

# Azure Data Engineer Associate Certification Guide

A hands-on reference guide to developing your data engineering skills and preparing for the DP-203 exam

Newton Alex



<b>Capítulo 12: Diseño de la seguridad de las políticas y normas de datos .....</b>	<b>4</b>
12.1. Requisitos técnicos.....	4
12.2. Introducción a los requisitos de seguridad y privacidad.....	5
12.3. Diseñar e implementar el cifrado de datos en reposo y en tránsito .....	6
12.3.1. Cifrado en reposo.....	6
Cifrado en reposo en Azure Storage .....	6
Cifrado en reposo en Azure Synapse SQL .....	7
Always Encrypted (siempre encriptado) .....	9
12.3.2. Cifrado en tránsito .....	11
Habilitación del cifrado en tránsito para Azure Storage .....	11
Habilitar el cifrado en tránsito para Azure Synapse SQL.....	12
12.4. Diseñar e implementar una estrategia de auditoría de datos .....	13
12.4.1. Auditoría de almacenamiento.....	13
Usando una configuración de diagnóstico clásica.....	13
Usando Azure Monitor .....	14
12.4.2. Auditoría en SQL.....	15
12.5. Diseño e implementación de una estrategia de enmascaramiento de datos .....	17
12.6. Diseño e implementación del control de acceso basado en roles de Azure y una lista de control de acceso tipo POSIX para Data Lake Storage Gen2.....	20
12.6.1. Restricción del acceso mediante Azure RBAC.....	20
12.6.2. Restricción de acceso mediante ACLs .....	21
¿Cómo decide Azure entre RBAC y ACLs si hay reglas conflictivas? .....	23
Limitaciones de RBAC y ACL .....	23
12.7. Diseñar e implementar la seguridad a nivel de filas y columnas .....	25
12.7.1. Diseño de la seguridad a nivel de filas .....	25
12.7.2. Diseñar la seguridad a nivel de columna.....	27
12.8. Diseñar y aplicar una política de conservación de datos .....	28
12.9. Diseño para purgar datos en función de los requisitos del negocio .....	30
12.9.1. Purga de datos en Azure Data Lake Storage Gen2.....	30
12.9.2. Purgar datos en Azure Synapse SQL.....	31
12.10. Gestión de identidades, claves y secretos en diferentes tecnologías de plataformas de datos.....	32
12.10.1. Azure Active Directory .....	32
Identidades gestionadas .....	34

12.10.2. Azure Key Vault .....	34
12.10.3. Claves de acceso y claves de acceso compartidas en Azure Storage.....	39
12.11. Implementación de endpoints seguros (privados y públicos) .....	41
12.12. Implementación de tokens de recursos en Azure Databricks.....	46
12.13. Carga de un DataFrame con información sensible .....	49
12.14. Escribir datos encriptados en tablas o archivos Parquet .....	52
12.15. Diseño de la privacidad de los datos y gestión de la información sensible .....	53
12.15. Microsoft Defender .....	54
Microsoft Defender para el almacenamiento .....	54
Microsoft Defender para SQL.....	54
Resumen.....	55

# Capítulo 12: Diseño de la seguridad de las políticas y normas de datos

---

Bienvenido a la siguiente sección del programa de estudios, Diseñar e implementar la seguridad de los datos. Esta sección representa alrededor del 10-15% de las preguntas de la certificación. Según el programa de estudios, esta sección debería tener dos capítulos: uno centrado en los aspectos de diseño y otro en los de implementación. Pero, para asegurar un mejor flujo de temas y evitar demasiados cambios de contexto, he fusionado los detalles de diseño e implementación en este único capítulo. Una vez que hayas completado este capítulo, deberás ser capaz de reconocer la información sensible y ser capaz de diseñar e implementar varias técnicas de manejo de información sensible como enmascaramiento de datos, seguridad a nivel de filas y columnas, implementación de acceso basado en roles y listas de acceso controlado, habilitación de encriptación y más. También conocerá las buenas prácticas para el manejo de claves, secretos y certificados, y comprenderá los detalles de implementación de bajo nivel del manejo de datos seguros en Spark. En general, podrá ocuparse del diseño y la implementación de los aspectos de seguridad y privacidad de los datos de su data lake.

En este capítulo cubriremos los siguientes temas:

- Diseño e implementación del cifrado de datos en reposo y en tránsito
- Diseño e implementación de estrategias de auditoría de datos
- Diseño e implementación de estrategias de enmascaramiento de datos
- Diseño e implementación del control de acceso basado en roles de Azure y listas de control de acceso tipo POSIX para Data Lake Storage Gen2
- Diseño e implementación de seguridad a nivel de filas y columnas
- Diseño e implementación de políticas de retención de datos
- Diseño e implementación de la purga de datos basada en los requisitos del negocio
- Gestión de identidades, claves y secretos en diferentes tecnologías de plataformas de datos
- Implementación de endpoints seguros (privados y públicos)
- Implementación de tokens de recursos en Azure Databricks
- Carga de DataFrames con información sensible
- Escribir datos encriptados en tablas o archivos Parquet
- Diseño de la privacidad de los datos y gestión de la información sensible

## 12.1. Requisitos técnicos

Para este capítulo, necesitarás lo siguiente

- Una cuenta de Azure (gratuita o de pago)

¡Empecemos!

## 12.2. Introducción a los requisitos de seguridad y privacidad

¿Cómo se diseña la seguridad y la privacidad de los datos? Bien, tomemos un ejemplo e intentemos recorrer algunos escenarios. Consideremos nuestro fiel ejemplo de la empresa Imaginary Airport Cabs (IAC) que hemos utilizado en los capítulos anteriores. Ya hemos visto que la empresa de taxis recibe una gran cantidad de viajes, clientes e información de los conductores. También hemos aprendido a almacenar los datos en el data lake y en los almacenes SQL. Ahora, vamos a profundizar un poco más en el tema del almacenamiento y a averiguar cómo salvaguardar la información confidencial y privada.

Consideremos los siguientes requisitos del equipo de seguridad del CAI:

- Los datos deben ser almacenados y transferidos de forma segura, ya que se trata de sistemas en la nube, y nadie más que los empleados del IAC debe tener acceso a los datos.
- Los cambios en los datos y cualquier actividad en los datos deben ser registrados por razones de cumplimiento.
- No todo el mundo debería tener acceso a todos los datos. Debe ser sobre una base de necesidad de conocimiento.
- Mantenga la privacidad del cliente a toda costa.
- Los datos antiguos deben eliminarse de forma segura al cabo de un mes.

Estos parecen ser requisitos bastante estándar para cualquier empresa. Ahora, entremos en cada uno de los temas de este capítulo y aprendamos cómo ayudan a cumplir los requisitos anteriores.

### 12.3. Diseñar e implementar el cifrado de datos en reposo y en tránsito

Las preguntas habituales de cualquier persona que quiera almacenar datos en una nube pública serían las siguientes:

- ¿Qué seguridad tienen mis datos?
- ¿Pueden los empleados de la empresa de la nube acceder a mis datos?
- ¿Puede cualquier persona ajena a la empresa acceder a mis datos?

Las empresas de la nube, como Azure, suelen responder a estas inquietudes mediante el cifrado en reposo y en tránsito. Este también es el primer requisito de nuestro ejemplo de requisitos para IAC. Veamos en detalle el cifrado en reposo.

#### 12.3.1. Cifrado en reposo

El cifrado en reposo es el proceso de cifrar los datos antes de escribirlos en los discos y descifrarlos cuando lo solicitan las aplicaciones. El cifrado en reposo protege los datos contra el robo de discos físicos, la recuperación de datos de discos perdidos, el acceso no autorizado a los datos por parte de empleados malintencionados de la empresa en la nube, etc. A menos que alguien tenga la clave de descifrado o posea recursos de supercomputación increíblemente potentes (del tipo que podrían tener los gobiernos -aunque, incluso con los superordenadores actuales, es extremadamente difícil si la clave de cifrado es lo suficientemente fuerte y grande), los datos no pueden ser recuperados. Simplemente aparecerán como un galimatías si alguien intenta copiar directamente los datos de los discos.

Esta forma de seguridad se ha convertido en un requisito fundamental para cualquier dato almacenado en la nube, y Azure hace un buen trabajo al proporcionar opciones de cifrado en reposo para la mayoría de sus soluciones de almacenamiento. En este tema, veremos las opciones de cifrado en reposo disponibles en Azure Storage y Azure Synapse SQL.

El cifrado en reposo y en tránsito suele ser necesario para cumplir con diversas normativas. Por lo tanto, no se trata sólo de las preocupaciones de los clientes; también puede ser requerido por la ley. Puede conocer las distintas normativas y los niveles de cumplimiento que ofrecen los distintos servicios de Microsoft aquí: <https://docs.microsoft.com/en-us/compliance/regulatory/offering-home>.

Conozcamos ahora cómo Azure Storage y Synapse SQL pools proporcionan cifrado en reposo.

#### Cifrado en reposo en Azure Storage

Azure Storage proporciona cifrado en reposo por defecto. Asegura tus datos sin que ni siquiera lo solicites. De hecho, no se puede desactivar el cifrado de Azure Storage. Azure Storage utiliza sus propias claves para cifrar los datos. También ofrece la opción de que los clientes utilicen sus propias claves de cifrado. Esto proporciona un control adicional al usuario. Estas claves proporcionadas por

el usuario se denominan Claves Administradas por el Cliente (CMK). Puede habilitar las CMK desde la pantalla de Azure Storage, como se muestra aquí:

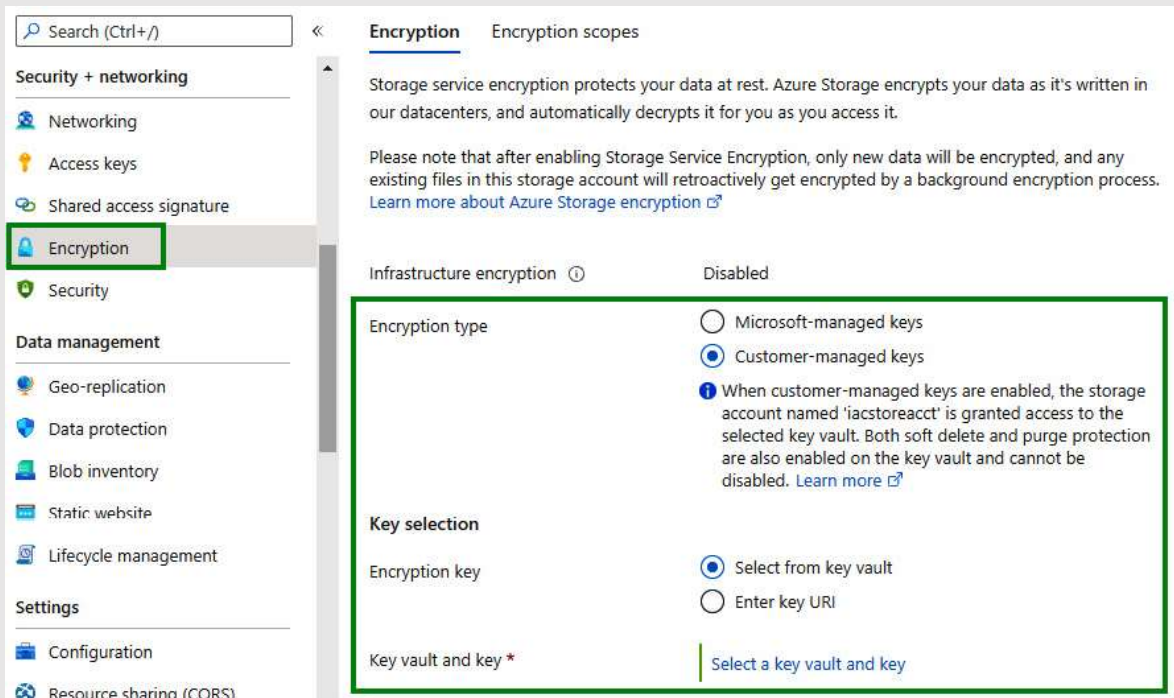


Figura 12.1 - Habilitación de CMKs en Azure Storage

Uno de los requisitos de la CMK es que la clave del cliente debe estar almacenada de forma segura en Azure Key Vault. Piensa en Azure Key Vault como una versión online de una bóveda física. Puede almacenar todas sus contraseñas, claves secretas, claves de acceso, etc. en Key Vault y las aplicaciones pueden acceder a estas claves de forma segura durante el tiempo de ejecución. Este método asegura que los secretos y contraseñas no necesitan ser almacenados como parte de la base de código. Aprenderemos más sobre Key Vault más adelante en este capítulo.

Puedes aprender más sobre las CMKs aquí: <https://docs.microsoft.com/en-us/azure/storage/common/customer-managed-keys-overview>.

El cifrado mediante CMKs resolvería las preocupaciones de IAC sobre el almacenamiento seguro de los datos. Veamos ahora como Synapse SQL encripta los archivos de datos.

