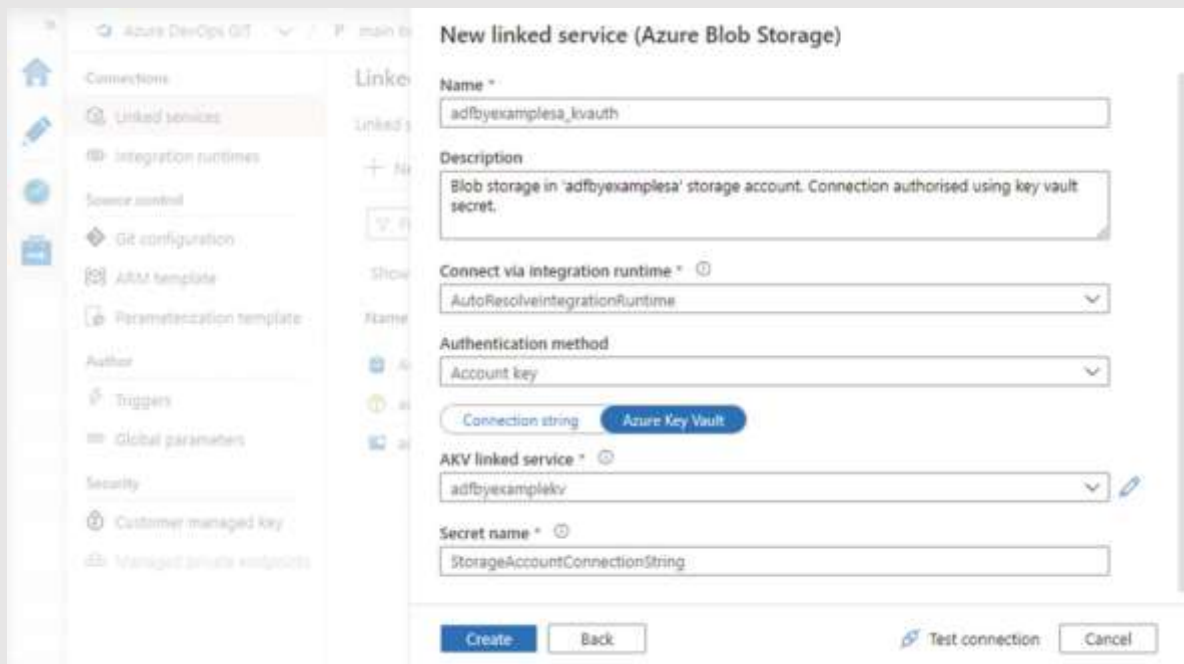


Figura 5-6 Blob storage linked service utilizando un key vault secret

El nuevo servicio vinculado obtiene las credenciales de la key vault en tiempo de ejecución, obteniendo el valor de su secreto nombrado, autorizado mediante ADF instance's managed identity. A diferencia de su anterior servicio vinculado de cuenta de almacenamiento, el nuevo servicio vinculado no ha sido publicado en ADF, porque ya no hay ninguna parte de su definición que requiera un almacenamiento seguro.

## 5.2. Utilizar Dataset Parameters

En el Capítulo 3, usted creó una serie de datasets del ADF, a medida que los necesitaba, para representar diferentes archivos de origen. El dataset "ABS\_CSV\_CandyShack" (mostrado en la Figura 5-7) tiene las siguientes características:

- ❖ Es de tipo DelimitedText, indicado en el espacio del lienzo en la parte superior de la figura. Algunas propiedades mostradas en la pestaña Conexión son específicas de este tipo de archivo, por ejemplo, el Column delimiter y el Row delimiter.
- ❖ Tiene un servicio vinculado específico, una blob storage account.
- ❖ Tiene una ruta de archivo que consiste en un contenedor, un directorio y un archivo. Cada vez que se utiliza un dataset con una ruta de archivo, se anula su configuración de directorio y de archivo mediante la configuración de comodines en la pestaña de configuración de Origen de la actividad Copiar datos.
- ❖ Tiene un esquema de archivo asociado, visible en la pestaña Schema. Cuando utilizó el dataset "ABS\_CSV\_CandyShack", anuló su esquema con los ajustes de la pestaña de configuración de la actividad de copia de datos. Si esto no está claro, mire la pestaña Schema del dataset Candy Shack: contiene el esquema de los datos de Sweet Treats (de cuando clonó el dataset). No coincide con el esquema almacenado en la actividad de copia de datos.

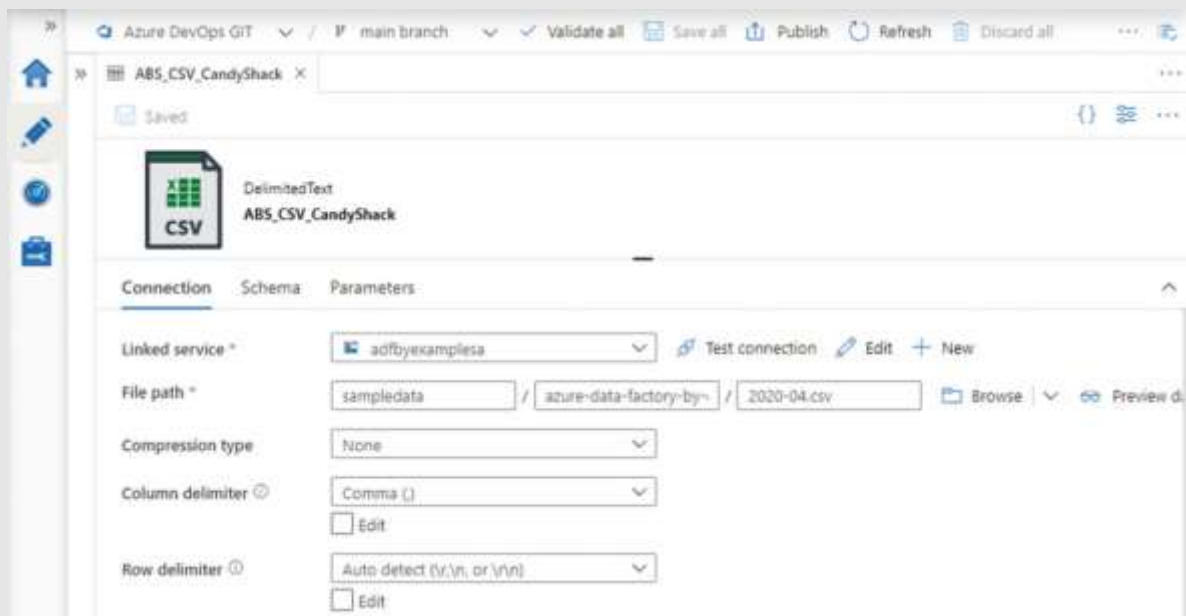


Figura 5-7 Configuración del dataset "ABS\_CSV\_CandyShack".

De hecho, los datasets "ABV\_CSV\_CandyShack" y "ABS\_CSV\_SweetTreats" son equivalentes e intercambiables: son del mismo tipo, utilizan el mismo servicio vinculado y especifican el mismo contenedor. Los valores en los que difieren -la ruta del archivo y el esquema- son proporcionados en tiempo de ejecución por la actividad Copiar datos. Lo mismo ocurre con los dos dataset de archivos JSON.

Los datasets intercambiables son redundantes: pueden ser sustituidos por un único dataset genérico y reutilizable. Los datasets genéricos tienen varias ventajas:

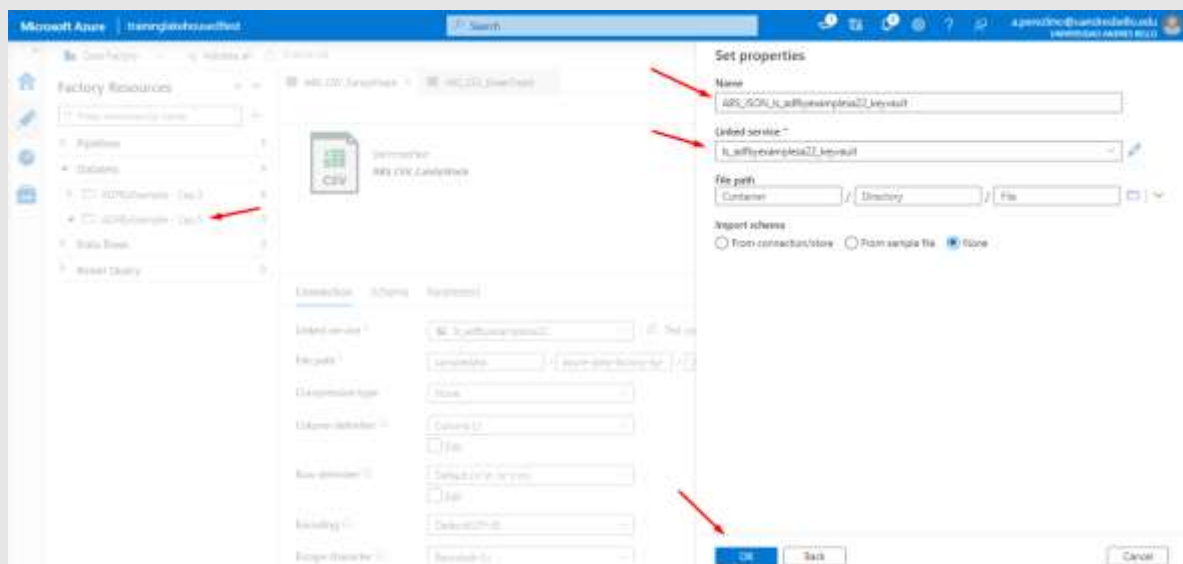
- ❖ Se evita tener que crear múltiples datasets equivalentes que sólo difieren en la ubicación del archivo y el esquema.
- ❖ Un único dataset genérico (que sustituye a los datasets intercambiables) puede etiquetarse claramente como tal. Un dataset específico de la fuente sugiere un acoplamiento más estrecho con su ubicación de origen de lo que realmente es el caso.
- ❖ Su colección de dataset está menos desordenada, lo que la hace más pequeña y fácil de gestionar.

Podría crear un dataset como éste con los conocimientos que ya tiene, pero seguiría necesitando un nombre de contenedor codificado. En la siguiente sección, creará un dataset independiente del archivo utilizando parámetros de dataset para especificar los componentes de la ruta del archivo en tiempo de ejecución.

### 5.2.1. Crear un dataset parametrizado

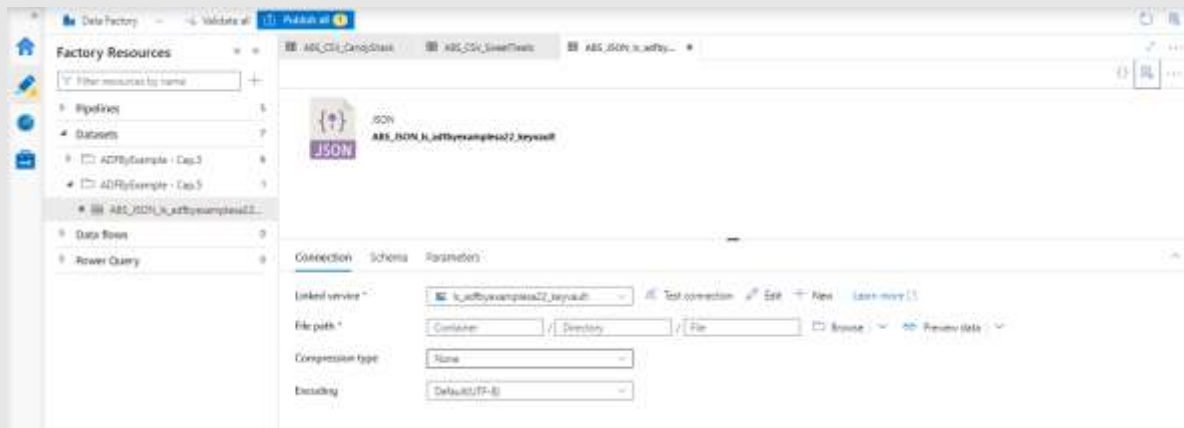
En el ADF UX, cree una nueva carpeta de datasets "Capítulo 5", luego cree un nuevo dataset dentro de ella como sigue:

1. En la hoja Nuevo dataset, elija Azure Blob Storage y, a continuación, Continúe.
2. Seleccione el formato JSON y, a continuación, Continuar.
3. Nombre su dataset "ABS\_JSON\_<newlinkedservice>", donde <newlinkedservice> es el nombre del servicio vinculado que creó en la sección anterior. Seleccione ese servicio vinculado en el desplegable Servicio vinculado.
4. Deje todas las demás opciones con sus valores predeterminados y haga clic en Aceptar para crear el dataset. El dataset y su hoja de Propiedades se abren automáticamente - cierre la hoja de Propiedades.

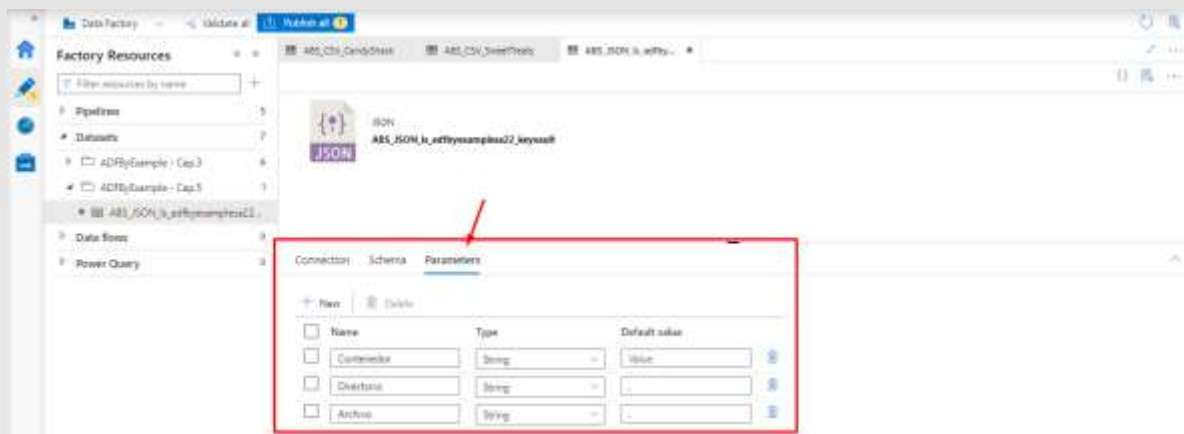


El nuevo dataset aún no se puede utilizar porque no contiene información sobre la ruta del archivo. Para inyectar la información de la ruta del archivo en tiempo de ejecución, cree y utilice los parámetros del dataset como sigue:

1. Seleccione la pestaña de configuración Parámetros del nuevo dataset y haga clic en el botón + Nuevo.



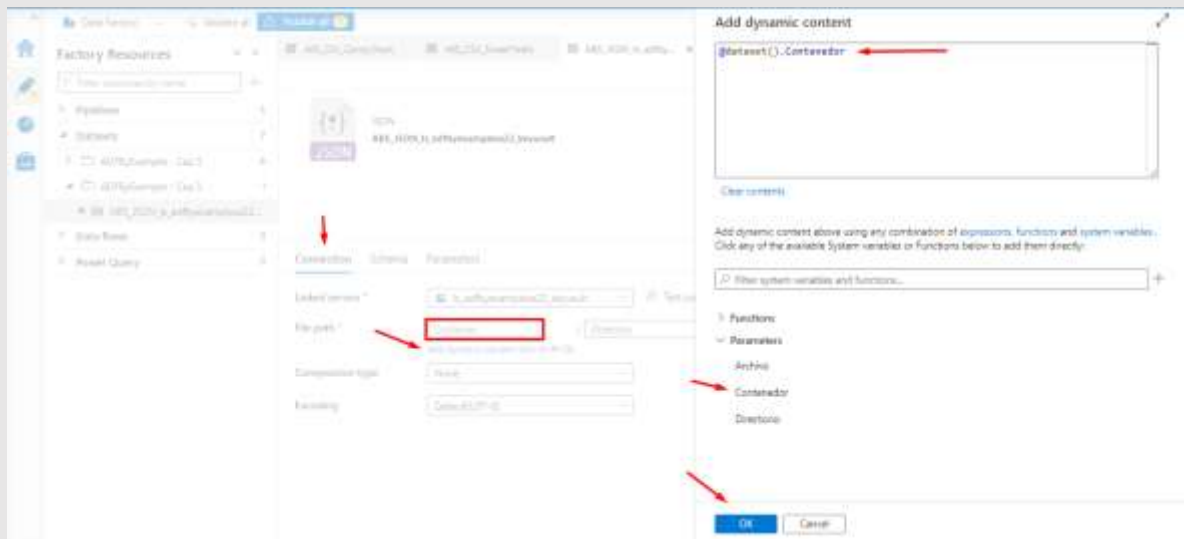
2. Nombre el nuevo parámetro "Contenedor" y asegúrese de que su tipo es "String". Deje su valor por defecto en blanco.
3. Cree otros dos parámetros llamados "Directorio" y "Archivo" de la misma manera, dándole a cada uno un valor por defecto de "." (punto).