#### Cifrado en reposo en Azure Synapse SQL

Esta sección se aplica a las tecnologías de Azure SQL en general. En Azure Synapse SQL, la encriptación en reposo se lleva a cabo utilizando una característica llamada Transparent Data Encryption (TDE). En TDE, el cifrado ocurre en tiempo real a nivel de página. Las páginas se cifran antes de escribirse en el disco y se descifran antes de volver a leerse en la memoria. A diferencia de Azure Storage, TDE debe habilitarse manualmente para Azure Synapse SQL. Pero para otras tecnologías SQL, como Azure SQL, está habilitada por defecto.

Puede habilitar TDE en Azure Synapse SQL desde la pantalla de SQL pool en la pestaña Transparent data encryption:

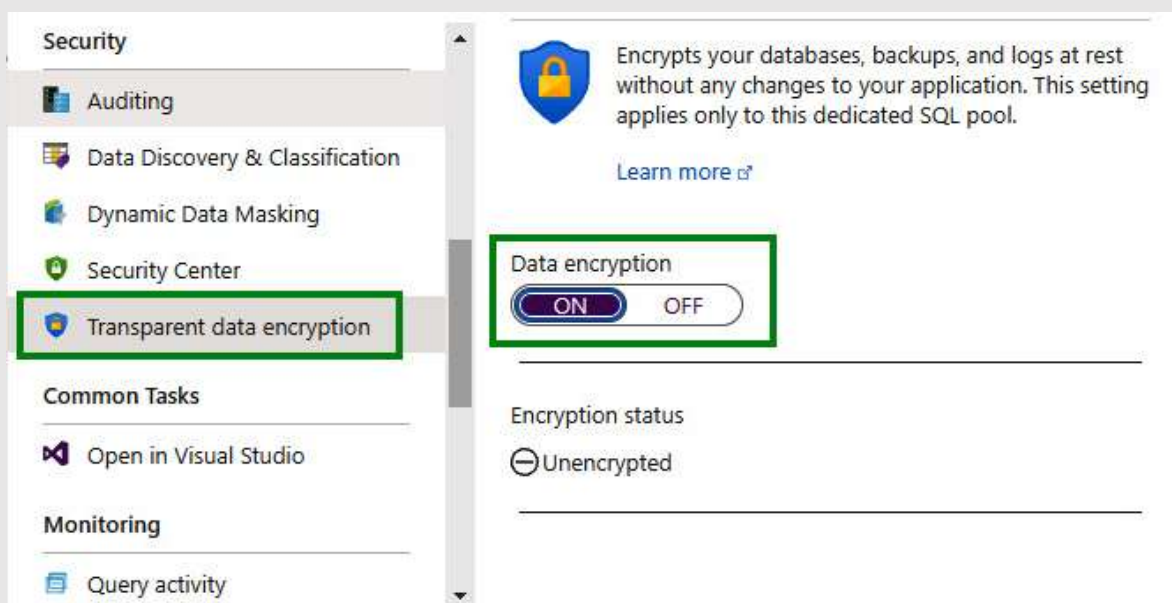


Figura 12.2 - Habilitación de TDE mediante el portal de Azure

También puede habilitar TDE ejecutando la siguiente sentencia como admin user en la terminal de Azure Synapse SQL:

```
ALTER DATABASE <TABLENAME> SET ENCRYPTION ON;
```

#### NOTA:

TDE cifra la base de datos y la protege contra el robo de datos cifrando también los archivos de copia de seguridad y de snapshot.

Puede obtener más información sobre TDE aquí: <https://docs.microsoft.com/en-us/azure/synapse-analytics/sql-data-warehouse/sql-data-warehouse-encryption-tde-tsql>.

Azure Synapse SQL también ofrece la opción de que los clientes traigan sus propias claves de cifrado. Si necesita configurar una Clave Administrada por el Cliente (CMK), deberá habilitar la doble encriptación usando una CMK durante la creación de un área de trabajo de Synapse misma, como se muestra en la siguiente captura de pantalla.



## Workspace encryption



Double encryption configuration cannot be changed after opting into using a customer-managed key at the time of workspace creation.

Choose to encrypt all data at rest in the workspace with a key managed by you (customer-managed key). This will provide double encryption with encryption at the infrastructure layer that uses platform-managed keys. [Learn more](#)

Double encryption using a customer-managed key

☒ Enable ☐ Disable

Encryption key \*

☒ Select a key ☐ Enter a key identifier

Key vault and key \*

Select key vault and key



Azure Key Vaults in the same region as the workspace will be listed.

Managed identity \*

☒ User assigned ☐ System assigned

User assigned identity \*

Select user assigned identity

Figura 12.3 - Configurar una CMK en Azure Synapse SQL

Puedes aprender mas sobre CMKs con TDE aqui: <https://docs.microsoft.com/en-us/azure/azure-sql/database/transparent-data-encryption-tde-overview?tabs=azure-portal#customer-managed-transparent-data-encryption---bring-your-own-key>.

Veamos a continuación la característica Always Encrypted de Azure SQL.

### [Always Encrypted \(siempre encriptado\)](#)

Always Encrypted es una característica proporcionada por Azure SQL y las bases de datos de SQL Server para cifrar las columnas seleccionadas de la base de datos utilizando los drivers de cliente. El driver de cliente Always Encrypted obtiene la clave de cifrado de una ubicación segura como Azure Key Vault para cifrar o descifrar los datos de la columna especificada. Como la clave de cifrado nunca está disponible para el motor de la base de datos, los administradores de la base de datos no pueden acceder a los datos; sólo los propietarios de los datos que tienen acceso a las claves de cifrado podrán acceder a los datos.

Hay dos tipos de claves utilizadas para el cifrado permanente:

- **Column encryption key** - La clave que se utiliza para cifrar/descifrar una columna
- **Column master key** - La clave de protección para encriptar las column encryption keys

A continuación se muestra un código de ejemplo para cifrar las dos columnas Email y SSN de una tabla Cliente:

```
CREATE COLUMN MASTER KEY CMK
WITH (
    KEY_STORE_PROVIDER_NAME = 'AZURE_KEY_VAULT',
    KEY_PATH = 'KeyVault/key/path'
);
-----
CREATE COLUMN ENCRYPTION KEY CEK
WITH VALUES (
    COLUMN_MASTER_KEY = CMK,
    ALGORITHM = 'RSA_OAEP',
    ENCRYPTED_VALUE = 0x020002134.....
);
-----
CREATE TABLE Customer (
    [name] VARCHAR(30),
    [email] VARCHAR(10)
        COLLATE Latin1_General_BIN2 ENCRYPTED WITH (COLUMN_ENCRYPTION_KEY = CEK,
        ENCRYPTION_TYPE = RANDOMIZED,
        ALGORITHM = 'AEAD_AES_256_CBC_HMAC_SHA_256'),
    [phone] VARCHAR (12),
    [SSN] VARCHAR (11)
        COLLATE Latin1_General_BIN2 ENCRYPTED WITH (COLUMN_ENCRYPTION_KEY = CEK,
        ENCRYPTION_TYPE = DETERMINISTIC ,
        ALGORITHM = 'AEAD_AES_256_CBC_HMAC_SHA_256'),
);
```

La opción **DETERMINISTIC** especificada para el correo electrónico asegura que el driver cliente siempre genera el mismo valor encriptado para el texto plano dado. La opción **RANDOMIZED**, por otro lado, genera un valor encriptado diferente cada vez.

## CONSEJO

Si planea utilizar la columna encriptada en JOINS, INDEXES, AGREGADOS, etc., utilice el tipo Deterministic y no un tipo random.

Hay cuatro permisos de base de datos que son necesarios para Always Encrypted.

- **ALTER ANY COLUMN MASTER KEY** - Para crear y borrar column master keys
- **ALTER ANY COLUMN ENCRYPTION KEY** - Para crear y borrar column encryption keys
- **VIEW ANY COLUMN MASTER KEY DEFINITION** - Para leer las column master keys para consultar las columnas encriptadas
- **VIEW ANY COLUMN ENCRYPTION KEY DEFINITION** - Para leer las column master keys para consultar las columnas encriptadas

#### NOTA

Dado que el cifrado y el descifrado se realizan mediante un driver de cliente, las operaciones del lado del servidor, como SELECT INTO, UPDATE y BULK INSERT, no funcionarán con las columnas Always Encrypted.

Puede obtener más información sobre Always Encrypted aquí: <https://docs.microsoft.com/en-us/sql/relational-databases/security/encryption/always-encrypted-database-engine?view=sql-server-ver15>.

A continuación, veamos el cifrado en tránsito.

#### 12.3.2. Cifrado en tránsito

Otro concepto de seguridad importante es el cifrado en tránsito. Se trata de cifrar los datos que se envían por cable o, en otras palabras, cualquier dato que se mueva de un lugar a otro. Ejemplos de movimiento de datos podrían ser los datos que lee una aplicación, los datos que se replican a una zona diferente o los datos que se descargan de la nube. Salvaguardar los datos durante la transferencia es tan importante como mantenerlos seguros mientras se almacenan.

El cifrado en tránsito suele realizarse mediante dos protocolos, Secure Sockets Layer (SSL) o Transport Layer Security (TLS). El soporte de SSL está siendo descontinuado para algunos servicios de Azure, por lo que TLS es el protocolo de red preferido para encriptar datos durante el tránsito.

Conozcamos cómo Azure Storage y los pools de Synapse SQL proporcionan cifrado en tránsito.

#### Habilitación del cifrado en tránsito para Azure Storage

Veamos cómo habilitar TLS en Azure Storage.

Puedes ir a la página de inicio de Blob o ADLS Gen2 storage y seleccionar la pestaña Configuration en Settings. En la página de configuración, podrás configurar la versión mínima de TLS. La

recomendación es ir con la versión 1.2 de TLS. Aquí hay una captura de pantalla de la página de la versión TLS.

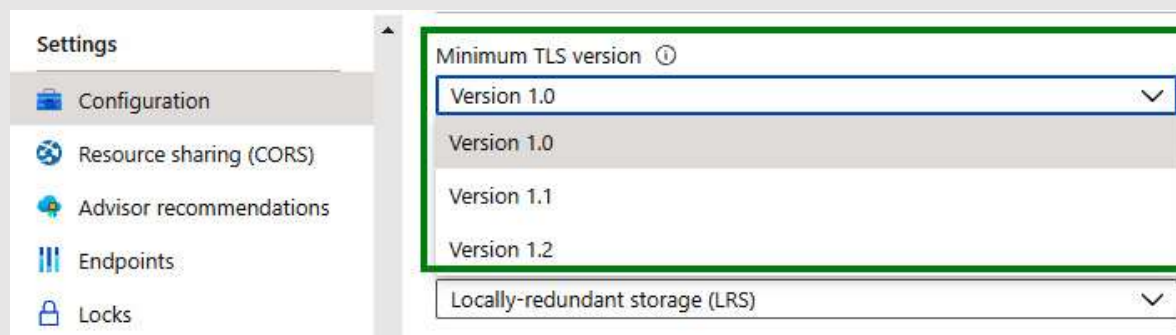


Figura 12.4 - Habilitación de TLS en Azure Storage

Así es como podemos habilitar TLS para Azure Storage. A continuación, vamos a ver cómo configurar TLS en un pool de Synapse SQL.

#### Habilitar el cifrado en tránsito para Azure Synapse SQL

Azure Synapse SQL asegura automáticamente los datos en tránsito utilizando protocolos TLS. De hecho, Synapse SQL impone la encriptación de todas las conexiones independientemente de la configuración de Encrypt o TrustServerCertificate en la cadena de conexión. Por lo tanto, no necesitamos hacer ninguna configuración adicional de nuestro lado.

Puedes aprender más sobre la protección de la información en Azure Synapse SQL y otras variantes de Azure SQL aquí: <https://docs.microsoft.com/en-us/azure/azure-sql/database/security-overview>.

Aparte del cifrado, las otras formas de asegurar los datos mientras están en tránsito es configurar redes privadas virtuales (VPN) dedicadas o utilizar Azure ExpressRoute.

Puede encontrar más información sobre las VPN aquí: <https://docs.microsoft.com/en-us/azure/vpn-gateway/>.

**ExpressRoute** proporciona una conexión privada desde su red local a la nube de Azure. Utiliza proveedores de conectividad para la conexión y no utiliza la Internet pública. Por ello, las conexiones son rápidas, fiables y seguras.

Puede encontrar más información sobre Azure ExpressRoute aquí: <https://docs.microsoft.com/en-us/azure/expressroute/>.

Veamos a continuación las estrategias de auditoría de datos.

## 12.4. Diseñar e implementar una estrategia de auditoría de datos

Este era el segundo requisito de nuestro ejemplo de requisitos de IAC: realizar un seguimiento de las actividades en el almacén de datos con fines de cumplimiento. La auditoría de datos es el proceso de seguimiento de las actividades realizadas en un servicio. Esto se suele hacer mediante registros y métricas. Veamos cómo Azure Storage soporta la auditoría de datos.