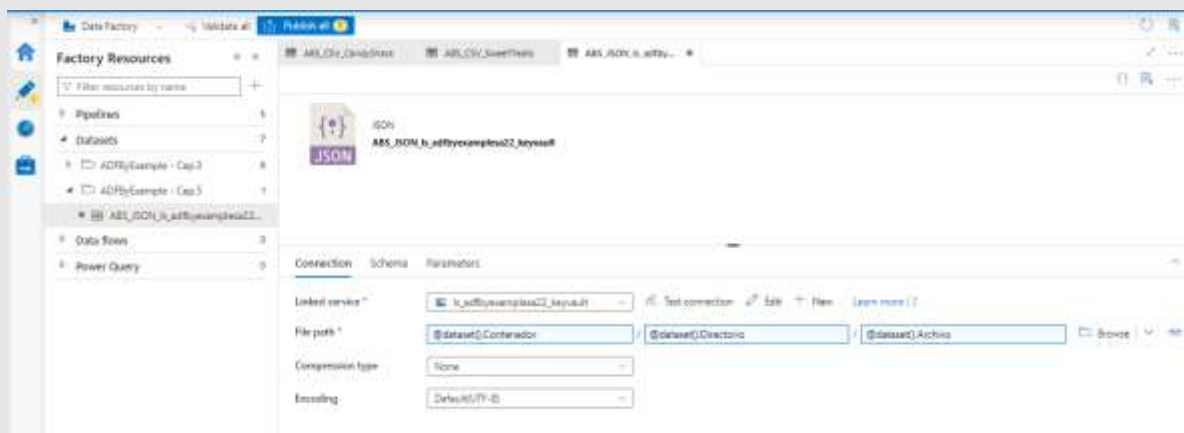
4. Vuelva a la pestaña Conexión y haga clic en el campo Contenedor de la ruta del archivo. Aparecerá el mensaje Añadir contenido dinámico [Alt+Mayúsculas+D]: inicie el constructor de expresiones.
5. Observará que las opciones disponibles para la selección en el constructor de expresiones son diferentes a las que encontró en el Capítulo 4. (Por ejemplo, las variables del sistema relacionadas con el pipeline no están disponibles). En la parte inferior del constructor de expresiones hay una sección de Parámetros - haga clic en el parámetro "Container".



6. La expresión `@dataset().Container` aparece en el panel de expresión. Haga clic en Finalizar.



7. Repita los pasos 3-5 para los campos Directorio y Archivo de la ruta de acceso, utilizando los parámetros correspondientes del dataset. Guarde el dataset.



La Figura 5-8 muestra la pestaña de configuración de la conexión del dataset utilizando sus tres parámetros.

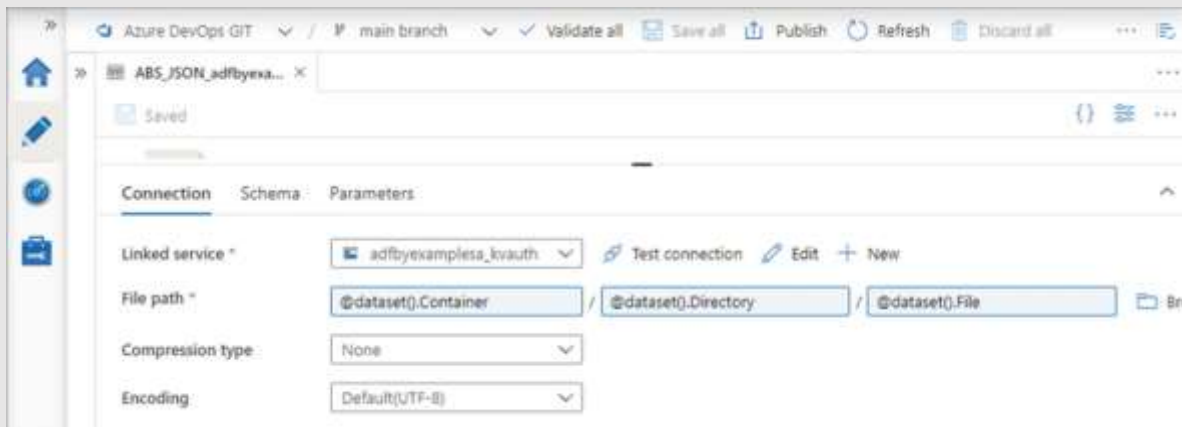


Figura 5-8 Conjunto de datos JSON con nombre de contenedor parametrizado

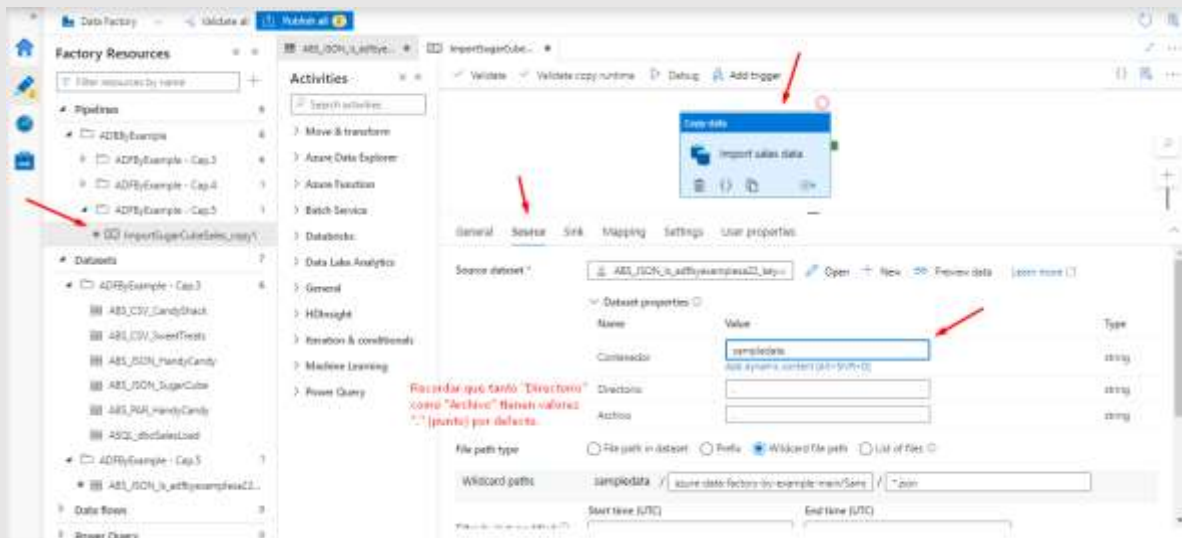
**Nota** La sintaxis `dataset().ParameterName` identifica un parámetro llamado "ParameterName" dentro del contexto del conjunto de datos donde se define el parámetro. El prefijo `dataset()` sólo puede utilizarse dentro de la definición de un dataset y sólo para referirse a los parámetros propios de ese dataset.

### 5.2.2. Utilizar el dataset parametrizado

Cree una nueva carpeta de pipelines "Capítulo 5", luego cree un nuevo pipeline clonando el pipeline "ImportSugarCubeSales" del Capítulo 3. Arrastra el pipeline clonado a la carpeta "Capítulo 5".

El pipeline clonado de Sugar Cube contiene una sola actividad - la actividad de Copiar datos que se utiliza para copiar los datos de los archivos JSON en tu base de datos Azure SQL.

1. Seleccione la actividad Copiar datos y, a continuación, abra su pestaña de configuración de origen.
2. Cambie su dataset de origen de "ABS\_JSON\_SugarCube" a su nuevo dataset parametrizado. Cuando se selecciona un dataset parametrizado, aparece una región de propiedades del dataset debajo del menú desplegable del dataset de origen: aquí es donde se especifican los valores o las expresiones para los parámetros del dataset.
3. Introduzca el valor "sampledata" para el parámetro "Container".



La Figura 5-9 muestra la pestaña de configuración de Origen de la actividad Copiar datos especificando el nombre del contenedor. Ejecute el pipeline y utilice su cliente SQL para verificar que los datos de Sugar Cube se han leído utilizando el dataset parametrizado y se han cargado correctamente.

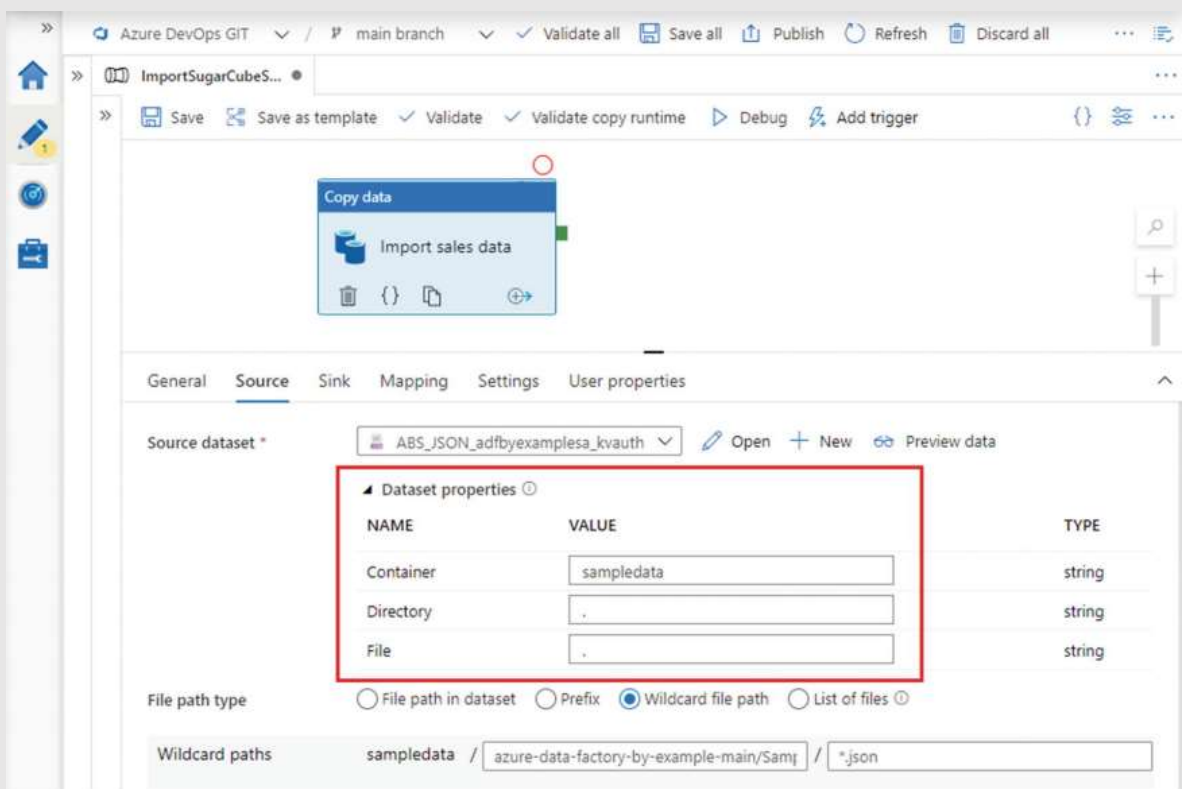
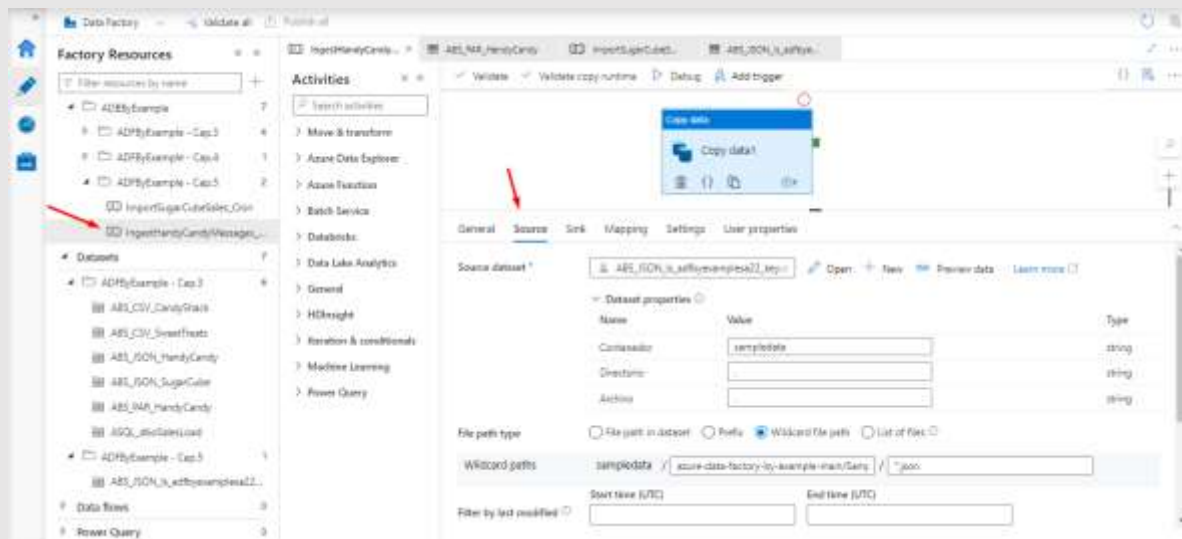


Figura 5-9 Actividad de copia de datos configurada para pasar los parámetros del dataset