### 12.4.1. Auditoría de almacenamiento

Azure Storage soporta el registro de auditoría a través del registro de Storage Analytics. Esto se llama ahora classic monitoring. Hay una versión más nueva de registro disponible en Azure Monitor. El soporte de almacenamiento de Azure Monitor estaba en modo de vista previa en el momento de escribir este libro. Veamos ambas formas de habilitar el registro de auditoría. El registro de Storage Analytics puede ser habilitado como se muestra en la siguiente captura de pantalla:

Usando una configuración de diagnóstico clásica

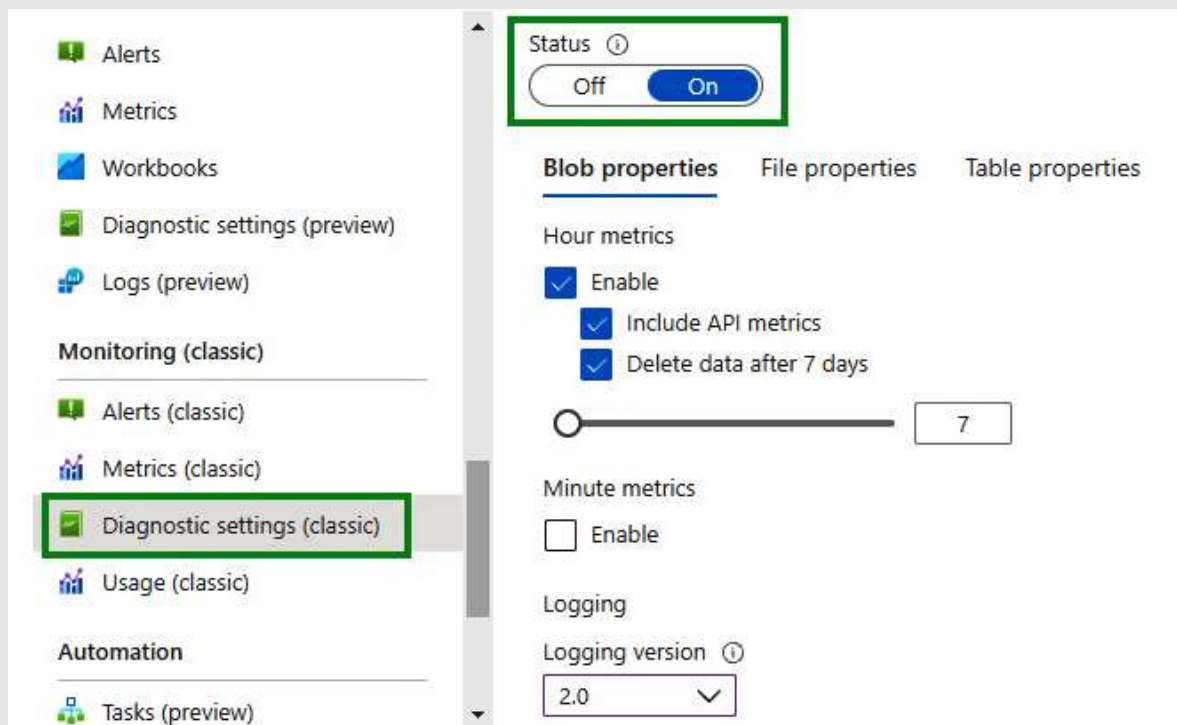


Figura 12.5 - Habilitación del registro de métricas para la auditoría en el almacenamiento de Azure

Una vez habilitado, los registros se almacenarán en un contenedor llamado \$logs bajo su cuenta de almacenamiento. Puede utilizar cualquier explorador de datos o herramientas de procesamiento de datos para ver y analizar los datos de esta carpeta.

## NOTA IMPORTANTE

Los registros pueden tardar hasta una hora en aparecer en el contenedor \$logs. Storage Analytics no vacía los registros inmediatamente. Lo hace a intervalos regulares o cuando tiene suficientes datos para descargarlos en el blob.

Puedes aprender más sobre el registro de Azure Storage Analytics aquí: <https://docs.microsoft.com/en-us/azure/storage/common/storage-analytics-logging>.

A continuación, vamos a ver cómo lograr esto usando Azure Monitor.

### Usando Azure Monitor

Azure Monitor es el servicio de monitorización completo de Azure. Está habilitado por defecto para la mayoría de los servicios de Azure. Recoge ciertas métricas y registros por defecto para cada servicio y puede ser configurado para recoger registros y métricas más detalladas según sea necesario. En el caso de Azure Storage, Azure Monitor comienza a recopilar métricas y registros una vez que habilitamos la configuración de Diagnóstico en la pantalla de Azure Storage en el portal de Azure, como se muestra en la siguiente captura de pantalla:

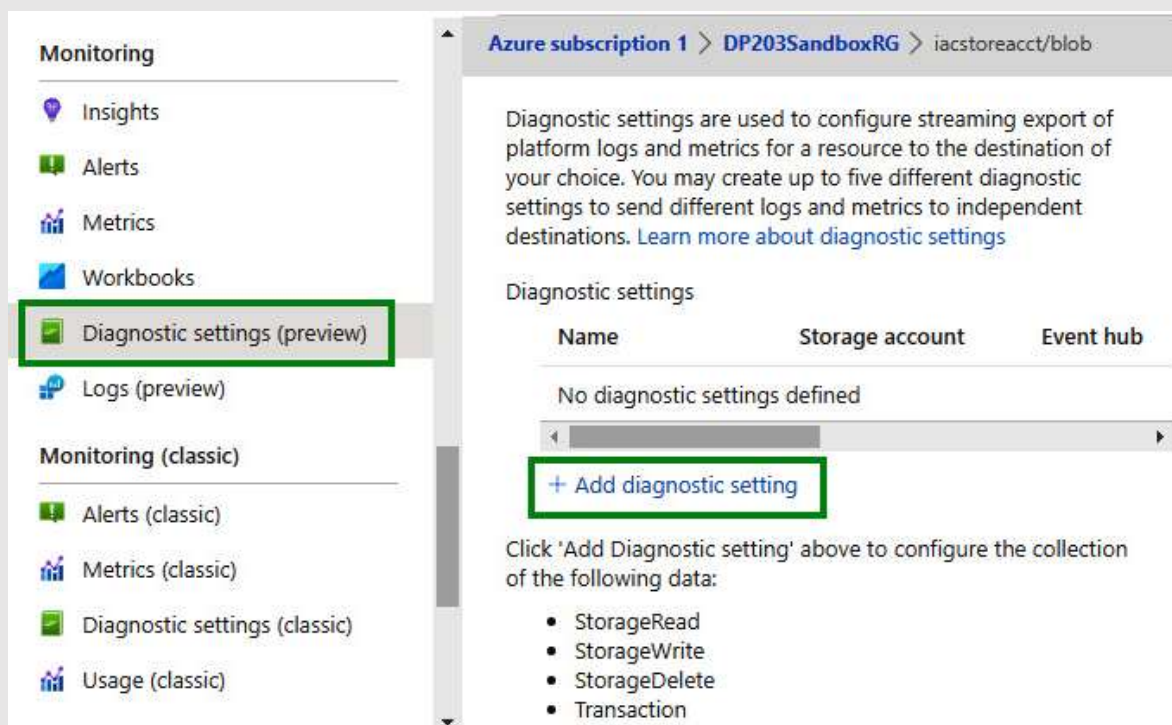






Figura 12.6 - Azure Monitoring - Configuración de diagnóstico

Haga clic en el enlace + Añadir configuración de diagnóstico para configurar la configuración de diagnóstico, como se muestra en la siguiente captura de pantalla:

## Diagnostic setting

 Save  Discard  Delete  Feedback

A diagnostic setting specifies a list of categories of platform logs and/or metrics that you want to collect from a resource, and one or more destinations that you would stream them to. Normal usage charges for the destination will occur. [Learn more about the different log categories and contents of those logs](#)

Diagnostic setting name \*

### Category details

log

☐ StorageRead

☐ StorageWrite

☐ StorageDelete

metric

☐ Transaction

Destination details

☐ Send to Log Analytics workspace

☐ Archive to a storage account

☐ Stream to an event hub

☐ Send to partner solution

Figura 12.7 - Configuración de un ajuste de diagnóstico mediante la monitorización de Azure

En la pantalla de configuración de diagnóstico, puede especificar qué registros y métricas desea registrar y a qué ubicación/herramienta enviarlos. Azure Monitor comenzará a grabar y enviar los registros y métricas a partir de ese momento al servicio configurado. También puedes seleccionar el destino al que quieres enviar los logs. Recuerda que habrá un coste asociado al almacenamiento de los logs en cualquiera de estos destinos, similar al coste de almacenar cualquier otro dato en ellos.

Puedes aprender más sobre la monitorización de Azure para el almacenamiento aquí: <https://docs.microsoft.com/en-us/azure/storage/blobs/monitor-blob-storage>.

Veamos a continuación la auditoría en SQL.

#### 12.4.2. Auditoría en SQL

Azure Synapse SQL proporciona la opción de rastrear todos los eventos y actividades de la base de datos a través de su función de Auditoría. Puedes habilitar fácilmente la auditoría desde la pestaña Azure SQL Auditing en la página del portal de SQL pool. Un ejemplo se muestra en la siguiente



captura de pantalla. Ofrece múltiples opciones de almacenamiento, como Azure Storage, Log Analytics y Event Hub como destino de los registros de auditoría. Puede configurar el destino según sus necesidades empresariales. Por ejemplo, si desea realizar un análisis en tiempo real de los registros de auditoría, puede enviarlos al centro de eventos. Si desea almacenarlos con fines de cumplimiento, puede optar por almacenarlos en Azure Blob storage. Si quieres hacer alguna consulta y análisis ad hoc, entonces usa Log Analytics. Aprenderemos más sobre Log Analytics más adelante en este capítulo.

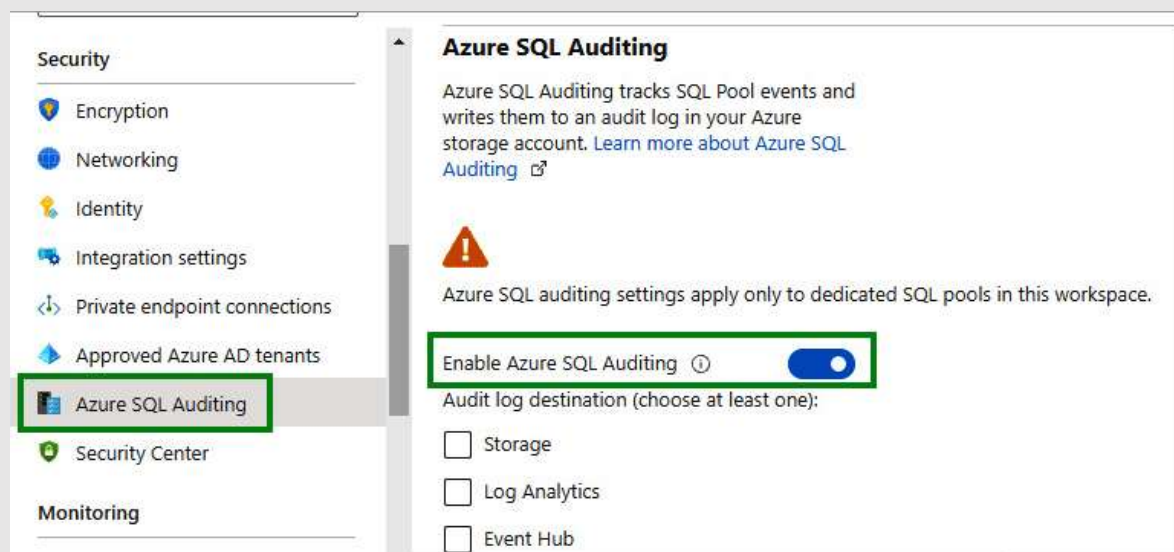


Figura 12.8 - Configuración de Azure Synapse SQL Auditing

Puedes aprender más sobre SQL Auditing aquí: <https://docs.microsoft.com/en-us/azure/azure-sql/database/auditing-overview>.

Veamos a continuación las estrategias de enmascaramiento de datos.



## 12.5. Diseño e implementación de una estrategia de enmascaramiento de datos

El enmascaramiento de datos es una técnica utilizada en las tecnologías SQL para ocultar los datos sensibles en los resultados de las consultas SQL de los usuarios sin privilegios. Por ejemplo, la información de la tarjeta de crédito de un cliente puede enmascarse como XXXX-XXXX-XXXX-1234 en lugar de mostrar el número completo al consultar una tabla de clientes en Synapse SQL. Los datos en sí no se modifican en las tablas, pero las consultas y las vistas modifican los datos dinámicamente para enmascarar la información sensible.