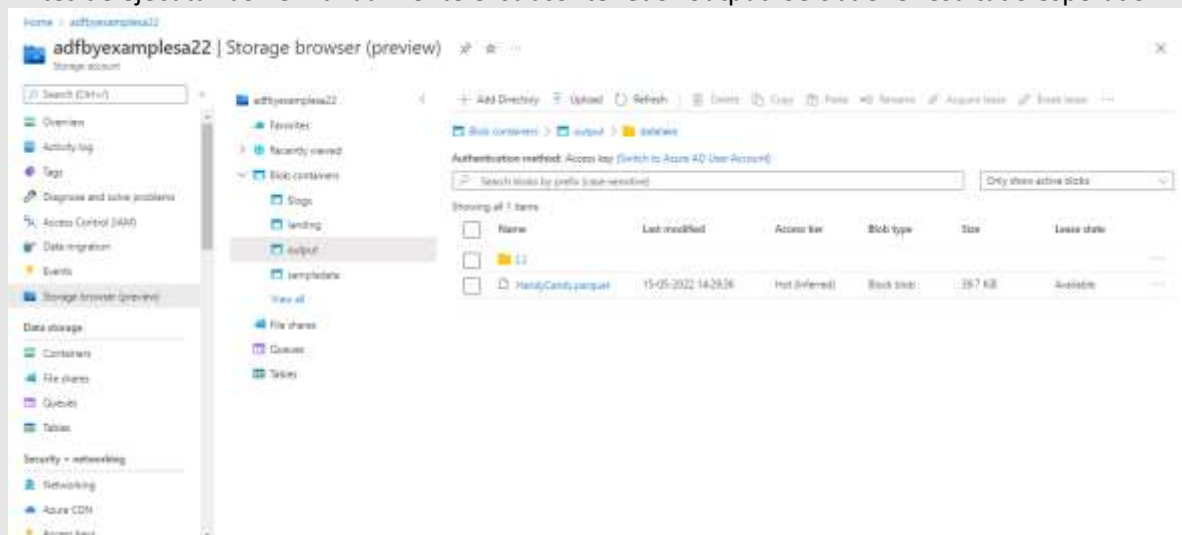
**Consejo** Puede hacer que un parámetro sea opcional dándole un valor por defecto - si no se proporciona ningún valor en tiempo de ejecución, se utilizará el valor por defecto en su lugar. En este ejemplo, los valores predeterminados se anularon porque la actividad de datos de copia especificó una ruta de tipo de archivo de "Ruta de archivo comodín", pero sin ellos, aún tendría que proporcionar los valores de los parámetros para que el pipeline sea válido.

### 5.2.3. Reutilizar el dataset parametrizado

Repita los pasos de la sección anterior, esta vez clonando el pipeline "IngestHandyCandyMessages" del Capítulo 3 en la carpeta "Capítulo5". Ejecute el proceso resultante y compruebe los resultados. (Recordemos que este pipeline transforma los mensajes JSON de origen en un archivo Parquet en el blob storage, por lo que hay que utilizar Azure Storage Explorer para ver su efecto).



Antes de ejecutar borre manualmente el subcontenedor output. Se obtiene resultado esperado.



El principal resultado aquí es que ambos pipelines utilizan ahora el mismo dataset de origen - de hecho, cualquier pipeline que lea datos JSON de su cuenta blob storage podría utilizar este dataset. Esto es una mejora significativa:

- ❖ Se necesitan menos datasets de origen - se podría optar por especificar un único dataset parametrizado para cada combinación de servicio vinculado al blob storage/tipo de archivo.
- ❖ El dataset de origen parametrizado es genérico por diseño. Cualquiera que vea el dataset en el futuro no corre el riesgo de creer que el dataset está acoplado a una ubicación de archivo específica.

Dado que todos los datos de origen están en el contenedor "sampledata", sería razonable preguntarse por qué parametrizar el contenedor del dataset. La forma en que se organizan los datos de origen puede ser una consideración en este caso:

- ❖ Si sus datos de origen están en varios contenedores, un dataset parametrizado proporciona una útil reutilización.
- ❖ Si sus datos de origen están garantizados para ocupar siempre un único contenedor, entonces un dataset con un nombre de contenedor codificado le libera de proporcionar un valor de parámetro cada vez que lo utilice.

Si sus datos de origen se encuentran en múltiples storage accounts, es posible que desee ir un paso más allá e implementar un dataset genérico por tipo de archivo. Esto requeriría que los detalles de conexión del servicio vinculado se inyectaran en tiempo de ejecución, lo cual es el tema de la siguiente sección.

### 5.3. Utilizar Linked Service Parameters

Al parametrizar el dataset en la sección anterior, se ha incrementado su reutilización, ya que ahora puede utilizarse para referirse a cualquier archivo JSON en la cuenta de almacenamiento subyacente. Un dataset que pudiera referirse a cualquier archivo JSON en cualquier storage account sería aún más reutilizable, pero requeriría que la cuenta de almacenamiento también se especificara en tiempo de ejecución.

La storage account utilizada por un dataset se especifica en el servicio vinculado que utiliza y puede proporcionarse en tiempo de ejecución mediante un parámetro del servicio vinculado. Pasar una cadena de conexión de cuenta de almacenamiento (storage account connection string) como valor de parámetro sería posible, pero inseguro. Configurar un servicio vinculado para que utilice un secreto de key vault significa que se puede conseguir el mismo efecto de forma segura, permitiendo que el nombre del secreto se pase como parámetro.

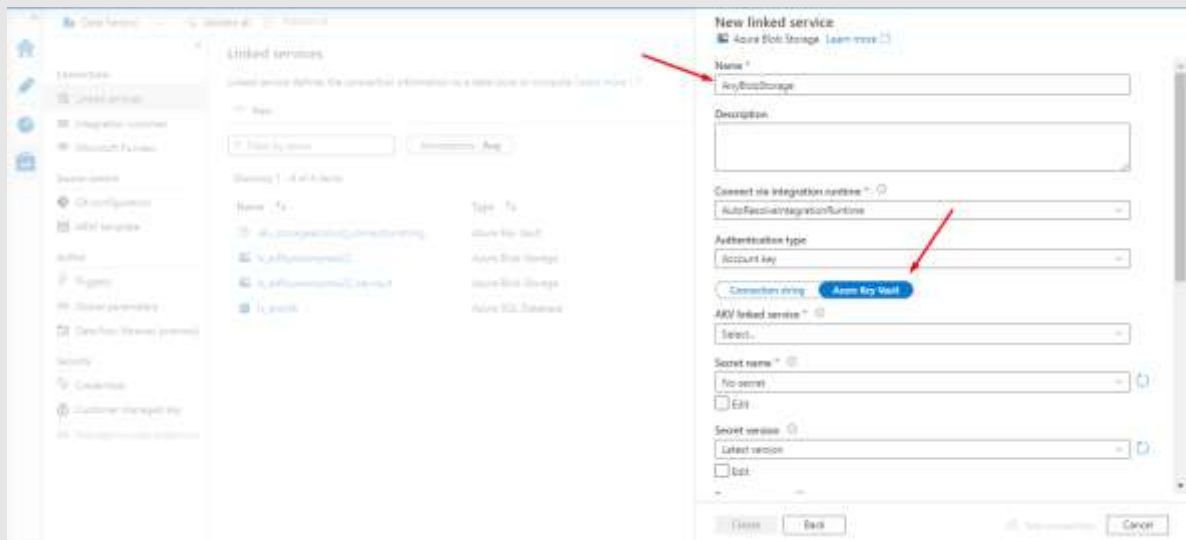
#### 5.3.1. Crear un linked service parametrizado

En la primera parte de este capítulo, has creado un key vault y has añadido un secreto: la cadena de conexión de tu cuenta de almacenamiento. Creaste un nuevo servicio vinculado a la cuenta de almacenamiento, utilizando el key vault secret name correspondiente para acceder al valor de la cadena de conexión en tiempo de ejecución. En esta sección, crearás un servicio vinculado parametrizado para blob storage accounts, permitiéndote pasar diferentes secret names en tiempo de ejecución - esto permitirá que el servicio vinculado sea usado para conectarse a diferentes cuentas de almacenamiento.

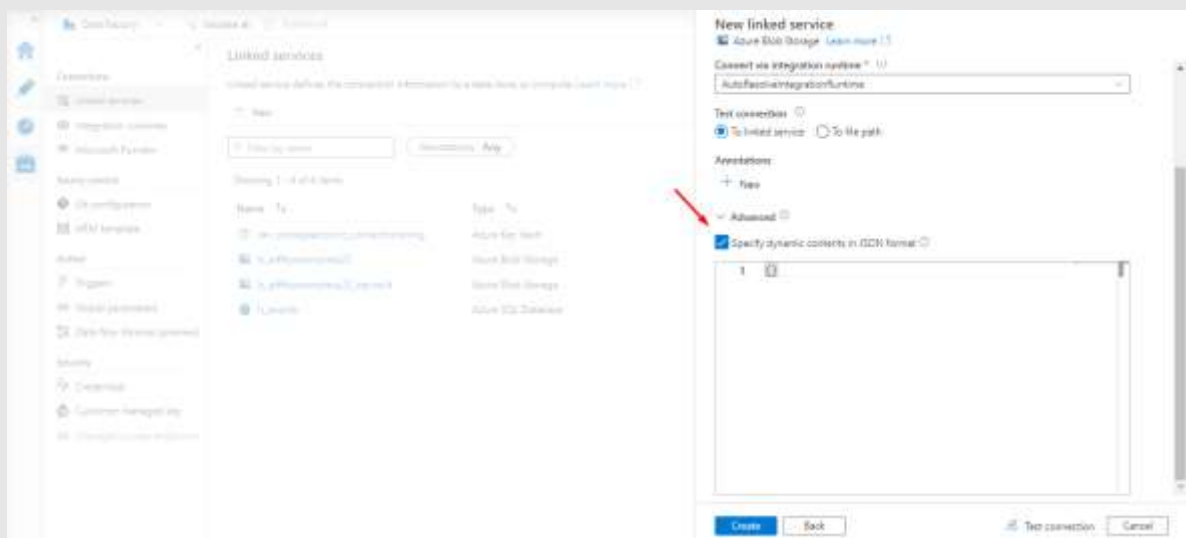
Es posible que ya hayas notado que el editor de servicios vinculados carece de algunas de las características que has estado utilizando para las propiedades dinámicas. Los campos de entrada para definir los parámetros no siempre están presentes, y al editar las propiedades puede que no se le ofrezca un enlace para lanzar el constructor de expresiones. Esto no significa que los servicios vinculados carezcan de soporte para parámetros o expresiones, sólo que no siempre son soportados por la UX de ADF.

Recuerde que todas las definiciones de recursos de ADF se almacenan como archivos JSON - puede ver estos archivos en su repositorio Git. Los parámetros y expresiones de los servicios vinculados pueden configurarse especificándolos directamente en la definición JSON de un servicio vinculado.

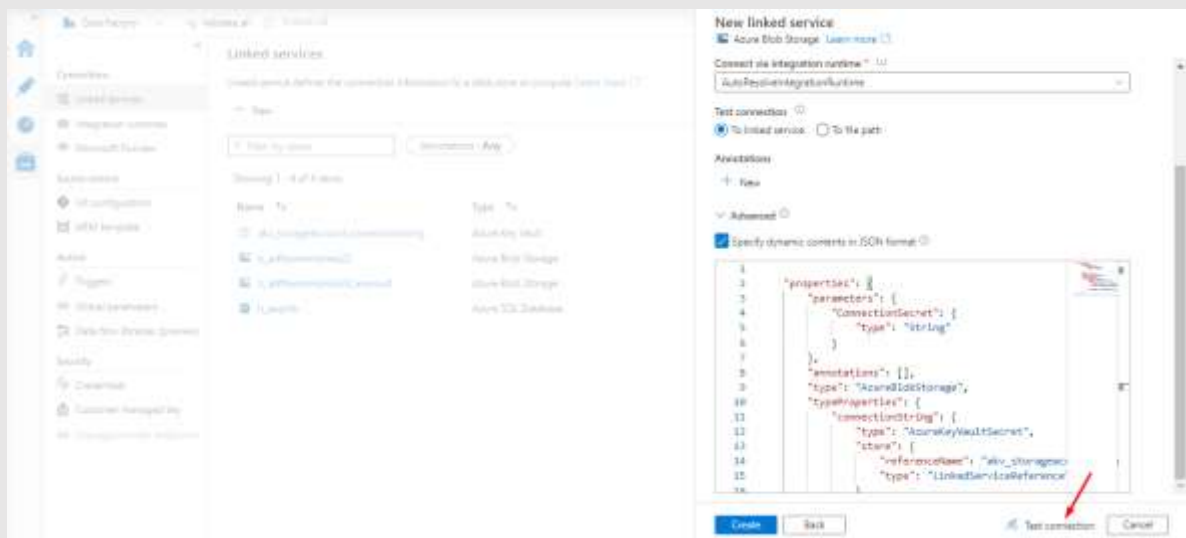
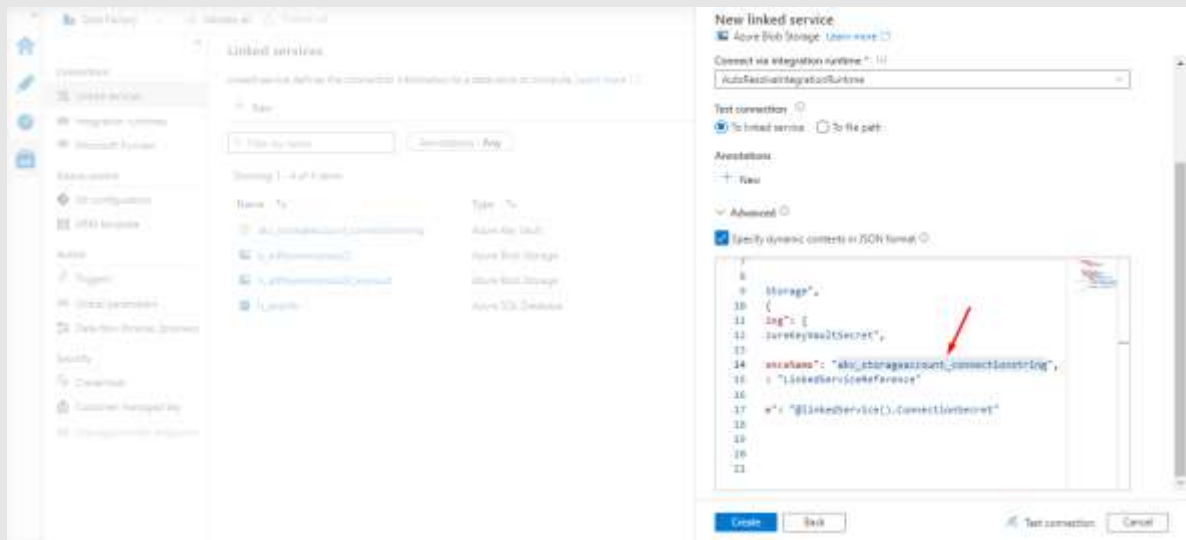
1. Abra el hub de gestión de ADF UX y seleccione Servicios vinculados en la sección Conexiones de su barra lateral.
2. Haga clic en el botón + Nuevo para añadir un nuevo servicio vinculado, luego seleccione el almacén de datos Azure Blob Storage y haga clic en Continuar.
3. Llame al nuevo servicio vinculado "AnyBlobStorage" - podría utilizarse para conectarse a cualquier cuenta de almacenamiento de Azure.
4. Usando el botón debajo del desplegable Authentication type, cambie el tipo de conexión a Azure Key Vault.



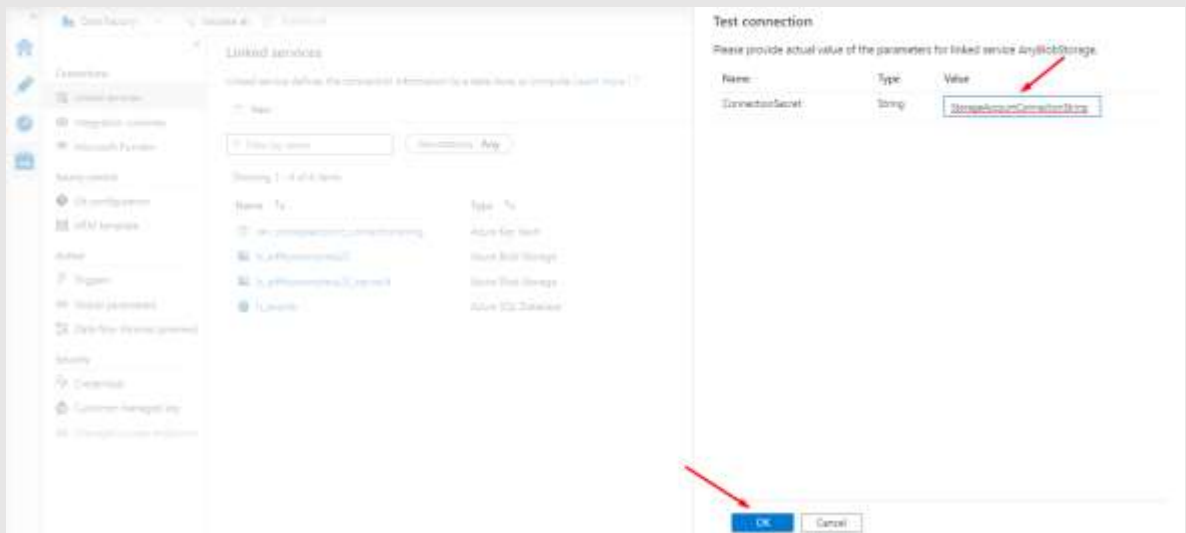
- Desplácese hacia abajo hasta la parte inferior de la hoja New linked service y expanda la sección Avanzado. Marque la casilla Especificar contenidos dinámicos en formato JSON.



- Debajo de la casilla de verificación Especificar contenidos dinámicos en formato JSON hay un área de entrada de texto. Sustituya su contenido por defecto "{}" por el objeto JSON mostrado en el Listado 5-1. Tendrá que actualizar el código con el nombre de su propio key vault linked service (coloque el linked service vinculado al Key Vault: **akv\_storageaccount\_connectionstring**, dado que el ejemplo del libro hizo lo mismo, **adfbexamplekv** representa lo mismo), sustituyendo adfbexamplekv en el listado. (La ubicación del valor se indica en la Figura 5-10).

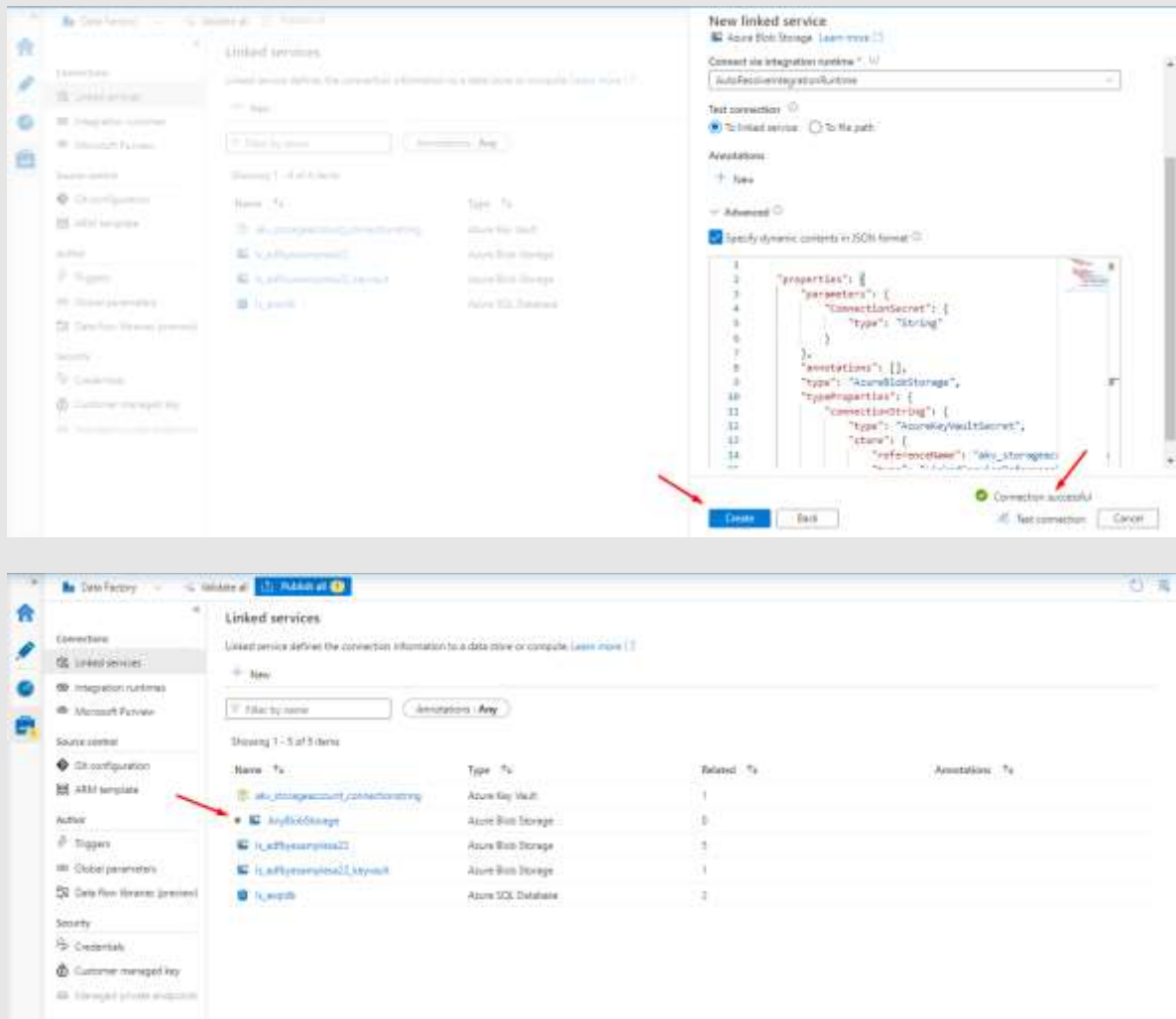


- Haga clic en Probar conexión en la parte inferior de la hoja. La UX del ADF le pide ahora que proporcione un valor para el parámetro "ConnectionSecret" - introduzca el nombre del secreto del key vault que contiene la cadena de conexión de su cuenta de almacenamiento, y haga clic en OK.





8. La Figura 5-10 muestra la hoja completada después de una prueba de conexión exitosa. Cuando la prueba se haya completado con éxito, haga clic en Crear para guardar el nuevo servicio vinculado.



**New linked service (Azure Blob Storage)**

Name \*  
AnyBlobStorage

Description  
Connect to any blob storage account with a connection string in the key vault

Connect via integration runtime \*  
AutoResolveIntegrationRuntime

Test connection  
☒ To linked service
 ☐ To file path

Annotations  
 + New

Name

Advanced  
☒ Specify dynamic contents in JSON format

```

1 {
2   "properties": {
3     "parameters": {
4       "ConnectionSecret": {
5         "type": "String"
6       }
7     },
8     "annotations": [],
9     "type": "AzureBlobStorage",
10    "typeProperties": {
11      "connectionString": {
12        "type": "AzureKeyVaultSecret",
13        "store": {
14          "referenceName": "adfbyexamplekv",
15          "type": "LinkedServiceReference"
16        }
17      }
18    }
19  }
20 }
  
```

Connection successful

Create Back Test connection Cancel

Figure 5-10 Linked service JSON configuration indicating the key vault linked service name

```

{
  "properties": {
    "parameters": {
      "ConnectionSecret": {
        "type": "String"
      }
    },
    "annotations": [],
    "type": "AzureBlobStorage",
    "typeProperties": {
      "connectionString": {
        "type": "AzureKeyVaultSecret",
        "store": {
          "referenceName": "adfbyexamplekv",
          "type": "LinkedServiceReference"
        }
      }
    }
  }
}
  
```

```

    },
    "secretName": "@linkedService().ConnectionSecret"
  }
}
}
}

```

Listado 5-1 Definición de servicio vinculado dinámico en JSON

Observa detenidamente el Listado 5-1 para entender sus componentes. Consiste en un objeto JSON con un único campo de propiedades que contiene:

- ❖ Un campo **parameters**, que define un único parámetro string llamado ConnectionSecret
- ❖ Un campo **type**, que indica que este servicio vinculado es una conexión de blob storage
- ❖ Un campo **typeProperties** que especifica la cadena de conexión de blob storage

La cadena de conexión del blob storage también tiene un **type**, que indica que el valor de la cadena de conexión se almacena externamente en un secreto de key vault. **store** identifica el key vault (utilizando su linked service connection name), y secretName especifica el secreto de en el vault. El valor de secretName es referenciado en una expresión:

```
@linkedService().ConnectionSecret
```

que hace referencia al parámetro ConnectionSecret del servicio vinculado, definido en el campo de parámetros del servicio descrito anteriormente. La definición del parámetro y su uso en la expresión es lo que permite inyectar la key vault secret name en tiempo de ejecución, desde fuera de la definición del servicio vinculado.

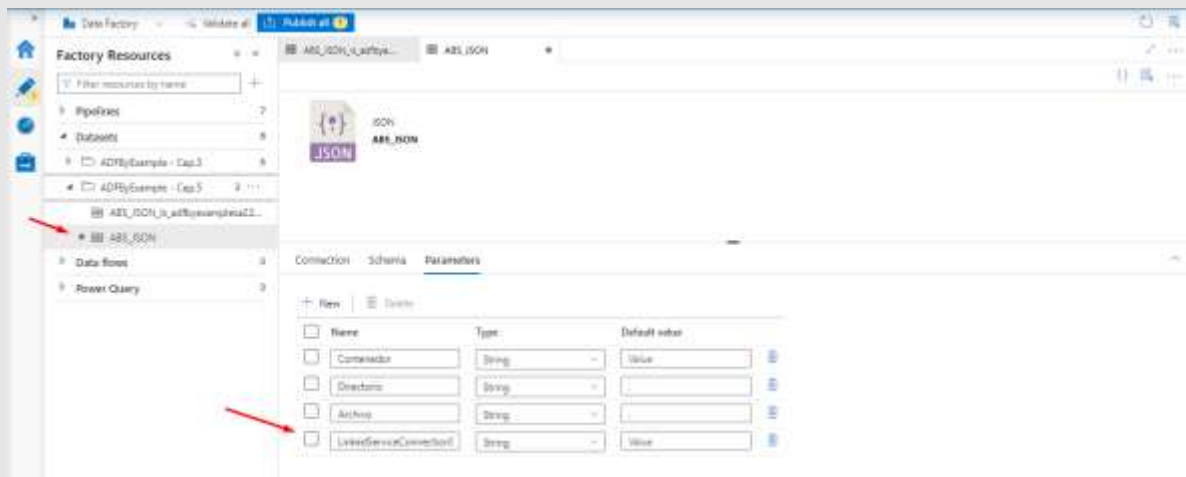
Nota Al igual que en el caso de los datasets, la sintaxis linkedService() sólo puede utilizarse dentro de una definición de servicio vinculado y únicamente para referirse a los parámetros propios del servicio vinculado.

### 5.3.2. Aumentar la reutilización del dataset

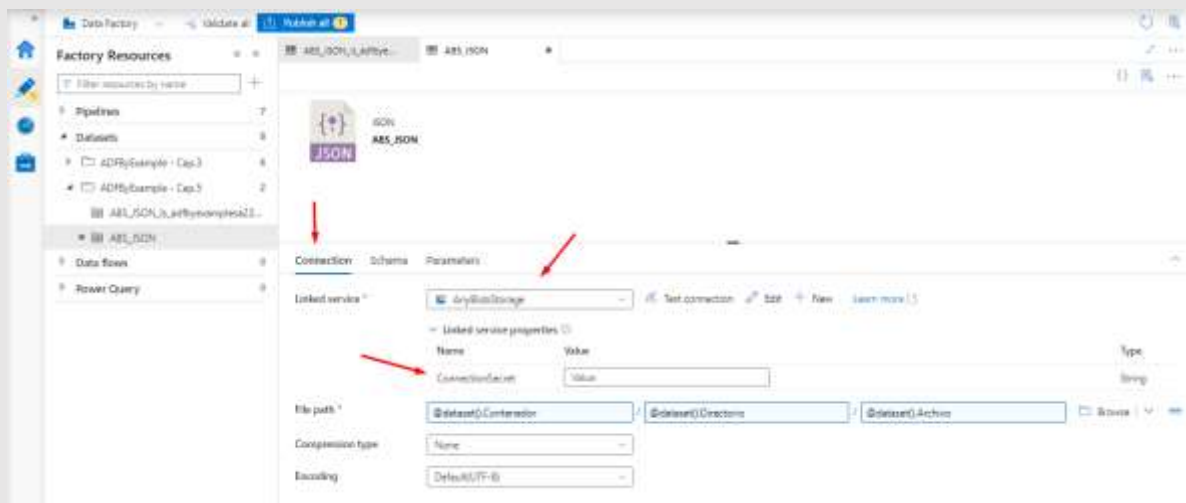
Utilice su servicio vinculado parametrizado para implementar un conjunto de datos con mayor reutilización de la siguiente manera:

1. Vuelva al espacio de trabajo de creación y cree un nuevo conjunto de datos clonando el conjunto de datos parametrizado de este capítulo. Nómbrelo "ABS\_JSON" - podrá representar cualquier archivo JSON en cualquier cuenta de Azure blob storage.