Esta característica ayuda a hacer cumplir los dos siguientes requisitos de IAC:

- No todo el mundo debe tener acceso a todos los datos: debe ser en función de la necesidad de conocerlos.
- Mantener la privacidad del cliente a toda costa.

Puede crear fácilmente una máscara de datos en Azure Synapse SQL (y también en Azure SQL) utilizando una función llamada Dynamic Data Masking (DDM). La siguiente captura de pantalla muestra como se puede hacer esto en Azure Synapse SQL:

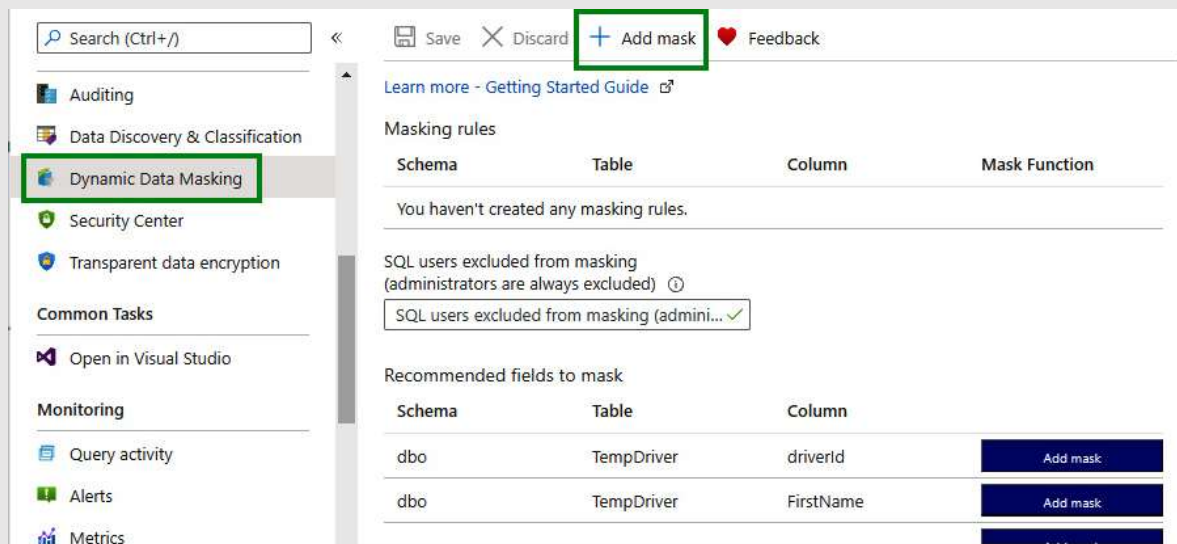


Figura 12.9 - Configuración de DDM desde el portal de Azure

Desde la pantalla anterior, puede hacer clic en el enlace + Añadir máscara para crear una nueva máscara. Por ejemplo, si quieres crear una máscara de correo electrónico, sólo tienes que seleccionar la opción Email de la lista desplegable, como se muestra en la siguiente captura de pantalla:

**Add masking rule** ...

Add
 Delete

Mask name

dbo\_CustomerContact\_Email

Select what to mask

Schema \*

dbo

Table \*

CustomerContact

Column \*

Email (varchar)

Select how to mask

Masking field format

Email (aXXX@XXXX.com)

Default value (0, xxxx, 01-01-1900)

Credit card value (xxxx-xxxx-xxxx-1234)

Email (aXXX@XXXX.com)

Number (random number range)

Custom string (prefix [padding] suffix)

Figura 12.10 - Creación de una máscara de correo electrónico

La captura de pantalla anterior también muestra otras opciones que puede utilizar, como Valor de la tarjeta de crédito, Número y Cadena personalizada.

También puede configurar DDM utilizando T-SQL en Azure Synapse SQL, como se muestra aquí:

```
ALTER TABLE dbo.DimCustomer
ALTER COLUMN emailId ADD MASKED WITH (FUNCTION = 'email()');
```

## NOTA

DDM no cifra la columna. Sólo enmascara los valores durante las consultas.

Puedes aprender más sobre DDM aquí: <https://docs.microsoft.com/en-us/azure/azure-sql/database/dynamic-data-masking-overview>.

Veamos a continuación las políticas RBAC y ACL de Azure, que también se ocupan de un requisito similar de restringir el acceso a los datos.

## 12.6. Diseño e implementación del control de acceso basado en roles de Azure y una lista de control de acceso tipo POSIX para Data Lake Storage Gen2

Esta sección también se ocupa de restringir el acceso a los datos a usuarios no autorizados y satisface el siguiente requisito de nuestros requisitos de IAC de muestra:

*No todo el mundo debería tener acceso a todos los datos. Debe ser sobre la base de la necesidad de conocer.*

Azure utiliza y recomienda el principio de mínimo privilegio, que significa asignar el menor privilegio posible requerido para realizar una tarea. Veamos cómo RBAC y ACLs ayudan a lograr este objetivo.

### 12.6.1. Restricción del acceso mediante Azure RBAC

Azure Role-Based Access Control (Azure RBAC) es un sistema de autorización que controla quién puede acceder a qué recursos en Azure. Azure RBAC trabaja mano a mano con Azure AAD. Intentemos entender los fundamentos de RBAC antes de entrar en los detalles.

RBAC tiene tres componentes:

- **Security principal:** Puede ser cualquier usuario, grupo o identidad gestionada (cuentas de servicio cuyo ciclo de vida está completamente gestionado por Azure) creado dentro de AAD. Puedes pensar en el principal de servicio como la parte de "quién" de la autorización. Es la entidad para la que estamos solicitando permiso. Pueden ser personas reales o cuentas de servicio que se utilizan para ejecutar servicios de forma automática sin intervención humana.
- **Rol:** Piensa en los ejemplos de roles de administrador o roles de invitado de sólo lectura que has utilizado para entrar en cualquier sistema. Un rol de administrador habría tenido acceso completo para crear, leer, escribir, borrar, etc., mientras que una cuenta de invitado podría haber tenido sólo acceso de lectura. Un rol básicamente define qué acciones puede realizar un usuario. Azure tiene una enorme lista de roles predefinidos, como Propietario, Colaborador y Lector, con la lista correcta de permisos ya asignados. Por lo tanto, puedes optar por utilizar uno de ellos en lugar de crear un nuevo rol.
- **Alcance:** El alcance se refiere a todos los recursos a los que se debe aplicar el rol. ¿Quiere que las reglas se apliquen sólo a un grupo de recursos? ¿Sólo a un contenedor en almacenamiento? ¿A varios contenedores? Y así sucesivamente.

Para definir una regla RBAC, necesitamos definir los tres puntos anteriores y asignar un rol y un alcance al principal de seguridad.

Ahora, veamos cómo lograr esto para un data lake. Desde la página de inicio de Azure Storage, selecciona Control de Acceso (IAM). Allí, puedes añadir asignaciones de roles, como se muestra en la siguiente captura de pantalla. Puedes seleccionar el rol desde el campo Rol, el principal de seguridad desde el campo Asignar acceso a, y finalmente, el ámbito, en este caso, sería la propia cuenta de Storage:

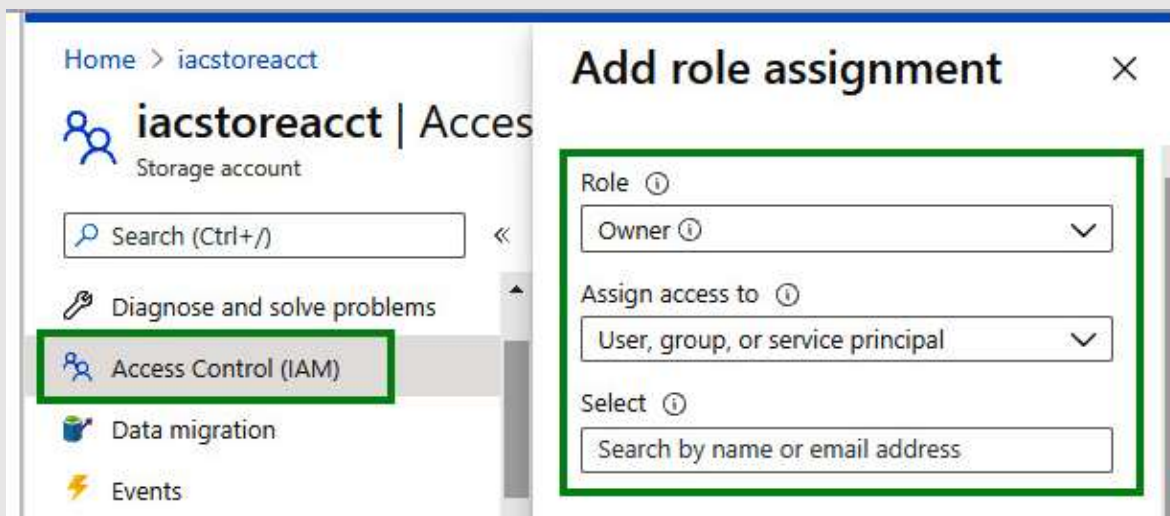


Figura 12.11 - Configuración de la asignación de roles RBAC en ADLS Gen2

Puedes aprender más sobre Azure RBAC aquí: <https://docs.microsoft.com/en-us/azure/role-based-access-control/>.

Veamos a continuación las Listas de Control de Acceso (ACLs).

#### 12.6.2. Restricción de acceso mediante ACLs

Mientras que Azure RBAC proporciona un acceso de grano grueso, como quién puede leer/escribir datos en una cuenta, las ACL proporcionan un acceso de grano más fino, como quién puede leer datos de un directorio específico o un archivo. RBAC y ACL se complementan entre sí para proporcionar un amplio espectro de control de acceso.

Cada directorio y archivo en Azure Storage tiene una ACL. Puede asignar cualquiera de (o todos) los permisos de lectura, escritura y ejecución a directores de seguridad individuales (usuarios) o grupos para proporcionarles el acceso necesario al archivo o directorio. Las ACLs están habilitadas por defecto para ADLS Gen2.

Así es como podemos asignar ACLs en ADLS Gen2. Simplemente haga clic con el botón derecho del ratón en el nombre del archivo o carpeta y seleccione Manage ACL:

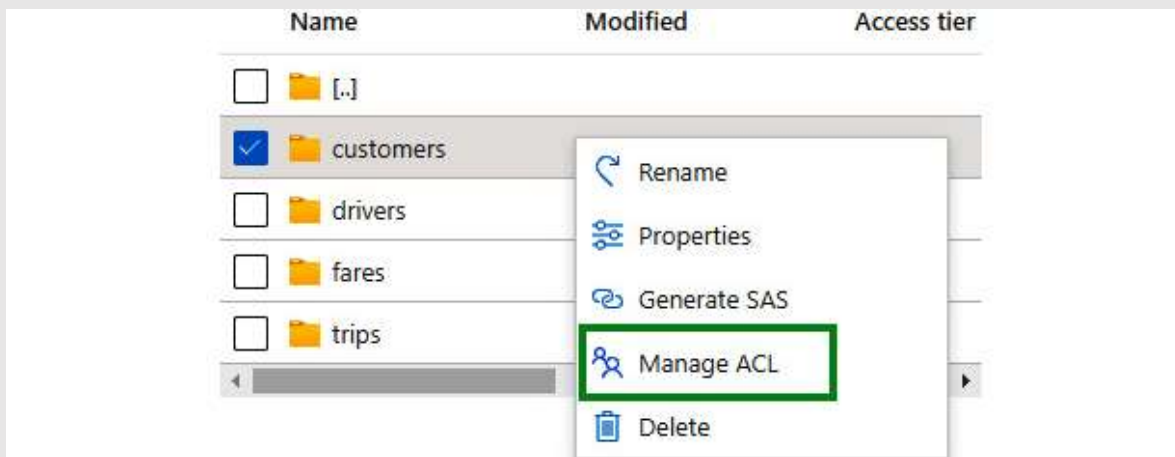


Figura 12.12 - Haga clic con el botón derecho en los archivos o carpetas para seleccionar Manage ACL

En la pantalla Manage ACL, puede asignar acceso de Lectura, Escritura y Ejecución a los principales bajo Security principal, como se muestra en la siguiente captura de pantalla:

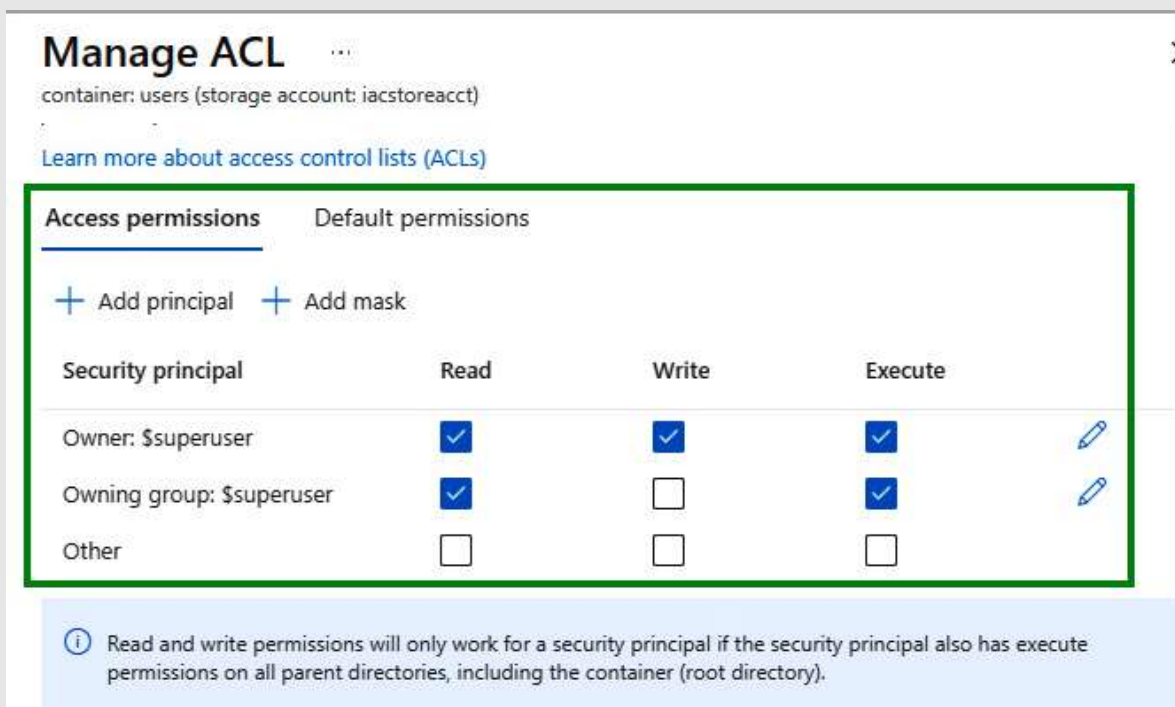


Figura 12.13 - Configuración de la ACL en ADLS Gen2

Puede configurar el nivel de acceso adecuado para los usuarios según sus necesidades. Veamos a continuación el orden en que Azure evalúa el RBAC y las ACL.

¿Cómo decide Azure entre RBAC y ACLs si hay reglas conflictivas?

A continuación se reproduce un diagrama de flujo de Azure que muestra cómo se toma la decisión de autorización entre RBAC y ACL:

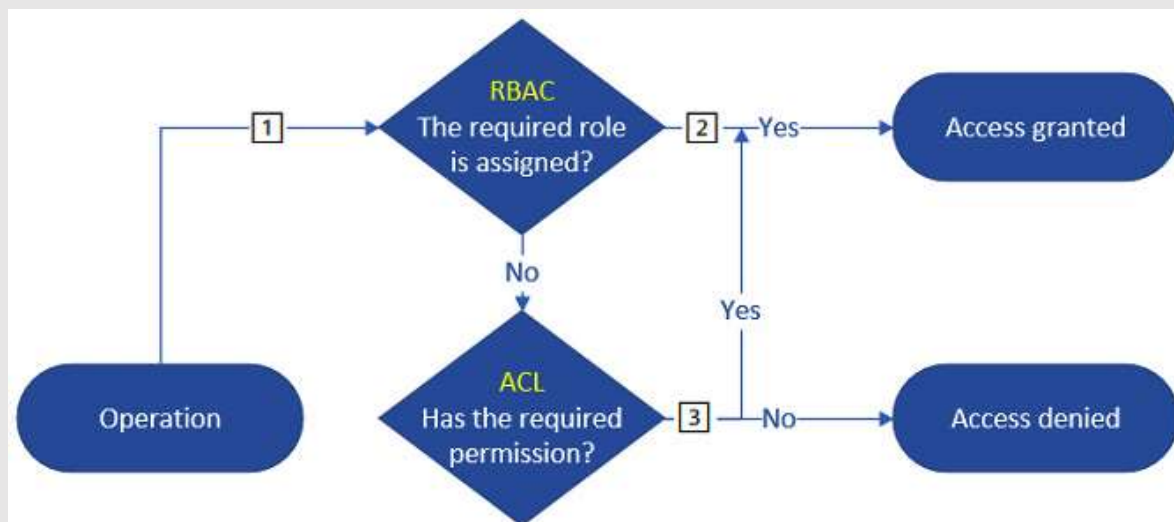


Figura 12.14 - Secuencia de evaluación de RBAC y ACL

En caso de conflicto, Azure da prioridad al RBAC. Así que ten en cuenta esta regla de prioridad mientras diseñas los aspectos de seguridad/cumplimiento de tu data lake.

A continuación vamos a conocer algunas de las limitaciones de RBAC y ACLs.

#### Limitaciones de RBAC y ACL

Aquí hay algunas otras restricciones en RBAC y ACL que debe tener en cuenta al diseñar los requisitos de seguridad y privacidad:

- Azure RBAC permite 2.000 asignaciones de roles por suscripción.
- ACL permite hasta 32 entradas ACL por archivo y directorio. Este número es un poco restrictivo, así que asegúrese de no acabar añadiendo demasiados usuarios individuales; en su lugar, cree grupos y añada sólo grupos a las ACL.

Puede obtener más información sobre las ACL aquí: <https://docs.microsoft.com/en-us/azure/storage/blobs/data-lake-storage-access-control>.

Azure también admite otros dos métodos de autenticación, denominados autorización de clave compartida y autorización de firma de acceso compartida (SAS). La autorización de clave compartida implica compartir una clave de acceso, que básicamente da acceso de tipo administrativo al recurso a cualquiera que posea la clave. Las SAS son ligeramente mejores porque puedes definir qué acciones están permitidas con la clave SAS.

Si utilizas los métodos de autorización Shared Key o SAS, anularán tanto el RBAC como las ACL. La recomendación es utilizar RBAC y ACLs de AAD siempre que sea posible. Aprenderemos más sobre las claves SAS más adelante en este capítulo.

Vamos a aprender sobre la seguridad a nivel de fila y columna en Azure Synapse SQL. Dado que esto también trata de restringir el acceso a los datos, lo he agrupado junto con las secciones RBAC y ACL.



## 12.7. Diseñar e implementar la seguridad a nivel de filas y columnas

Azure Synapse SQL (y Azure SQL) proporcionan una seguridad de grano fino muy útil a nivel de filas y columnas en una tabla. Aprendamos a utilizar estas funciones para restringir el acceso a los datos, empezando por la seguridad a nivel de filas.

### 12.7.1. Diseño de la seguridad a nivel de filas

La seguridad a nivel de filas restringe el acceso a ciertas filas de la tabla a usuarios no autorizados. En un nivel alto, puede pensar en esto como algo similar al uso de condiciones WHERE en una sentencia SELECT. La seguridad a nivel de fila se logra creando políticas de seguridad. Veremos un ejemplo de cómo crear una regla de este tipo en las próximas páginas. Estas reglas residen en la propia base de datos. Por lo tanto, independientemente de cómo se acceda a los datos, ya sea a través de consultas, vistas o cualquier otro método, se aplicará la restricción de acceso a los datos.

Veamos un ejemplo utilizando de nuevo nuestro escenario de IAC. Imaginemos que la empresa IAC está intentando lanzar su servicio en un grupo de nuevas ubicaciones, pero quieren mantener los detalles en secreto ya que no quieren que se filtre la noticia. Así que definen dos conjuntos de usuarios, uno llamado HiPriv\_User que tiene acceso a todas las filas y otro llamado LowPriv\_Users que no tiene acceso a todas las filas. Veamos cómo implementar este ejemplo en Azure Synapse SQL. Inicie un Synapse dedicated pool y abra un editor desde el área de trabajo de Synapse:

Cree un nuevo esquema para almacenar nuestra política de acceso a las filas:

```
CREATE SCHEMA Security;
```

Crea una función T-SQL que tenga la lógica para decidir quién tiene acceso a los datos de pre-lanzamiento. En este caso, podemos asumir que todos los tripId >= 900 son las ubicaciones de pre-lanzamiento:

```
CREATE FUNCTION Security.tvf_securitypredicate(@tripId AS int)
    RETURNS TABLE
WITH SCHEMABINDING
AS RETURN SELECT 1 AS tvf_securitypredicate_result
WHERE @tripId < 900 OR USER_NAME() = 'HiPriv_User';
```

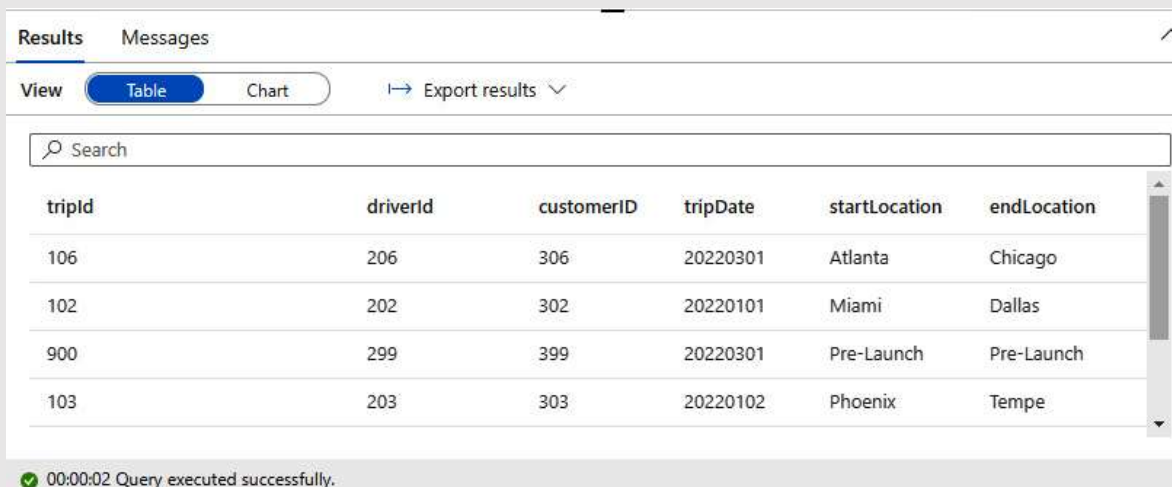
Cree una política de seguridad utilizando la función definida anteriormente:

```
CREATE SECURITY POLICY PrivFilter
ADD FILTER PREDICATE Security.tvf_securitypredicate(tripId)
ON dbo.TripTable WITH (STATE = ON);
```

Ahora, pruébalo con HiPriv\_User:

```
EXECUTE AS USER = 'HiPriv_User';
SELECT * from dbo.TripTable
```

Cuando se ejecuta como HiPriv\_User, todas las filas, incluyendo las filas de pre-lanzamiento con ID >= 900, aparecen, como se muestra en la siguiente captura de pantalla:



Results Messages

View Table Chart Export results

Search

tripId	driverId	customerID	tripDate	startLocation	endLocation
106	206	306	20220301	Atlanta	Chicago
102	202	302	20220101	Miami	Dallas
900	299	399	20220301	Pre-Launch	Pre-Launch
103	203	303	20220102	Phoenix	Tempe


00:00:02 Query executed successfully.

Figura 12.15 - Aparecen todas las filas, incluyendo las de pre-lanzamiento

Ahora, vamos a probarlo con LowPriv\_User:

```
EXECUTE AS USER = 'LowPriv_User';  
SELECT * from dbo.TripTable
```

Cuando se ejecuta como LowPriv\_User, las líneas previas al lanzamiento se ocultan, como se muestra en la siguiente captura de pantalla:



Results Messages

View Table Chart Export results

Search

tripId	driverId	customerID	tripDate	startLocation	endLocation
106	206	306	20220301	Atlanta	Chicago
102	202	302	20220101	Miami	Dallas
103	203	303	20220102	Phoenix	Tempe
101	201	301	20220101	New York	New Jersey

00:00:02 Query executed successfully.

Figura 12.16 - Seguridad a nivel de fila que bloquea las filas de ubicación de pre-lanzamiento

Espero que se haya hecho una buena idea del concepto de seguridad a nivel de filas. Ahora, veamos la seguridad a nivel de columna.

### 12.7.2. Diseñar la seguridad a nivel de columna

La seguridad a nivel de columna es similar a la función de enmascaramiento de datos que vimos anteriormente en este capítulo en la sección Diseño e implementación de una estrategia de enmascaramiento de datos. Pero en lugar de sólo enmascarar los valores de la columna, aquí restringimos completamente el acceso a la columna a usuarios no autorizados. En el caso de la seguridad a nivel de columna también, las reglas residen en la propia base de datos. Por lo tanto, independientemente de cómo se acceda a los datos -por ejemplo, a través de consultas, vistas o cualquier otro método- se aplicará la restricción de acceso a los datos.

He aquí un ejemplo de cómo implementar las restricciones de columna.