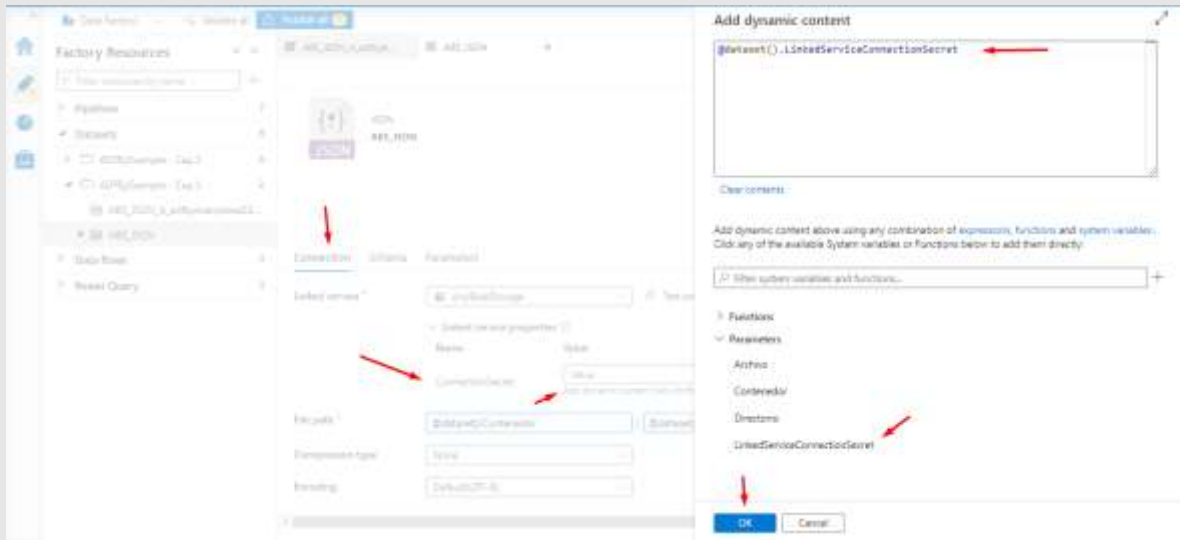
2. En la pestaña de configuración de parámetros del conjunto de datos clonado, añade un cuarto String parameter llamado "LinkedServiceConnectionSecret".



3. En la pestaña de configuración de la conexión del conjunto de datos, elija el servicio vinculado "AnyBlobStorage" del menú desplegable Servicio vinculado.
4. Al seleccionar un servicio vinculado parametrizado se muestra una región de propiedades de servicio vinculado debajo del desplegable de servicio vinculado. Rellene el parámetro "ConnectionSecret" del servicio vinculado con una expresión que devuelva el valor del parámetro "LinkedServiceConnectionSecret" del conjunto de datos.



5. Pruebe el valor de su parámetro de servicio vinculado haciendo clic en Probar conexión (a la derecha del menú desplegable Servicio vinculado). El ADF UX le pedirá un valor para "LinkedServiceConnectionSecret" - proporcione el nombre del secreto del key vault que contiene la cadena de conexión de su cuenta de almacenamiento. Cuando la prueba se haya completado con éxito, guarde los cambios.



La Figura 5-11 muestra la configuración del nuevo dataset parametrizado que utiliza el servicio vinculado parametrizado. Este dataset puede utilizarse para hacer referencia a cualquier archivo JSON en cualquier cuenta de Azure blob storage para la que se haya almacenado una cadena de conexión en su key vault.

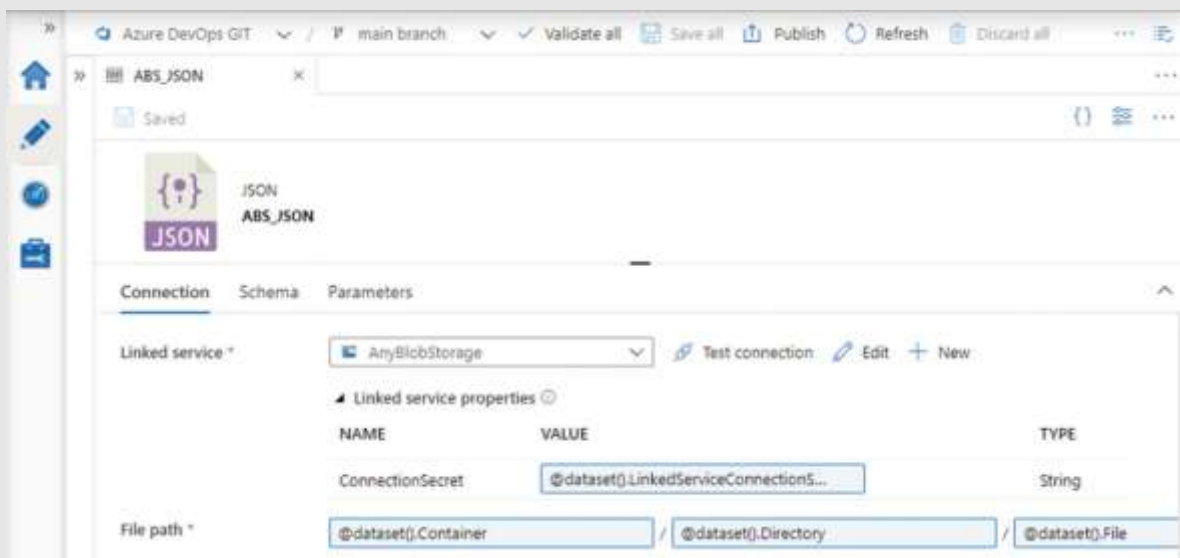


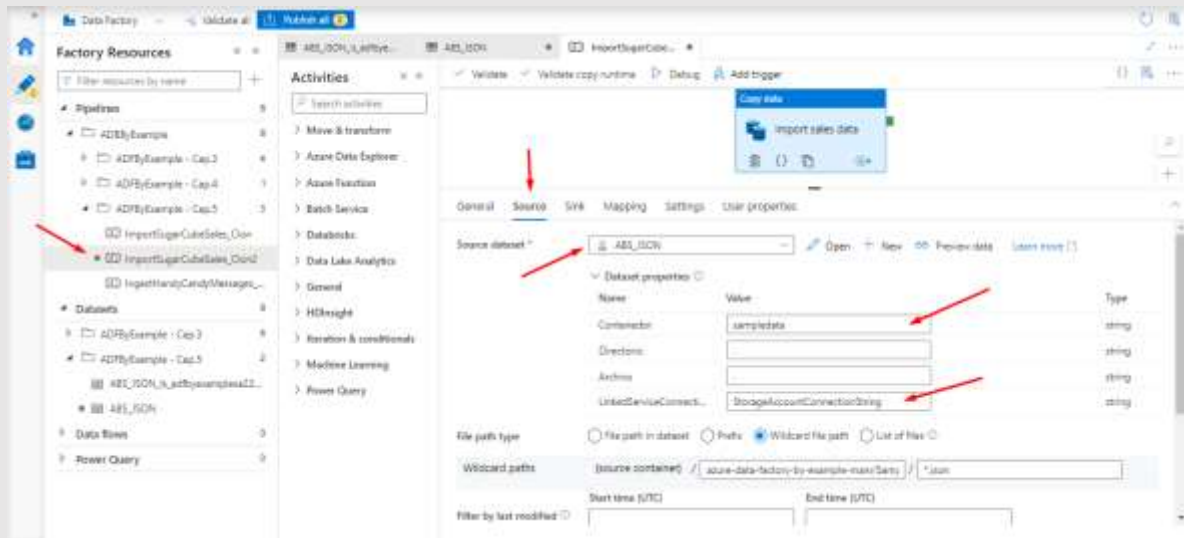
Figura 5-11 Conjunto de datos parametrizado utilizando un servicio vinculado parametrizado

### 5.3.3. Utilice el nuevo dataset

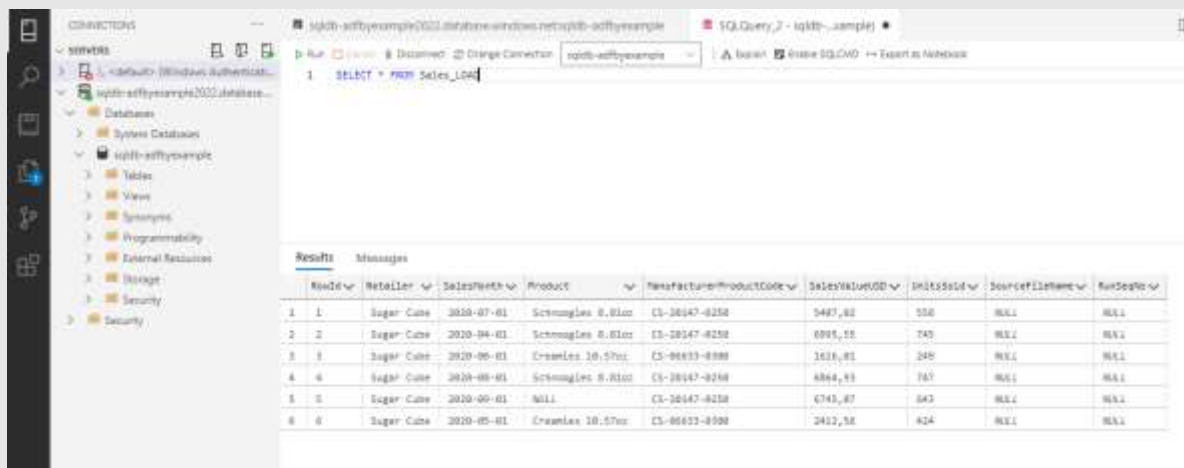
Para probar el nuevo dataset, clona el pipeline de Sugar Cube que creaste anteriormente en el capítulo. Cambie el dataset de origen de la actividad Copiar datos a "ABS\_JSON" y proporcione los dos valores de parámetros obligatorios:

- ❖ Establece "ContainerName" con el valor "sampledata".

- ❖ Establezca "LinkedServiceConnectionSecret" con el nombre del secreto de la key vault que contiene la cadena de conexión de su cuenta de almacenamiento.



Verifique los resultados utilizando su cliente SQL y/o la pestaña Output en el panel de configuración del pipeline.



Verifique los resultados utilizando su cliente SQL y/o la pestaña Salida del panel de configuración del pipeline.

#### 5.3.4. ¿Por qué parametrizar los servicios vinculados?

La parametrización de los datasets de ADF es una poderosa forma de evitar la creación de datasets específicos para cada archivo, permitiendo reutilizar un único dataset para archivos del mismo tipo en diferentes ubicaciones de carpetas. Del mismo modo, un servicio vinculado parametrizado puede reutilizarse para diferentes almacenes de datos del mismo tipo de almacenamiento.

Una razón para hacer esto podría ser reducir el número de servicios vinculados necesarios para conectarse a múltiples almacenes del mismo tipo - por ejemplo, múltiples cuentas de blob storage. Otra razón importante es que es probable que su inquilino de Azure contenga varias instancias de Azure Data Factory para el desarrollo, las pruebas y la producción. La parametrización de los servicios vinculados mediante un key vault le permite centralizar la información de configuración fuera de su instancia de Data Factory, reduciendo la complejidad de copiar los recursos de la fábrica entre entornos. La publicación de los recursos de ADF y la gestión de múltiples entornos se tratan en mayor profundidad en el Capítulo 10.

## 5.4. Utilizar Pipeline Parameters

Los archivos de datos de ejemplo que subió a su cuenta de blob storage en el capítulo 2 incluyen datos de ventas de otro cliente del fabricante ficticio de confitería ABC: un minorista en línea llamado "Desserts4All". Desserts4All informa de las ventas de los productos de ABC en archivos CSV, utilizando exactamente el mismo formato que Sweet Treats.

Podría reutilizar su pipeline de Sweet Treats para cargar los datos de Desserts4All, pero la información de la ubicación de los archivos para el pipeline - la ruta de la carpeta Wildcard y el nombre del archivo Wildcard - están codificados en la actividad Copiar datos. En esta sección, utilizará los parámetros del pipeline para especificar sus valores en tiempo de ejecución, creando un único pipeline que puede procesar datos de ventas de cualquier proveedor.

### 5.4.1. Crear un pipeline parametrizado

Comience clonando su pipeline del Capítulo 4 - "ImportSweetTreatsSales\_Audit" - en la carpeta "Capítulo 5". Nombre el nuevo pipeline "ImportSTFormatFolder".

1. Debajo del lienzo de creación, en el panel de configuración del pipeline, encontrará una pestaña de Parámetros - selecciónela.
2. Utilice el botón + Nuevo para crear un nuevo parámetro de tipo "String". Nómbrelo "WildcardFolderPath". Repita el proceso para crear un parámetro de tipo "String" llamado "WildcardFileName", dándole un valor por defecto de "\*.csv". La Figura 5-12 muestra la pestaña de Parámetros completada.