Consideremos nuestro ejemplo de IAC, la tabla de dimensión DimCustomer. Aquí está la definición de la tabla:

```
CREATE TABLE dbo.DimCustomer
(
    [customerId] INT NOT NULL,
    [name] VARCHAR(40) NOT NULL,
    [emailId] VARCHAR(40),
    [phoneNum] VARCHAR(40),
    [city] VARCHAR(40)
)
```

Para restringir el acceso, necesita usar el comando GRANT, como se muestra aquí:

```
GRANT SELECT ON dbo.DimCustomer (customerId, name, city) TO LowPriv_User;
```

Aquí, sólo le damos a LowPriv\_User acceso a las columnas customerId, name y city. LowPriv\_User no tendrá acceso a las columnas emailId o phoneNum. Si ejecuta la siguiente consulta como LowPriv\_User, obtendrá un error:

```
SELECT * FROM Customer;
-- The SELECT permission was denied on the column 'emailId' of the object 'DimCustomer', database 'DedicatedSmall', schema 'dbo'. The SELECT permission was denied on the column 'phoneNum' of the object 'DimCustomer', database 'DedicatedSmall', schema 'dbo'.
```

Las funciones de restricción de filas y columnas son especialmente útiles cuando sólo un subconjunto muy pequeño de los datos de la tabla es sensible. Esto evita la necesidad de dividir la tabla y almacenar los datos sensibles por separado.

Puede obtener más información sobre la seguridad a nivel de columna aquí: <https://docs.microsoft.com/en-us/azure/synapse-analytics/sql-data-warehouse/column-level-security>.

Veamos ahora las políticas de retención de datos.

## 12.8. Diseñar y aplicar una política de conservación de datos

El último requisito de la IAC era eliminar los datos de forma segura para que no lleguen a manos de ningún usuario malintencionado.

*Los datos antiguos deben ser eliminados de forma segura después de un periodo de tiempo.*

En esta sección sobre la retención de datos y en la siguiente sobre la purga de datos se explican algunas de las técnicas que se pueden utilizar para lograr una limpieza regular y segura de los datos.

En el capítulo 2, Diseño de una estructura de almacenamiento de datos, en la sección Diseño de una solución de archivo de datos, aprendimos sobre la gestión del ciclo de vida de los datos. Podemos utilizar el mismo servicio para diseñar nuestras políticas de retención de datos también. La pantalla de gestión del ciclo de vida de los datos tiene opciones para mover los datos a los niveles Cool o Archive o eliminar el propio blob. Veamos una captura de pantalla del Capítulo 2, Diseño de una estructura de almacenamiento de datos, de nuevo por conveniencia:

**Add a rule** ...

✓ Details    2 Base blobs

Lifecycle management uses your rules to automatically move blobs to cooler tiers or to delete them. If you create multiple rules, the associated actions must be implemented in tier order (from hot to cool storage, then archive, then deletion).

+ Add if-then block

**If**

Base blobs were \*

☒ Last modified

More than (days ago) \*

30

**Then**

Delete the blob

**Move to cool storage**  
For infrequently accessed data that you want to keep on cool storage for at least 30 days.

**Move to archive storage**  
Use if you don't need online access and want to keep the object for 180 days or longer.

**Delete the blob**  
Deletes the object per the specified conditions.

Previous    Add

Figura 12.17 - Configuración de la gestión del ciclo de vida de los datos

En función de nuestros requisitos, podemos configurar lotes enteros de datos para que se eliminen después de un tiempo determinado.

Puedes aprender más sobre la gestión del ciclo de vida de los datos en Azure Data Lake Storage aquí: <https://docs.microsoft.com/en-us/azure/storage/blobs/lifecycle-management-overview>.

Veamos a continuación las otras opciones disponibles para purgar datos de los almacenes de data lake y de los almacenes basados en SQL.

## 12.9. Diseño para purgar datos en función de los requisitos del negocio

La purga de datos es otro concepto que se solapa con la retención de datos. La purga de datos se refiere al proceso de eliminación segura de datos según los requisitos del negocio. Veamos las opciones disponibles en Azure Data Lake Storage y Azure Synapse SQL.

### 12.9.1. Purga de datos en Azure Data Lake Storage Gen2

Hay dos maneras en las que podemos lograr la eliminación programada de datos o la purga de datos. Una es la misma que la retención de datos, utilizando la gestión del ciclo de vida en Azure Storage. La segunda técnica es utilizar Azure Data Factory. Cada vez que eliminamos datos en Azure, se asegura que las ubicaciones de almacenamiento se sobrescriben para que los datos eliminados no puedan ser recuperados por un usuario malicioso. Puedes consultar la política de protección de datos de Microsoft aquí: <https://docs.microsoft.com/en-us/azure/security/fundamentals/protection-customer-data>.

Podemos eliminar archivos de Azure Storage de forma periódica o condicional utilizando la actividad Delete en el ADF. Ya hemos visto la actividad Delete como parte de la sección Regressing to a previous stage en el Capítulo 9, Designing and Developing a Batch Processing Solution. Volvamos a verlo aquí con un ejemplo diferente. En este caso, hemos configurado la actividad Eliminar para que borre todos los archivos con más de 30 días de antigüedad:

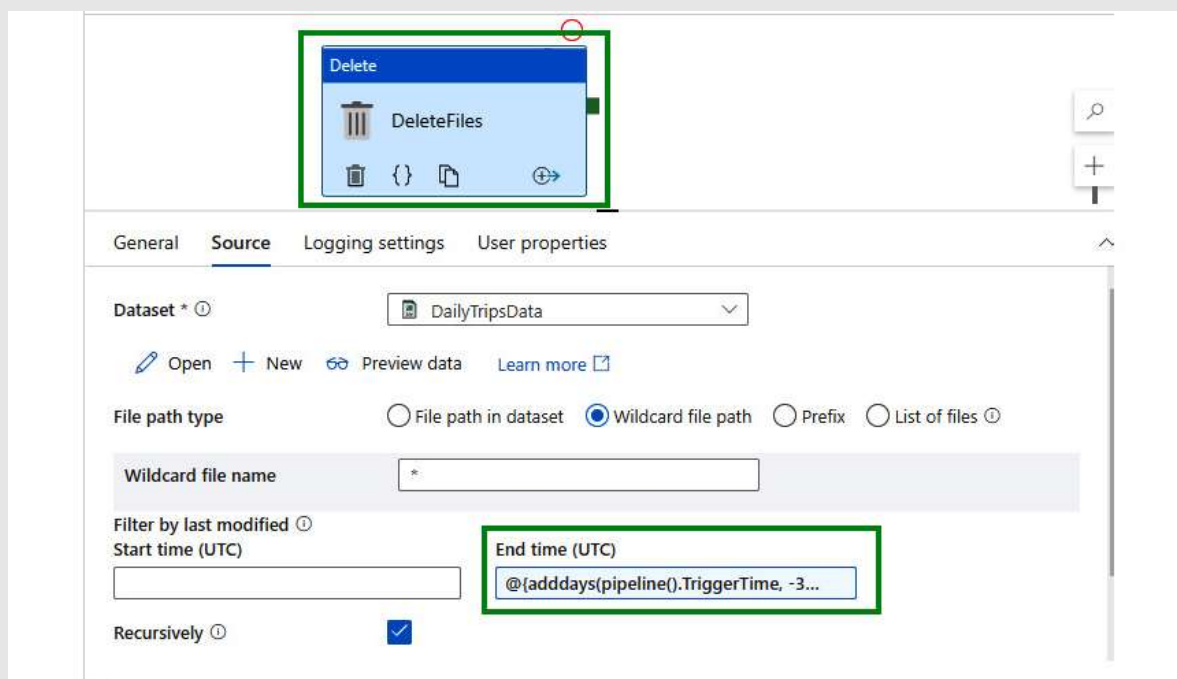


Figura 12.18 - Opciones de eliminación del ADF

En la sección End time (UTC), hemos añadido la siguiente línea para seleccionar todos los archivos de más de 30 días - @{adddays(pipeline().TriggerTime, -30)}. Esto se encarga de eliminar todos los datos de la carpeta configurada que tengan más de 30 días.

Puede obtener más información sobre la actividad ADF Delete aquí: <https://azure.microsoft.com/en-in/blog/clean-up-files-by-built-in-delete-activity-in-azure-data-factory/>.

Veamos a continuación cómo purgar datos en SQL.

#### 12.9.2. Purgar datos en Azure Synapse SQL

La purga en Synapse SQL se hace usando el comando TRUNCATE. Aquí hay un ejemplo de cómo se puede hacer donde dbo.DimCustomer es un nombre de tabla:

```
TRUNCATE TABLE dbo.DimCustomer;
```

Veamos ahora cómo crear y gestionar claves, secretos e identidades gestionadas.

## 12.10. Gestión de identidades, claves y secretos en diferentes tecnologías de plataformas de datos

Existen principalmente dos tecnologías utilizadas en Azure para gestionar identidades, claves, secretos, certificados y, básicamente, cualquier cosa confidencial. Son Azure Active Directory (AAD) y Azure Key Vault. Hemos visto brevemente Azure Key Vault antes en este capítulo en la sección Cifrado en reposo en Azure Storage. Veamos estos dos servicios en detalle aquí.

### 12.10.1. Azure Active Directory

AAD es el servicio de gestión de identidad y acceso de Azure. Soporta la gestión de usuarios, grupos, principales de servicio, etc. Puedes pensar en los service principals como las cuentas de servicio utilizadas para ejecutar aplicaciones automáticamente. Estos service principals también se denominan aplicaciones AAD.

Veamos ahora un ejemplo de creación de usuarios en AAD:

1. Desde el portal de Azure, busca AAD o Azure Active Directory, y selecciona ese servicio.
2. Una vez dentro, puedes hacer clic en la pestaña Usuarios dentro de la categoría Gestionar, y luego seleccionar el enlace + Añadir, como se muestra en la siguiente captura de pantalla:

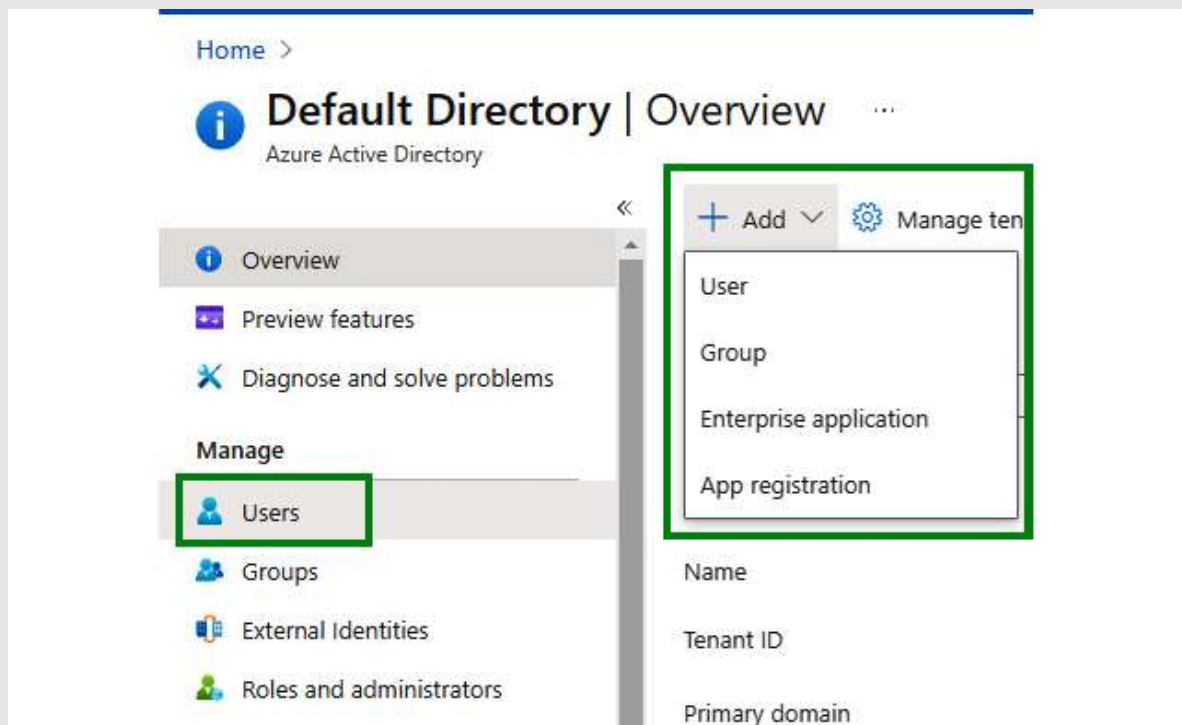


Figura 12.19 - Creación de un nuevo usuario en AAD



Esto abre la pantalla de Nuevo usuario, donde puede añadir los detalles del usuario y hacer clic en el botón Crear para crear un nuevo usuario, como se muestra en la siguiente captura de pantalla:

Home > Default Directory >

## New user

Default Directory

Got feedback?

### Identity

User name \* ⓘ Example: chris @ [Domain] ▼

Name \* ⓘ Example: 'Chris Green'

First name

Last name

### Groups and roles

Groups 0 groups selected

Roles User

Create

Figura 12.20 - Pantalla de creación de nuevos usuarios en AAD

4. Para gestionar los usuarios, puede ir a la pestaña Usuarios donde debería poder ver la lista de usuarios bajo su tenant de AAD. Puede seleccionar usuarios y realizar operaciones como editar, eliminar, etc.

La creación de grupos y aplicaciones es similar a la creación de nuevos usuarios, por lo que no entraremos en detalles al respecto. A continuación, vamos a ver otro concepto importante que los servicios y recursos (como VMs, bases de datos SQL, Synapse SQL, etc.) en Azure utilizan para autenticarse en AAD.

## Identidades gestionadas

Azure AAD admite otra característica importante denominada identidades gestionadas. Se trata de identidades que se asignan a instancias de servicios de Azure como VMs, Synapse, bases de datos SQL, etc. El ciclo de vida de estas identidades es gestionado automáticamente por AAD; de ahí que se llamen identidades gestionadas. Por ejemplo, cuando creamos una nueva área de trabajo de Synapse, ésta tiene una identidad gestionada creada automáticamente en AAD, y cuando se elimina el área de trabajo, la identidad gestionada se elimina automáticamente. Esta identidad gestionada puede ser utilizada por la instancia de Azure Synapse para autenticarse con AAD, en lugar de que los propietarios de la aplicación tengan que almacenar secretos dentro de la aplicación o tener secciones de código separadas para autenticar todos y cada uno de los servicios a AAD. Puedes encontrar el ID de la identidad gestionada en la página de resumen de tu aplicación. Un ejemplo para Synapse se muestra en la siguiente captura de pantalla:

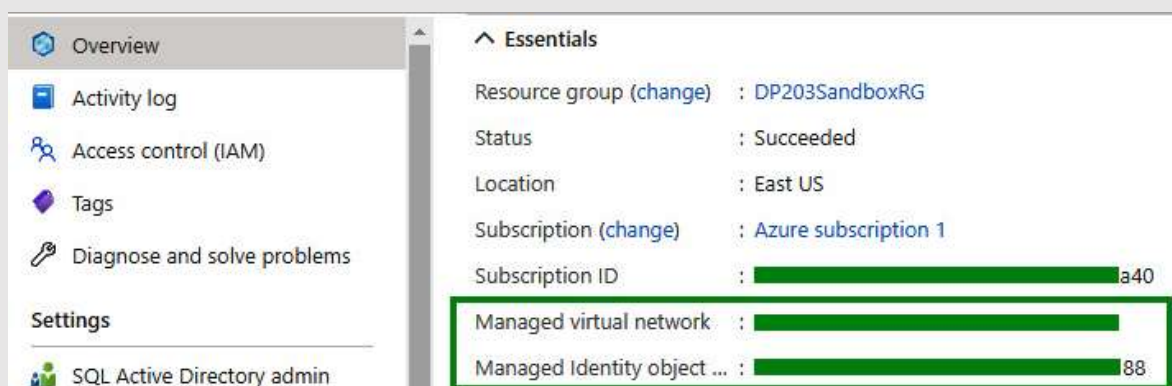


Figura 12.21 - Una identidad gestionada para Synapse

Puede aprender más sobre AAD aquí: <https://docs.microsoft.com/en-us/azure/active-directory/>.

A continuación, vamos a aprender a almacenar claves, secretos y certificados de forma segura utilizando Azure Key Vault.

### 12.10.2. Azure Key Vault

Azure Key Vault es otro servicio muy utilizado en Azure que se usa para almacenar claves, secretos y certificados de forma segura. Una bóveda de claves es como una bóveda del mundo real utilizada para almacenar cosas confidenciales. Azure Key Vault es una versión digital de una bóveda que cifra y descifra la información utilizando un cifrado AES de 256 bits. Proporciona las siguientes funcionalidades:

- **Key Management (Gestión de Claves)**: Ayuda a crear y gestionar las claves de cifrado
- **Secrets Management (Gestión de secretos)**: Ayuda a crear y almacenar secretos, contraseñas, URIs con claves, etc., a los que pueden acceder las aplicaciones y usuarios autorizados utilizando los enlaces de Key Vault

- **Certificate Management (Gestión de certificados)**: Ayuda a crear y gestionar certificados SSL y TLS para los recursos de Azure

Key Vault simplifica la gestión de secretos. Elimina la necesidad de almacenar secretos en código. Es un servicio centralizado, por lo que cada vez que necesitemos cambiar un secreto o una clave de cifrado, sólo tenemos que actualizarlo en Key Vault. También admite la rotación de certificados y claves sin intervención humana. Si tienes algún tipo de secreto en tus aplicaciones, es muy recomendable utilizar Azure Key Vault.

Veamos cómo generar algunas claves y secretos utilizando Key Vault:

1. Desde el portal de Azure, busca Key Vault y selecciona ese servicio.
2. Una vez dentro, selecciona la pestaña Keys y luego el enlace + Generate/Import, como se muestra en la siguiente captura de pantalla:

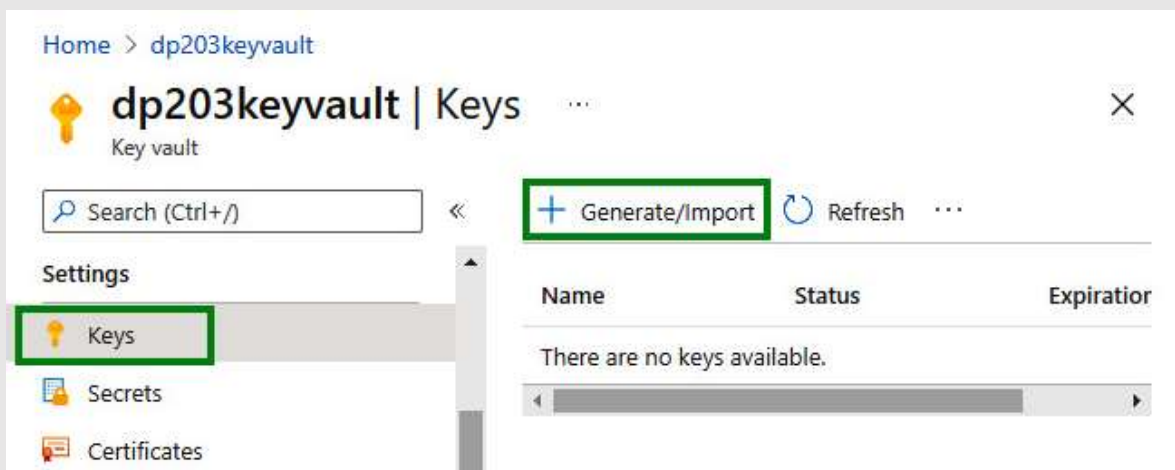


Figura 12.22 - Creación de claves, secretos y certificados en Key Vault

3. Cuando haga clic en el enlace +Generar/Importar, obtendrá la pantalla Crear una clave, como se muestra en la siguiente captura de pantalla:

## Create a key ...

Options	<div>Generate <span>▼</span></div>
Name <span>*</span> ⓘ	<input type="text"/>
Key type ⓘ	<div><input checked="" type="radio"/> RSA <input type="radio"/> EC</div>
RSA key size	<div><input checked="" type="radio"/> 2048 <input type="radio"/> 3072 <input type="radio"/> 4096</div>
Set activation date ⓘ	<input type="checkbox"/>
Set expiration date ⓘ	<input type="checkbox"/>
Enabled	<div>Yes No</div>
Tags	0 tags

Create

Figura 12.23 - Creación de nuevas claves de cifrado en Key Vault

4. Sólo tiene que introducir los detalles y hacer clic en Crear para crear una nueva clave de cifrado. Puede utilizar esta clave de cifrado para cifrar sus cuentas de almacenamiento o bases de datos.
5. De forma similar a las claves, también puedes crear secretos. Sólo tienes que seleccionar la pestaña Secretos en la página de Azure Key Vault y hacer clic en + Generar/Importar; verás la siguiente pantalla:

Home > dp203keyvault >

## Create a secret

Upload options

Manual

Name \* ⓘ

SynapseSecret ✓

Value \* ⓘ

..... ✓

Content type (optional)

Set activation date ⓘ

☐

Set expiration date ⓘ

☐

Create

Figura 12.24 - Creación de nuevos secretos en Key Vault

- Una vez que introduzcas los detalles y hagas clic en Create, se creará un nuevo secreto.

A continuación, vamos a ver cómo utilizar los secretos almacenados en Key Vault en otros servicios como Azure Synapse Analytics.

Para utilizar Key Vault en Synapse Analytics, primero tenemos que añadir el propio Key Vault como uno de los servicios vinculados. Ya hemos visto muchos ejemplos de cómo añadir un servicio vinculado, así que ya deberías estar familiarizado con el proceso. Una vez que tengamos registrado Key Vault como uno de los servicios vinculados, cada vez que añadamos cualquier otro servicio vinculado, tendremos la opción de utilizar los secretos directamente desde Azure Key Vault. La siguiente captura de pantalla muestra un ejemplo de creación de un nuevo servicio vinculado a Azure MySQL Database. Se puede ver la opción de utilizar un secreto de Azure Key Vault en lugar de introducir manualmente los detalles de la cadena de conexión:

### New linked service (Azure Database for MySQL)

**Name \***

**Description**

**Connect via integration runtime \*** ⓘ

**Connection string**

**Azure Key Vault**

**AKV linked service \*** ⓘ

**Secret name \*** ⓘ

Add dynamic content [Alt+Shift+D]

**Secret version** ⓘ

**Annotations**

+ New

▸ Parameters

Create Back Test connection Cancel

Figura 12.25 - Acceso a las claves desde Synapse Analytics

También se puede acceder a Azure Key Vault mediante la línea de comandos:

- También puede establecer una nueva contraseña utilizando la CLI, como se muestra aquí:

```
az keyvault secret set --vault-name "<KEYVAULT-NAME>" --name "SamplePassword" --value "SecretValue"
```

- Puede acceder a sus contraseñas utilizando la siguiente URL:

```
https://<KEYVAULT-NAME>.vault.azure.net/secrets/SamplePassword
```

- Puede ver la contraseña utilizando lo siguiente:

```
az keyvault secret show --name " SamplePassword " --vault-name "<KEYVAULT-NAME>" --query "value"
```

- Puede obtener más información sobre Azure Key Vault aquí:

<https://docs.microsoft.com/en-in/azure/key-vault/>.

Espero que esto te haya dado una comprensión bastante buena de los componentes involucrados en la gestión de identidades, claves y secretos en Azure.

Vamos a aprender un poco más sobre las claves de Azure Storage SAS que introdujimos en la sección RBAC de este capítulo, ya que esto es importante desde una perspectiva de certificación.

### 12.10.3. Claves de acceso y claves de acceso compartidas en Azure Storage

Azure Storage genera dos claves de acceso al almacenamiento de 512 bits cuando creamos una nueva cuenta de almacenamiento. Estas claves se pueden utilizar para acceder a los datos de las cuentas de almacenamiento. Cuando estas claves se utilizan para autenticar el acceso, se denomina método de autenticación con clave de acceso compartida.

Puede ver las claves de acceso desde el portal de almacenamiento. Seleccione la pestaña Seguridad + red y seleccione la opción Claves de acceso.

Las claves de acceso son como las contraseñas de root. Dan acceso completo a todos los recursos de una cuenta de almacenamiento. Por lo tanto, si necesitamos dar acceso restringido a alguien o a alguna aplicación, podemos utilizar la opción Azure RBAC de la que hablamos anteriormente en este capítulo o utilizar una Firma de Acceso Compartido (SAS). Una SAS es una URI que otorga acceso restringido por un período específico, a direcciones IP específicas, permisos específicos, etc. A diferencia de las claves de acceso, las claves SAS no tendrán permisos para modificar o eliminar cuentas.

Existen tres tipos de claves SAS:

- **Servicio SAS:** Este tipo de clave SAS proporciona acceso a uno solo de los servicios de almacenamiento, como Blobs, Tablas, Archivos o Colas. Las claves SAS de servicio se firman con las claves de acceso al almacenamiento.
- **Cuenta SAS:** Este tipo de clave SAS proporciona acceso a múltiples servicios de almacenamiento como Blobs, Tablas, Archivos y Colas. Las claves SAS de cuenta proporcionan acceso a operaciones de lectura, escritura y eliminación en múltiples servicios como contenedores de blobs, tablas, colas y archivos. Las claves SAS de cuenta se firman utilizando las claves de acceso de almacenamiento.

- **User Delegation SAS:** Si la clave SAS está firmada por AAD, el SAS se llama User Delegation SAS. Este es el enfoque de SAS recomendado por Azure. Pero esta opción sólo funciona para el almacenamiento Blob y no para otras opciones de almacenamiento como tablas, colas o servicios de archivos.