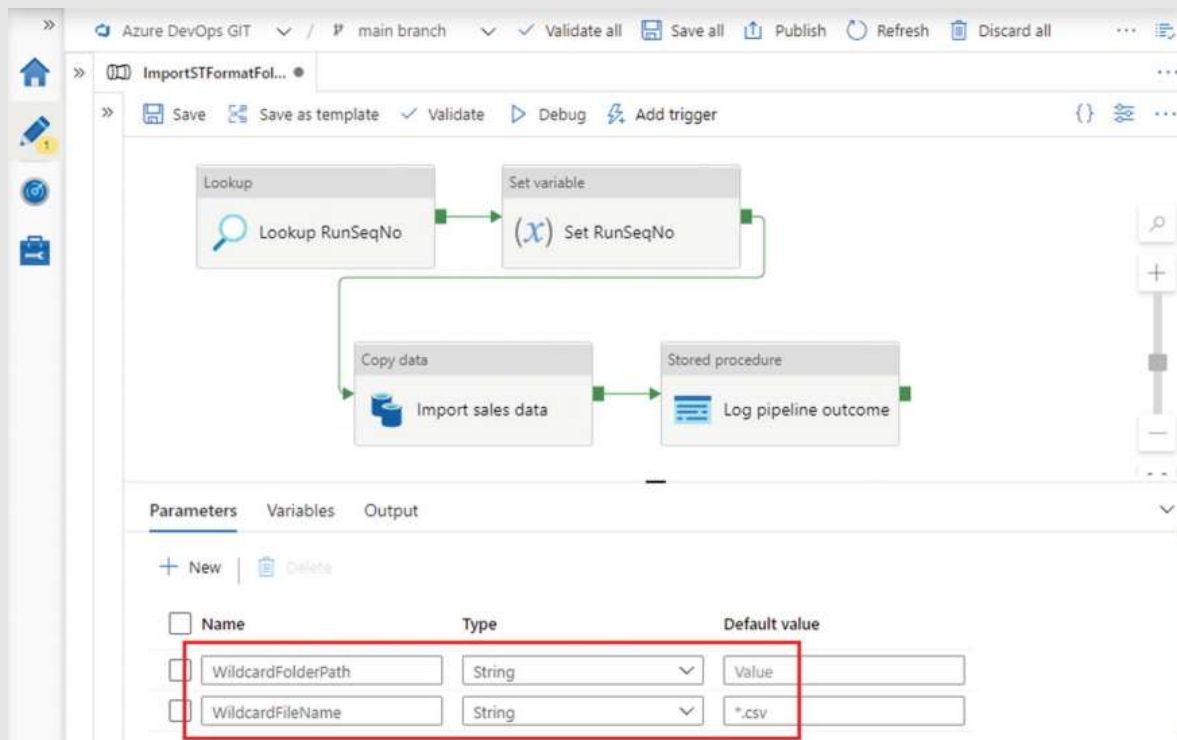
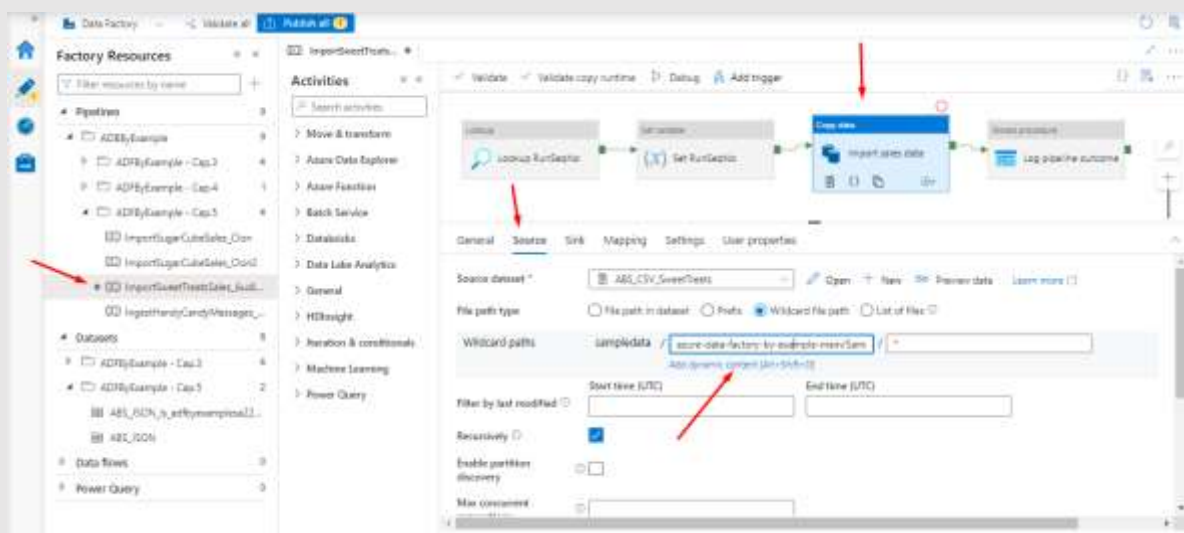
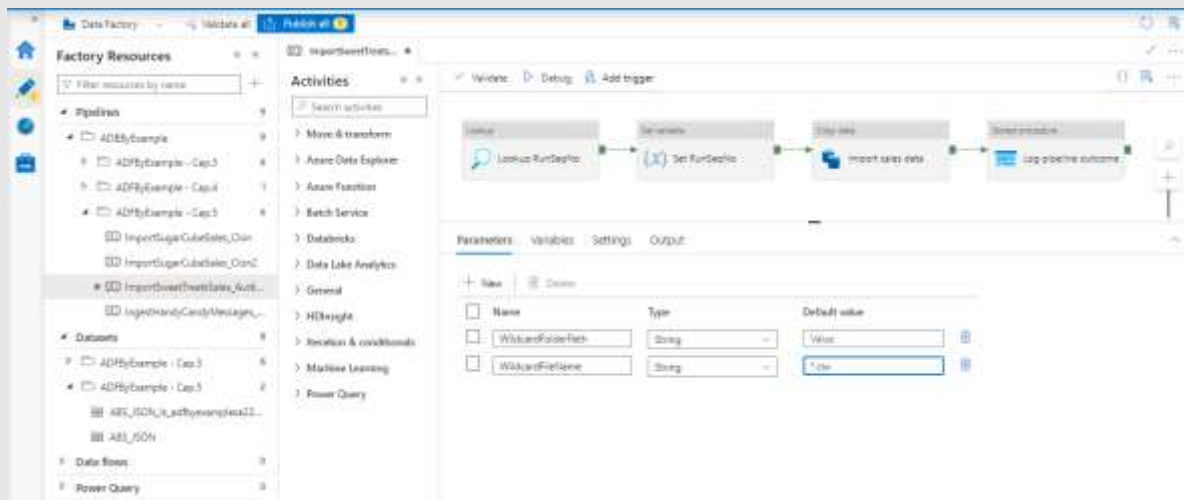


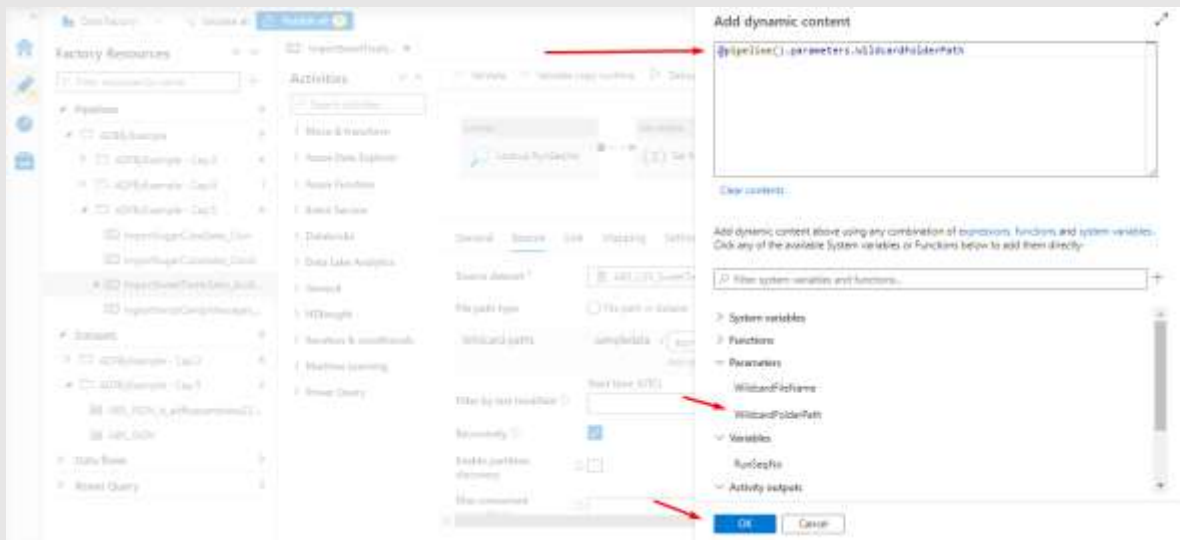
Figura 5-12 Definiciones de parámetros del pipeline



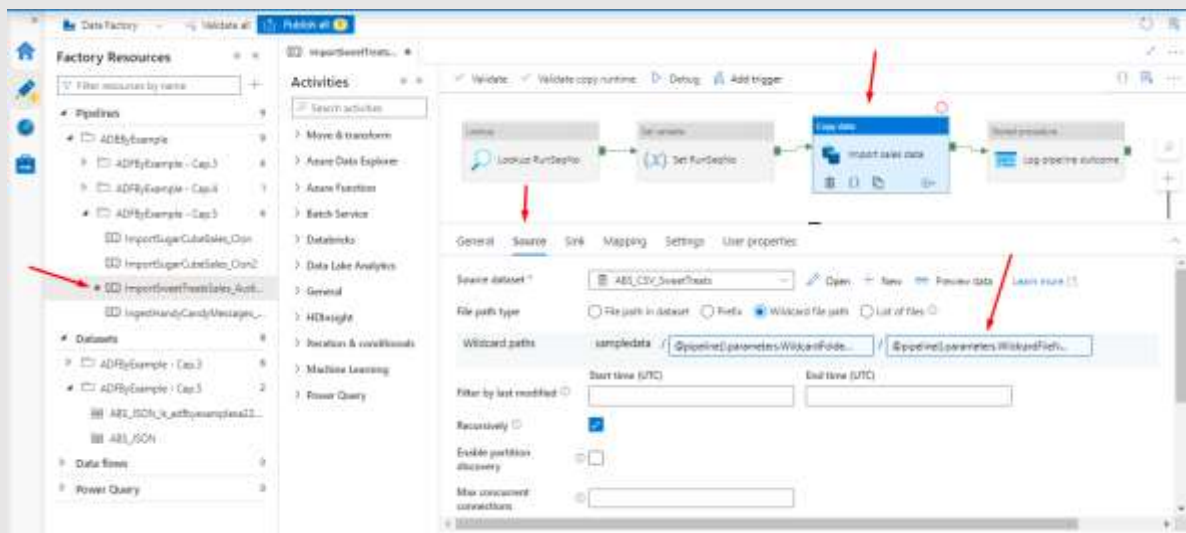
3. Seleccione la actividad de copia de datos del pipeline y abra su pestaña de configuración de origen. Haga clic en el componente Wildcard folder path de Wildcard paths e inicie el editor de expresiones.



4. Desplácese hacia abajo en el editor para encontrar la sección Parámetros - aquí encontrará una lista de los parámetros del pipeline que creó en el paso 2. Seleccione el parámetro "WildcardFolderPath".



5. La expresión `@pipeline().parameters.WildcardFolderPath` aparece en el panel de expresión - se hace referencia a los parámetros del pipeline utilizando la sintaxis `pipeline().parameters.ParameterName`. Haga clic en Finalizar.
6. Repita los pasos 4 y 5 para el componente de Wildcard file name de las rutas comodín, eligiendo esta vez el parámetro "WildcardFileName".



**Nota** La sintaxis `pipeline()` puede utilizarse en cualquiera de las actividades de un pipeline y se utiliza para referirse a variables del sistema (por ejemplo, `pipeline().RunId`) así como a parámetros. Por ello, los parámetros del pipeline deben dirigirse utilizando la sintaxis `pipeline().parameters.ParameterName`.

#### 5.4.2. Ejecutar el pipeline parametrizado

Cuando se ejecuta el pipeline parametrizado, el ADF UX le pide que proporcione valores para los parámetros del pipeline.

1. Proporcione los valores de los parámetros necesarios para los datos de Sweet Treats (la Figura 5-13 muestra las entradas pertinentes) y haga clic en OK para ejecutar la canalización. Observe que "WildcardFileName" se rellena previamente con el valor por defecto que especificó para el parámetro.

The screenshot shows the 'Pipeline run' dialog box in Azure Data Factory. The 'Parameters' section contains a table with the following data:

Name	Type	Value
WildcardFolderF	string	azure-data-factory-by-example-main/SampleData/SweetTreats/
WildcardFileNar	string	*.csv

Below the table, there are 'OK' and 'Cancel' buttons. The 'OK' button is highlighted with a blue border.

Figura 5-13 Entradas de parámetros necesarias para la ejecución del pipeline

2. Inspeccione el contenido de las tablas de la base de datos [dbo].[PipelineExecution] y [dbo].[Sales\_LOAD] para comprobar que los datos de Sweet Treats se han cargado correctamente.

RowId	RowId	PipelineKey	RunStartTime	RunEndTime	RunStatus	FilesRead	RowsRead	RowsCopied	Comments
1	1	9999999-1234-5678-9012-3456	2022-05-14 17:02:27.000	2022-05-14 17:02:27.000	Null	0	000	000	Null
2	2	2424242-8901-2345-6789-0123	2022-05-14 17:02:27.000	2022-05-14 17:02:27.000	Null	0	000	000	Null
3	3	7070707-4567-8901-2345-6789	2022-05-14 17:02:27.000	2022-05-14 17:02:27.000	Done	0	000	000	Null
4	4	4444444-1234-5678-9012-3456	2022-05-14 17:02:27.000	2022-05-14 17:02:27.000	Null	0	000	000	Null
5	5	4444444-1234-5678-9012-3456	2022-05-14 17:02:27.000	2022-05-14 17:02:27.000	Done	0	000	000	Null
6	6	4444444-1234-5678-9012-3456	2022-05-14 17:02:27.000	2022-05-14 17:02:27.000	Null	0	000	000	Null
7	7	1234567-8901-2345-6789-0123	2022-05-14 17:02:27.000	2022-05-14 17:02:27.000	Done	0	000	000	Null
8	8	1234567-8901-2345-6789-0123	2022-05-14 17:02:27.000	2022-05-14 17:02:27.000	Done	0	000	000	Pipeline ImportSweetTreats...
9	9	1234567-8901-2345-6789-0123	2022-05-14 17:02:27.000	2022-05-14 17:02:27.000	Done	0	000	000	Pipeline ImportSweetTreats...
10	10	1234567-8901-2345-6789-0123	2022-05-14 17:02:27.000	2022-05-14 17:02:27.000	Done	0	000	000	Pipeline ImportSweetTreats...

RowId	Retailer	Salesperson	Product	ManufacturerProductCode	SalespersonID	UnitsSold	SourceFileName	RowId
3	Sweet Treats	2018-05-01	Page 1.47oz	Null	1118,88	718	May-18/Sales.csv	18
4	Sweet Treats	2018-06-01	Snoogles 8.83oz	Null	6498,40	687	Jun-18/Sales.csv	18
5	Sweet Treats	2018-06-01	Snoogles 8.83oz	Null	271,32	28	Aug-18/Sales.csv	18
6	Sweet Treats	2018-06-01	Snoogles 8.83oz	Null	3922,33	489	Apr-18/Sales.csv	18
7	Sweet Treats	2018-07-01	Wagyu Wagyu 1.78oz	Null	819,53	148	Jul-18/Sales.csv	18
8	Sweet Treats	2018-07-01	Page 1.47oz	Null	437,71	279	Jul-18/Sales.csv	18
9	Sweet Treats	2018-07-01	The Original 18.57oz	Null	2487,48	880	Jul-18/Sales.csv	18
10	Sweet Treats	2018-07-01	Santa 7.89oz	Null	419,38	111	Jul-18/Sales.csv	18
11	Sweet Treats	2018-07-01	Salt Chocolate 20.48oz	Null	4203,55	857	Jul-18/Sales.csv	18
12	Sweet Treats	2018-07-01	Caramel 5.64oz	Null	3925,44	818	Jul-18/Sales.csv	18
13	Sweet Treats	2018-07-01	Montana 13.48oz	Null	7675,33	688	Jul-18/Sales.csv	18
14	Sweet Treats	2018-07-01	Fairy Sweet 1.79oz	Null	49,50	84	Jul-18/Sales.csv	18
15	Sweet Treats	2018-07-01	Gaugles 7.71oz	Null	3257,36	744	Jul-18/Sales.csv	18
16	Sweet Treats	2018-07-01	Associate 16.53oz	Null	2143,68	673	Jul-18/Sales.csv	18
17	Sweet Treats	2018-07-01	Piper 7.85oz	Null	2855,43	287	Jul-18/Sales.csv	18
18	Sweet Treats	2018-07-01	Samurai 4.41oz	Null	571,58	811	Jul-18/Sales.csv	18
19	Sweet Treats	2018-07-01	Cherry Caramel 7.89oz	Null	3682,78	713	Jul-18/Sales.csv	18
20	Sweet Treats	2018-07-01	Peanut 35.48oz	Null	4467,49	283	Jul-18/Sales.csv	18
21	Sweet Treats	2018-07-01	Synapse 18.08oz	Null	7289,81	729	Jul-18/Sales.csv	18
22	Sweet Treats	2018-07-01	Caramel 16.88oz	Null	3222,18	640	Jul-18/Sales.csv	18

- Ejecute el pipeline de nuevo, esta vez proporcionando los valores de los parámetros para los datos de Desserts4All. Para "WildcardFolderPath", edite el valor que utilizó para los datos de Sweet Treats, sustituyendo el nombre final de la carpeta "SweetTreats" por "Desserts4All". Utilice el mismo valor de "WildcardFileName" ("\*.csv").

The screenshot shows the 'Pipeline run' dialog box in Azure Data Studio. The 'Parameters' section is visible, showing two parameters: 'WildcardFolderPath' and 'WildcardFileName'. The 'WildcardFolderPath' parameter is highlighted with a red box, and a red arrow points to the 'OK' button at the bottom right of the dialog.

- Inspeccione el contenido de las tablas de la base de datos [dbo].[PipelineExecution] y [dbo].[Sales\_LOAD] de nuevo, verificando que esta vez los datos cargados se refieren a Desserts4All.

RowId	Retailer	SalesOrder	Product	Manufacturer	ProductCode	SalesOrderID	OrderID	SourceFile	RunSeq
1	Desserts4All	2018-05-01	Bananas 1.7lbs	NEL	10011,40	0413	01-10,100	11	
2	Desserts4All	2018-05-01	Bananas 1.7lbs	NEL	1348,04	0413	01-10,100	11	
3	Desserts4All	2018-05-01	Bananas 1.7lbs	NEL	1024,30	0413	01-10,100	11	
4	Desserts4All	2018-05-01	Bananas 1.7lbs	NEL	1040,07	0713	01-10,100	11	
5	Desserts4All	2018-05-01	Bananas 1.7lbs	NEL	4029,14	1009	04-10,100	11	
6	Desserts4All	2018-05-01	Bananas 1.7lbs	NEL	5152,20	2004	07-10,100	11	
7	Desserts4All	2018-05-01	Bananas 1.7lbs	NEL	0210,10	2045	01-10,100	11	
8	Desserts4All	2018-05-01	Bananas 2.4lbs	NEL	1002,50	0413	01-10,100	11	
9	Desserts4All	2018-05-01	Bananas 1.7lbs	NEL	10011,40	0413	01-10,100	11	
10	Desserts4All	2018-05-01	Butterbutter 3.0lbs	NEL	11323,00	1001	01-10,100	11	
11	Desserts4All	2018-05-01	Butterbutter 3.0lbs	NEL	12994,10	1002	01-10,100	11	
12	Desserts4All	2018-05-01	Butterbutter 25.5lbs	NEL	34740,82	1410	01-10,100	11	
13	Desserts4All	2018-05-01	Eggs 1.2lbs	NEL	10013,00	0413	01-10,100	11	
14	Desserts4All	2018-05-01	Eggs 1.2lbs	NEL	4034,17	0713	01-10,100	11	
15	Desserts4All	2018-05-01	Eggs 1.2lbs	NEL	10013,00	1500	01-10,100	11	
16	Desserts4All	2018-05-01	Eggs 1.2lbs	NEL	107740,40	1710	01-10,100	11	
17	Desserts4All	2018-05-01	Eggs 1.2lbs	NEL	1130,50	1024	01-10,100	11	
18	Desserts4All	2018-05-01	Eggs 1.2lbs	NEL	3020,04	1710	01-10,100	11	
19	Desserts4All	2018-05-01	Eggs 1.2lbs	NEL	7100,70	0413	01-10,100	11	
20	Desserts4All	2018-05-01	Eggs 1.2lbs	NEL	14247,70	0713	01-10,100	11	

RunSeq	PipelineRunID	RunStartDateTime	RunEndDateTime	RunStatus	FilesRead	RowsRead	RowsCopied	Comments
1	01e0b0d1-4a01-4a01-4a01-4a01	2022-05-14 17:02:27.000	2022-05-14 17:02:27.000	Done	0	000	000	
2	2d12f0b0-4a01-4a01-4a01-4a01	2022-05-14 17:02:27.000	2022-05-14 17:02:27.000	Done	0	000	000	
3	7f9f5b00-4a01-4a01-4a01-4a01	2022-05-14 17:02:27.000	2022-05-14 17:02:27.000	Done	0	000	000	
4	4d4d10d1-4a01-4a01-4a01-4a01	2022-05-14 17:02:27.000	2022-05-14 17:02:27.000	Done	0	000	000	
5	4d4d10d1-4a01-4a01-4a01-4a01	2022-05-14 17:02:27.000	2022-05-14 17:02:27.000	Done	0	000	000	
6	5b705d1-4a01-4a01-4a01-4a01	2022-05-14 17:02:27.000	2022-05-14 17:02:27.000	Done	0	000	000	
7	1d0f0d1-4a01-4a01-4a01-4a01	2022-05-14 17:02:27.000	2022-05-14 17:02:27.000	Done	0	000	000	
8	5d0f0d1-4a01-4a01-4a01-4a01	2022-05-14 17:02:27.000	2022-05-14 17:02:27.000	Done	0	000	000	Pipeline ImportSweetTreats...
9	5d0f0d1-4a01-4a01-4a01-4a01	2022-05-14 17:02:27.000	2022-05-14 17:02:27.000	Done	0	000	000	Pipeline ImportSweetTreats...
10	073d0d1-4a01-4a01-4a01-4a01	2022-05-14 17:02:27.000	2022-05-14 17:02:27.000	Done	0	000	000	Pipeline ImportSweetTreats...
11	073d0d1-4a01-4a01-4a01-4a01	2022-05-14 17:02:27.000	2022-05-14 17:02:27.000	Done	0	000	000	Pipeline ImportSweetTreats...

Como acaba de demostrar, el nuevo pipeline puede ahora cargar los datos almacenados en el formato de archivo de Sweet Treats desde cualquier carpeta que elija. Observe que la actividad de Copiar datos en el nuevo pipeline sigue utilizando el dataset "ABS\_CSV\_SweetTreats". El pipeline se ejecuta correctamente porque la actividad anula el directorio y la ruta del dataset en tiempo de ejecución (y todos sus datos de origen están en el mismo contenedor blob storage), pero esto le da a la definición del pipeline una apariencia inconsistente.

Un enfoque más limpio sería crear un dataset "ABS\_CSV" reutilizable, similar al dataset "ABS\_JSON" que creó anteriormente. Cree un dataset "ABS\_CSV" parametrizado y pruébelo ahora; lo volverá a necesitar en el capítulo 6.

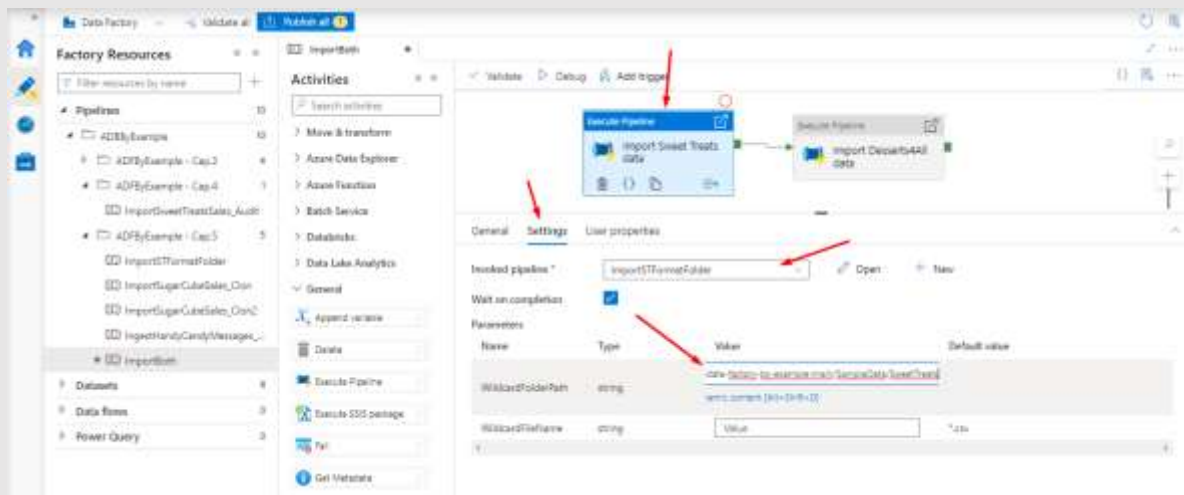
**Consejo** No olvide especificar First row as header en el nuevo dataset "ABS\_CSV".

### 5.4.3. Utilice la actividad Execute Pipeline

Introducir los valores de los parámetros manualmente es aceptable en el contexto de una sesión de ADF UX, pero la mayoría de las ejecuciones de pipeline no tienen lugar de forma interactiva. En las ejecuciones de pipeline no supervisadas, los valores de los parámetros deben ser suministrados por cualquier proceso que invoque el pipeline. En muchos casos, este proceso será otro pipeline de ADF, utilizando la actividad Execute Pipeline.

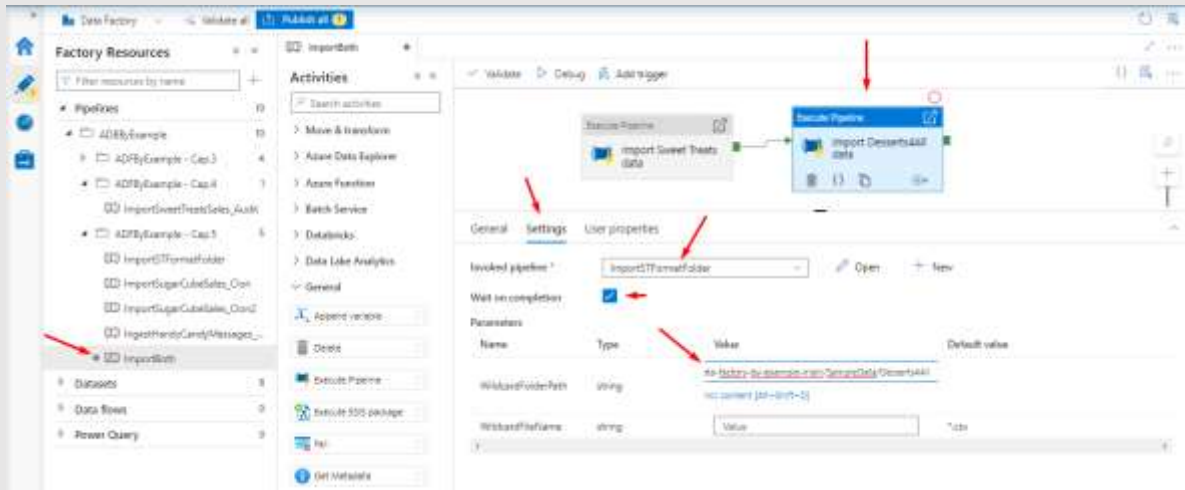
En esta sección, creará un nuevo pipeline que utiliza la actividad Execute Pipeline para ejecutar el pipeline "ImportSTFormatFolder" dos veces - una para los datos de Sweet Treats y otra para Desserts4All.

1. Cree un nuevo pipeline en la carpeta "Capítulo 5" y nómbrelo adecuadamente.
2. Arrastre una actividad Execute Pipeline al lienzo de creación. Esta actividad se encuentra en la sección General de la caja de herramientas de actividades.
3. En la pestaña de configuración de la actividad, seleccione "ImportSTFormatFolder" en el menú desplegable Invoked pipeline. Al hacerlo, los campos de entrada para los parámetros del pipeline aparecen inmediatamente debajo.
4. Asegúrese de que la casilla Wait on completion esté marcada y, a continuación, proporcione los valores de los parámetros para los datos de Sweet Treats, como antes.



5. Repita los pasos 2-4, esta vez suministrando los valores de los parámetros para los datos de Desserts4All.





6. Cree una dependencia de actividad entre las dos actividades (para evitar que se ejecuten simultáneamente). La Figura 5-14 muestra el pipeline configurado y la pestaña de configuración de la actividad Execute Pipeline.

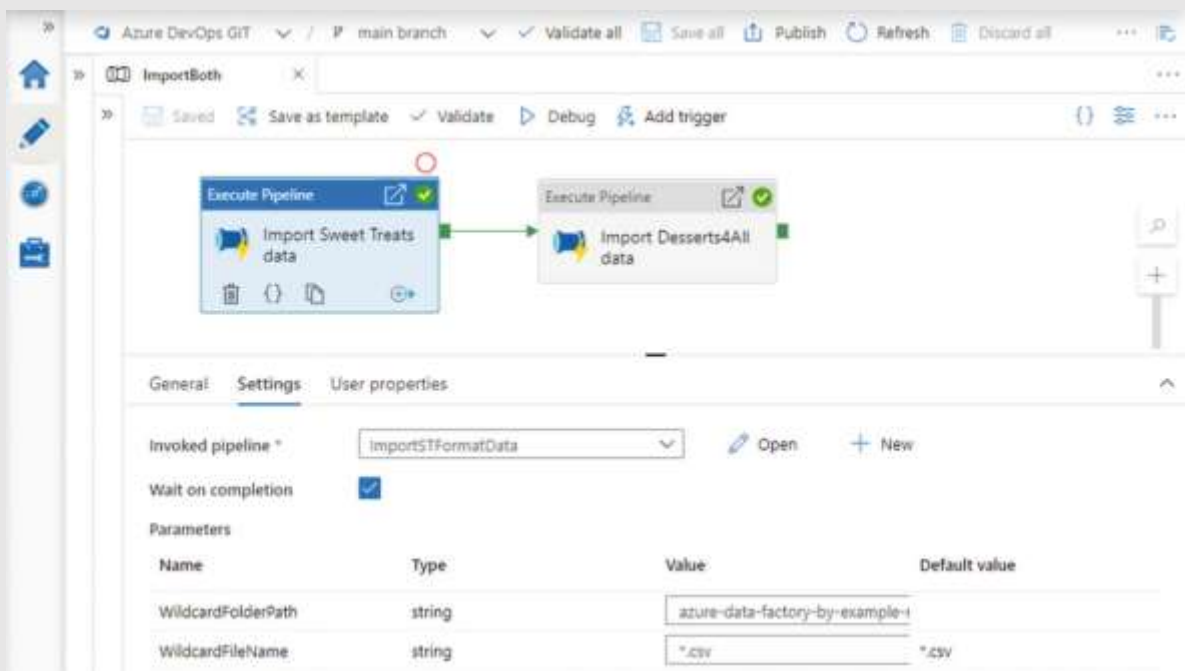


Figura 5-14 Actividades secuenciales de Execute Pipeline

**Sugerencia** Si deja el parámetro WildcardFileName en blanco, la actividad Execute Pipeline pasa un string vacío al pipeline "ImportSTFormatFolder" - los valores de los parámetros por defecto del pipeline no se sustituyen en esta situación.

Ejecute el pipeline y compruebe que se han creado dos nuevos registros en la tabla de la base de datos [dbo].[PipelineExecution]. La tabla [dbo].[Sales\_LOAD] sólo contendrá los datos cargados por la segunda actividad Execute Pipeline, ya que la actividad Copy data de la pipeline de carga trunca la tabla de destino antes de copiar los nuevos datos. Los pipelines que ha desarrollado hasta ahora

no persisten los datos cargados, pero en el Capítulo 6 ampliaré el pipeline de carga para transferir datos a una tabla para su almacenamiento permanente.

La pestaña Output del panel de configuración del pipeline contiene dos filas, una por cada actividad de Execute Pipeline. La salida de cada actividad incluye el ID de ejecución del pipeline que invocó, mostrado como un enlace - siga el enlace para ver la información de ejecución de la actividad para el pipeline invocado.

RowId	Retailer	SalesOrder	Product	Manufacturer	ProductCode	SalesOrderID	UnitsSold	SourceFileName	RunDate
1	Deserts4All	2018-05-01	BananaOne 1.7kg	NLI	1001148	0412	81-20.1kg	13	
2	Deserts4All	2018-05-01	BananaOne 1.7kg	NLI	1001148	0412	81-20.1kg	13	
3	Deserts4All	2018-05-01	BananaOne 1.7kg	NLI	1001148	0412	81-20.1kg	13	
4	Deserts4All	2018-05-01	BananaOne 1.7kg	NLI	1001148	0412	81-20.1kg	13	
5	Deserts4All	2018-05-01	BananaOne 1.7kg	NLI	1001148	0412	81-20.1kg	13	
6	Deserts4All	2018-05-01	BananaOne 1.7kg	NLI	1001148	0412	81-20.1kg	13	
7	Deserts4All	2018-05-01	BananaOne 1.7kg	NLI	1001148	0412	81-20.1kg	13	
8	Deserts4All	2018-05-01	BananaOne 1.7kg	NLI	1001148	0412	81-20.1kg	13	
9	Deserts4All	2018-05-01	BananaOne 1.7kg	NLI	1001148	0412	81-20.1kg	13	
10	Deserts4All	2018-05-01	BananaOne 1.7kg	NLI	1001148	0412	81-20.1kg	13	
11	Deserts4All	2018-05-01	BananaOne 1.7kg	NLI	1001148	0412	81-20.1kg	13	
12	Deserts4All	2018-05-01	BananaOne 1.7kg	NLI	1001148	0412	81-20.1kg	13	
13	Deserts4All	2018-05-01	BananaOne 1.7kg	NLI	1001148	0412	81-20.1kg	13	
14	Deserts4All	2018-05-01	BananaOne 1.7kg	NLI	1001148	0412	81-20.1kg	13	
15	Deserts4All	2018-05-01	BananaOne 1.7kg	NLI	1001148	0412	81-20.1kg	13	
16	Deserts4All	2018-05-01	BananaOne 1.7kg	NLI	1001148	0412	81-20.1kg	13	
17	Deserts4All	2018-05-01	BananaOne 1.7kg	NLI	1001148	0412	81-20.1kg	13	
18	Deserts4All	2018-05-01	BananaOne 1.7kg	NLI	1001148	0412	81-20.1kg	13	
19	Deserts4All	2018-05-01	BananaOne 1.7kg	NLI	1001148	0412	81-20.1kg	13	
20	Deserts4All	2018-05-01	BananaOne 1.7kg	NLI	1001148	0412	81-20.1kg	13	

RowId	PipelineRunID	PipelineName	RunDate	RunTime	RunStatus	FilesRead	RowsRead	RowsCopied	Comments
1	95000000-0000-0000-0000-0000	NLI	2022-05-14 07:14:55.000	NLI	0	000	000	NLI	
2	2022-05-14 17:02:27.000	NLI	2022-05-14 17:02:27.000	NLI	0	000	000	NLI	
3	2022-05-14 17:02:27.000	NLI	2022-05-14 17:02:27.000	NLI	0	000	000	NLI	
4	2022-05-14 17:02:27.000	NLI	2022-05-14 17:02:27.000	NLI	0	000	000	NLI	
5	2022-05-14 17:02:27.000	NLI	2022-05-14 17:02:27.000	NLI	0	000	000	NLI	
6	2022-05-14 17:02:27.000	NLI	2022-05-14 17:02:27.000	NLI	0	000	000	NLI	
7	2022-05-14 17:02:27.000	NLI	2022-05-14 17:02:27.000	NLI	0	000	000	NLI	
8	2022-05-14 17:02:27.000	NLI	2022-05-14 17:02:27.000	NLI	0	000	000	NLI	
9	2022-05-14 17:02:27.000	NLI	2022-05-14 17:02:27.000	NLI	0	000	000	NLI	
10	2022-05-14 17:02:27.000	NLI	2022-05-14 17:02:27.000	NLI	0	000	000	NLI	
11	2022-05-14 17:02:27.000	NLI	2022-05-14 17:02:27.000	NLI	0	000	000	NLI	
12	2022-05-14 17:02:27.000	NLI	2022-05-14 17:02:27.000	NLI	0	000	000	NLI	
13	2022-05-14 17:02:27.000	NLI	2022-05-14 17:02:27.000	NLI	0	000	000	NLI	

Name	Type	Run start	Duration	Status	Integration path
Import Deserts4All data	Execute Pipeline	2022-05-10T18:42:21	00:00:21	Succeeded	
Import Sweet Treats data	Execute Pipeline	2022-05-10T18:41:56	00:00:21	Succeeded	



**Para los desarrolladores de SSIS:** Al elegir un pipeline para la actividad Execute Pipeline, se le ofrece una lista desplegable fija - no hay opción de lanzar el editor de expresiones. A diferencia de la tarea Execute Package Task de SSIS, no se puede especificar un nombre de pipeline utilizando un parámetro o una variable. (Incluso en SSIS, esto requiere que el valor de la propiedad DelayValidation de la tarea sea "true" - el concepto de validación retardada no está soportado en ADF).

#### 5.4.4. Ejecución paralela

La dependencia de actividad entre las dos actividades de Execute Pipeline las obliga a ejecutarse secuencialmente - sin ella, las dos actividades se ejecutarán simultáneamente, en paralelo. Incluso con la dependencia en su lugar, la ejecución es sólo secuencial porque la primera actividad espera a que su ejecución del pipeline se complete - este es el efecto de la casilla de verificación Wait for completion.

En este caso, las dos actividades deben ejecutarse secuencialmente porque el pipeline invocado está mal diseñado para la ejecución en paralelo. Ambas ejecuciones truncan la tabla de destino de forma independiente, por lo que ejecutarlas en paralelo podría permitir que una ejecución truncara los datos mientras la otra está en proceso de cargarlos. En el Capítulo 6, se revisará el diseño del pipeline de carga para hacerlo seguro para la paralelización.

## 5.5. Parámetros globales

A diferencia de los otros tipos de parámetros cubiertos anteriormente en el capítulo, los parámetros globales no son un medio para inyectar diferentes valores en tiempo de ejecución. La función de los parámetros globales es, en cambio, proporcionar valores constantes, compartidos por todos los pipelines. Por ejemplo, si su inquilino de Azure contiene instancias de Azure Data Factory separadas para el desarrollo, las pruebas y la producción, un parámetro global es una forma conveniente de almacenar información específica de la instancia.

Los parámetros globales se crean y editan en la página de parámetros globales del centro de gestión de ADF UX. La Figura 5-15 muestra los parámetros que se utilizan para identificar el grupo de recursos de la fábrica y para indicar que se trata del entorno de desarrollo.

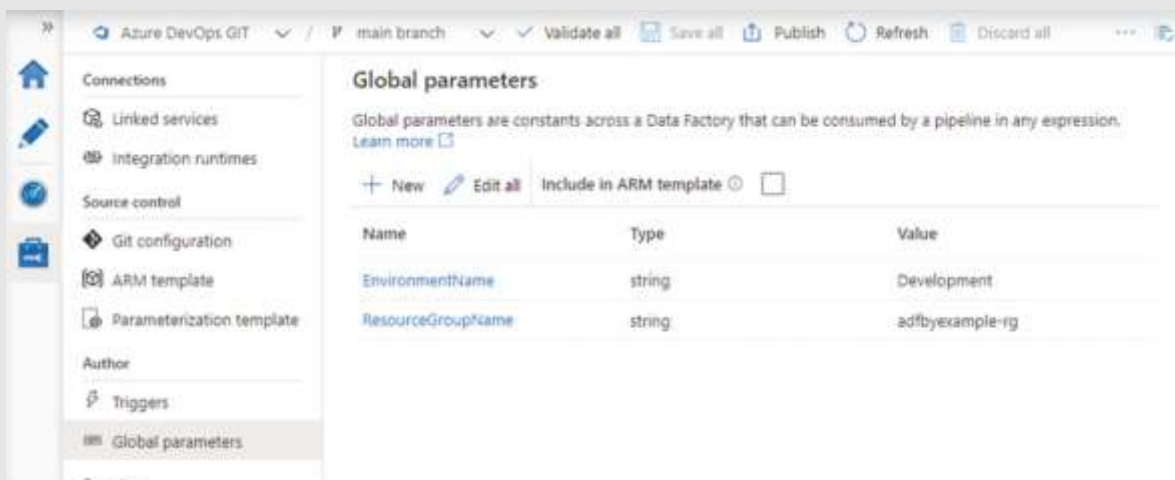


Figura 5-15 Editor de parámetros globales en el hub de gestión de ADF UX

Los parámetros globales pueden ser referenciados en expresiones dentro del ámbito de las actividades del pipeline y están disponibles para su selección en el constructor de expresiones del ADF. La sintaxis utilizada para referirse a un parámetro global es

```
pipeline().globalParameters.ParameterName
```

Por ejemplo, el parámetro global "EnvironmentName" mostrado en la Figura 5-15 es referido en una expresión ADF usando `pipeline().globalParameters.EnvironmentName`.

## 5.6. Repaso del capítulo

En este capítulo se han definido parámetros para pipelines, dataset y servicios enlazados. Los parámetros permiten que un recurso de fábrica requiera que se proporcionen ciertos valores de datos en tiempo de ejecución, explotando esos valores al utilizarlos en las expresiones de ADF. También has visto que las definiciones JSON de los recursos pueden editarse directamente - la UX de ADF suele ser una forma más fácil de gestionar los recursos, pero la capacidad de editar JSON directamente es a veces útil.

Un caso de uso común para los parámetros es permitir que los recursos de fábrica sean definidos genéricamente. Esto permite la reutilización, reduciendo el desorden en su fábrica de datos y evitando la reimplementación innecesaria de componentes de uso frecuente. Los servicios vinculados reutilizables pueden requerir que se pase información de conexión sensible en tiempo de ejecución: los secretos de Azure Key Vault proporcionan una forma segura de hacerlo.

El grado de parametrización de los recursos de fábrica depende en gran medida de sus propios casos de uso y patrones de utilización. A medida que continúe utilizando ADF, encontrará otros lugares en los que las decisiones de diseño sólo pueden tomarse en el contexto de sus propios requisitos técnicos específicos.

### 5.6.1. Conceptos clave

- ❖ **Runtime parameter:** Un marcador de posición en una definición de recurso de fábrica para un valor sustituido en tiempo de ejecución. Los parámetros de pipeline, dataset y servicios enlazados son parámetros de tiempo de ejecución; los parámetros globales no lo son.
- ❖ **Optional parameter:** Un parámetro en tiempo de ejecución puede hacerse opcional definiendo un valor por defecto para él. Si no se proporciona un valor en tiempo de ejecución, el valor por defecto se sustituye por el parámetro.
- ❖ **Reutilización:** Los parámetros en tiempo de ejecución permiten definir recursos de fábrica de forma reutilizable, utilizando diferentes valores de parámetros para variar el comportamiento de los recursos.
- ❖ **Parámetro global:** Un valor constante, compartido por todas las tuberías, que no se espera que cambie con frecuencia. Se hace referencia a los parámetros globales en las expresiones de ADF, dentro del ámbito de las actividades de la tubería, utilizando la sintaxis `pipeline().globalParameters.ParameterName`.
- ❖ **Pipeline parameter:** Parámetro de tiempo de ejecución para un pipeline de ADF. Se hace referencia a los parámetros del pipeline en las expresiones ADF, dentro del ámbito de las actividades del pipeline, utilizando la sintaxis `pipeline().parameters.ParameterName`.
- ❖ **Dataset parameter:** Parámetro de tiempo de ejecución para un dataset de ADF. Se hace referencia a los parámetros del conjunto de datos en las expresiones de ADF, dentro del ámbito del conjunto de datos, utilizando la sintaxis `dataset().ParameterName`.

- ❖ **Linked service parameter**: Parámetro en tiempo de ejecución para un servicio vinculado de ADF. Se hace referencia a los parámetros del conjunto de datos en las expresiones de ADF, dentro del ámbito del servicio vinculado, utilizando la sintaxis `linkedService().ParameterName`. La UX de ADF no siempre admite la gestión de parámetros para los servicios vinculados, pero los parámetros pueden definirse y utilizarse editando la definición JSON de un servicio vinculado.
- ❖ **Actividad Execute Pipeline**: Actividad de pipeline de ADF utilizada para ejecutar otro pipeline dentro de la misma instancia de fábrica de datos.
- ❖ **Azure Key Vault**: Un repositorio seguro para secretos y claves criptográficas.
- ❖ **Secreto**: Un par nombre/valor almacenado en un Azure Key Vault. El valor suele contener información sensible, como las credenciales de autenticación del servicio. Una forma segura de manejar esta información es referirse al secreto por su nombre - un servicio que requiere el valor del secreto puede recuperarlo del vault por su nombre si se le permite hacerlo.
- ❖ **Service principal**: Una identidad creada para su uso con una aplicación o servicio - como Azure Data Factory - que permite identificar el servicio cuando los recursos externos requieren autenticación y autorización.
- ❖ **Managed identity**: Una identidad gestionada asocia un principal de servicio con una instancia de Azure Data Factory (u otros recursos de Azure). Una identidad gestionada asignada por el sistema se crea automáticamente para las nuevas instancias de ADF creadas en el portal de Azure y se elimina automáticamente cuando se elimina su fábrica.
- ❖ **Access policy**: La política de acceso de un key vault define qué servicios o usuarios principales pueden acceder a los datos almacenados en el vault.

## 5.6.2. Para los desarrolladores de SSIS

Los parámetros de los pipelines de ADF son conceptualmente similares a los parámetros de los paquetes SSIS. Al igual que en los paquetes SSIS, los parámetros de canalización se especifican como parte de la definición del pipeline y se puede hacer referencia a ellos en cualquier parte de las actividades del pipeline.

Los servicios vinculados de la fábrica de datos se parecen mucho a los gestores de conexión a nivel de proyecto de SSIS, pero se comportan de forma algo diferente. Los gestores de conexión de SSIS no soportan la inyección externa de parámetros - los gestores de conexión a nivel de paquete se parametrizan utilizando expresiones SSIS que hacen referencia a parámetros o variables del paquete.

El concepto de una variable o parámetro con alcance al nivel exacto del gestor de conexión no se encuentra en SSIS. Los gestores de conexión a nivel de proyecto pueden hacer uso de parámetros de proyecto, pero el alcance de éstos es a nivel de proyecto (muy parecido al alcance a nivel de fábrica de los parámetros globales de ADF).

La actividad Execute Pipeline de ADF tiene el mismo propósito que Execute Package Task de SSIS, pero los pipelines invocados sólo pueden especificarse en tiempo de diseño. SSIS le permite anular esta restricción estableciendo la propiedad DelayValidation de Execute Package Task en true, pero las actividades de ADF no tienen un ajuste comparable.

Los paquetes SSIS se ejecutan frecuentemente utilizando trabajos de SQL Server Agent, lo que requiere que la cuenta de servicio del agente (o proxy) tenga acceso a los recursos externos. La identidad gestionada de una instancia de ADF sirve para el mismo propósito que una cuenta de dominio utilizada para ejecutar el servicio de agente de SQL Server - permite que los procesos ejecutados por la fábrica sean autenticados y autorizados contra almacenes de datos y servicios externos.