## NOTA

Dado que las claves de acceso se utilizan para firmar las claves de acceso compartido, como el servicio SAS y la cuenta SAS, estas claves se invalidarán cuando regenere nuevas claves de acceso para el almacenamiento.

Puedes aprender más sobre las claves SAS y cómo generarlas aquí: <https://docs.microsoft.com/en-us/rest/api/storageservices/authorize-with-shared-key>.

Veamos a continuación cómo utilizar los endpoints privados y públicos.



## 12.11. Implementación de endpoints seguros (privados y públicos)

Un endpoint público se refiere a la forma predeterminada de crear servicios de Azure (como Azure Storage, Azure Synapse y Azure SQL), donde se puede acceder al servicio desde una dirección IP pública. Así, cualquier servicio que crees en Azure sin configurar una red virtual (VNet) entraría en la categoría de endpoint público.

Por otro lado, (como ya habrás adivinado), los endpoints privados son configuraciones más seguras que implican direcciones IP privadas. Un endpoint privado es parte de un servicio más grande llamado servicio Private Link. El servicio Private Link hace que su servicio Azure esté disponible sólo en ciertas direcciones IP privadas dentro de sus VNets. Nadie de fuera de sus VNets será consciente de la existencia de dicho servicio. El punto final privado se refiere técnicamente a la interfaz de red que utiliza la IP privada de su VNet, y el servicio Private Link se refiere al servicio global que comprende los endpoints privados y la red privada ( link ) por la que atraviesa el tráfico. Cuando se establece un link privado a su servicio Azure, todos los datos de ese servicio atraviesan la red troncal de Microsoft sin estar expuestos a la Internet pública.

Veamos cómo crear un private link a una de nuestras áreas de trabajo existentes en Synapse:

1. Crear una VNet. Desde el portal de Azure, selecciona Redes virtuales y haz clic en la opción Crear red virtual. Obtendrás una pantalla como la que se muestra en la siguiente figura. Rellena los datos de esta pantalla:

Home > Virtual networks >

### Create virtual network

Basics IP Addresses Security Tags Review + create

Azure Virtual Network (VNet) is the fundamental building block for your private network in Azure. VNet enables many types of Azure resources, such as Azure Virtual Machines (VM), to securely communicate with each other, the internet, and on-premises networks. VNet is similar to a traditional network that you'd operate in your own data center, but brings with it additional benefits of Azure's infrastructure such as scale, availability, and isolation. [Learn more about virtual network](#)

**Project details**

Subscription \* ⓘ Azure subscription 1

Resource group \* ⓘ [Create new](#)

**Instance details**

Name \*

Region \* East US

[Review + create](#) < Previous Next : IP Addresses >

[Download a template for automation](#)

Figura 12.26 - Creación de una nueva VNet

2. En la pestaña de Direcciones IP, tendrá la posibilidad de especificar el rango de direcciones IP y las subredes, como se muestra en la siguiente captura de pantalla:

Home > Virtual networks >

## Create virtual network

Basics **IP Addresses** Security Tags Review + create

The virtual network's address space, specified as one or more address prefixes in CIDR notation (e.g. 192.168.1.0/24).

**IPv4 address space**

10.0.0.0/16 10.0.0.0 - 10.0.255.255 (65536 addresses)

☐ Add IPv6 address space

The subnet's address range in CIDR notation (e.g. 192.168.1.0/24). It must be contained by the address space of the virtual network.

Add subnet Remove subnet

<input type="checkbox"/> Subnet name	Subnet address range	NAT gateway
<input type="checkbox"/> default	10.0.0.0/24	-

Use of a NAT gateway is recommended for outbound internet access from a subnet. You can

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

[Download a template for automation](#)

Figura 12.27 - Configurar los detalles de la IP para la nueva VNet

3. Rellenamos el resto de pestañas y hacemos clic en Revisar + crear para crear la nueva VNet.
4. A continuación, tendremos que abrir la página del servicio Private Link desde el portal de Azure. Sólo tienes que buscar Private Link y verás el servicio en la parte superior.
5. En la página Private Link Center, seleccionamos la pestaña Private endpoints y hacemos clic en +Add:

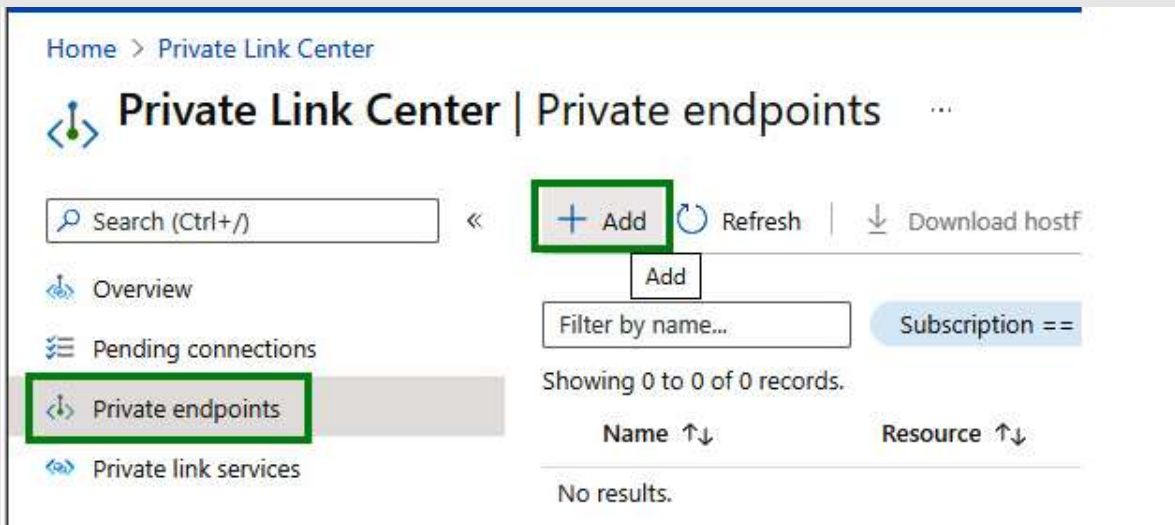


Figura 12.28 - Creación de un endpoint privado desde Private Link Center

6. En la siguiente página, puede seleccionar en qué recurso quiere que se cree el endpoint:

The screenshot shows the 'Create a private endpoint' wizard, specifically the 'Resource' step. The wizard has five steps: 'Basics', 'Resource' (current step), 'Configuration', 'Tags', and 'Review + create'. The 'Resource' step includes a 'Connection method' section with two radio buttons: 'Connect to an Azure resource in my directory.' (selected) and 'Connect to an Azure resource by resource ID or alias.' Below this are four dropdown menus: 'Subscription' (set to 'Azure subscription 1'), 'Resource type' (set to 'Microsoft.Synapse/workspaces'), 'Resource' (set to 'iacsynapsews'), and 'Target sub-resource' (set to 'Select a target sub-resource'). The 'Target sub-resource' dropdown is open, showing options: 'Sql', 'SqlOnDemand', and 'Dev'. At the bottom, there are navigation buttons: '< Previous' and 'Next : Configuration >'. A green box highlights the 'Resource' and 'Target sub-resource' dropdowns.

Figura 12.29 - Configuración del servicio en el que se creará el endpoint privado

7. En la pestaña de Configuración, podrá proporcionar los detalles de la VNet:

The screenshot shows the 'Create a private endpoint' wizard in the Azure Private Link Center. The 'Configuration' tab is active, showing the following details:

- Networking:** To deploy the private endpoint, select a virtual network subnet. [Learn more](#)
- Virtual network \***: analytics-vnet
- Subnet \***: analytics-vnet/default (10.0.0.0/24)
- Private DNS integration:** To connect privately with your private endpoint, you need a DNS record. We recommend that you integrate your private endpoint with a private DNS zone. You can also utilize your own DNS servers or create DNS records using the host files on your virtual machines. [Learn more](#)
- Integrate with private DNS zone:** ☒ Yes ☐ No
- Configuration name:** privatelink-sql-azur...
- Subscription:** Azure subscript...
- Resource group:** cloud-shell-stor...
- Private DNS zone:** (new) privatelink.sql...

Navigation buttons at the bottom: < Previous, Next : Tags >

Figura 12.30 - Configurando la VNet para el endpoint privado

8. Una vez introducidos los datos anteriores y habiendo pulsado el botón Revisar + crear en la última pantalla, se creará el endpoint privado.

A partir de ahora, el área de trabajo de Synapse puede ser accedida sólo dentro de la red de análisis que fue especificada en el ejemplo.

Puede aprender más sobre los endpoints privados aquí: <https://docs.microsoft.com/en-us/azure/private-link/create-private-endpoint-portal>.

Ahora que sabe como crear un endpoint privado, hay otra manera fácil de hacerlo usando una red virtual administrada y endpoints administrados. Cuando crea un área de trabajo de Synapse, bajo la pestaña de redes, hay una opción de red virtual gestionada, como se muestra en la siguiente captura de pantalla:

Home > Azure Synapse Analytics >

## Create Synapse workspace

\* Basics \* Security **Networking** Tags Review + create

Configure networking options for your workspace.

### Managed virtual network

Choose whether to set up a dedicated Azure Synapse-managed virtual network for your workspace. [Learn more](#)

Managed virtual network ☒ Enable ☐ Disable

Create managed private endpoint to primary storage account ☒ Yes ☐ No

Allow outbound data traffic only to approved targets ☐ Yes ☒ No

### Public network access

Choose whether to permit public network access to your workspace. You can modify the firewall rules after you enable this setting. [Learn more](#)

Public network access to workspace endpoints ☒ Enable ☐ Disable

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

Figura 12.31 - Creando una red virtual gestionada mientras se crea un área de trabajo Synapse

Cuando habilita la red virtual gestionada, Synapse se encarga de crear la VNet, crear los endpoints privados, crear las reglas de firewall adecuadas, crear las subredes adecuadas, etc. Esta es una manera muy conveniente y menos propensa a errores para crear endpoints privados. Una VNet privada gestionada y los endpoints no son diferentes a los creados manualmente. Solo que el ciclo de vida de las VNets y endpoints administrados son atendidos por el servicio anfitrión, que en nuestro caso es Synapse.

Veamos a continuación el uso de tokens de acceso en Azure Databricks.

## 12.12. Implementación de tokens de recursos en Azure Databricks

Azure Databricks proporciona tokens de acceso llamados Personal Access Tokens (PATs) que pueden ser utilizados para autenticar las APIs de Azure Databricks.

El siguiente ejemplo muestra cómo crear un nuevo Azure Databricks PAT:

1. Seleccione Configuración de usuario en la pestaña Configuración.

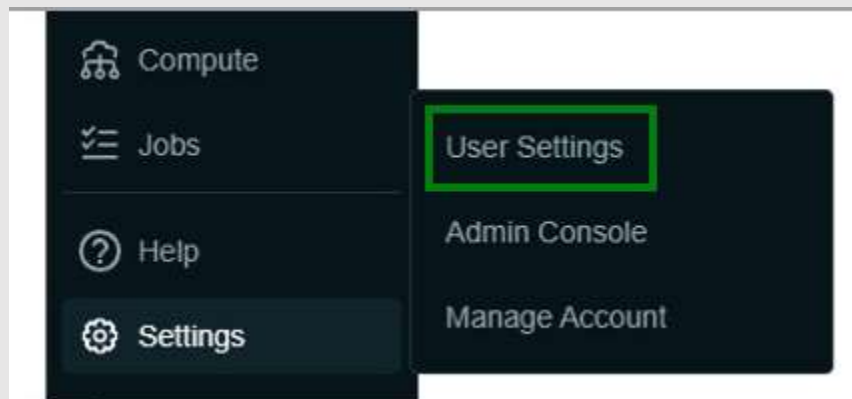


Figura 12.32 - Acceso a la configuración de usuario en Azure Databricks

2. Haz clic en el botón Generate New Token.

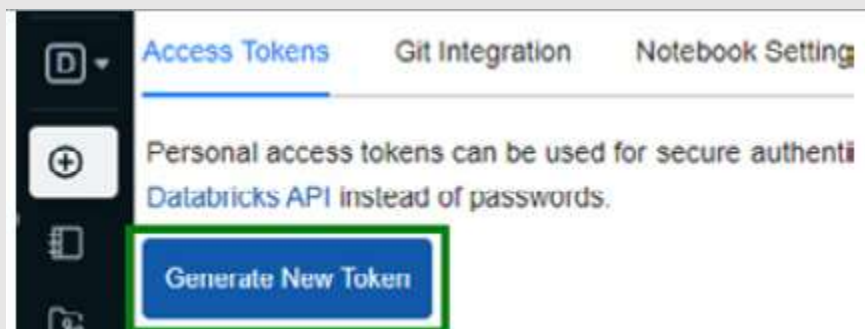


Figura 12.33 - El botón Generate New Token en Azure Databricks

3. Rellena los campos Comment y Lifetime necesarios para el token:



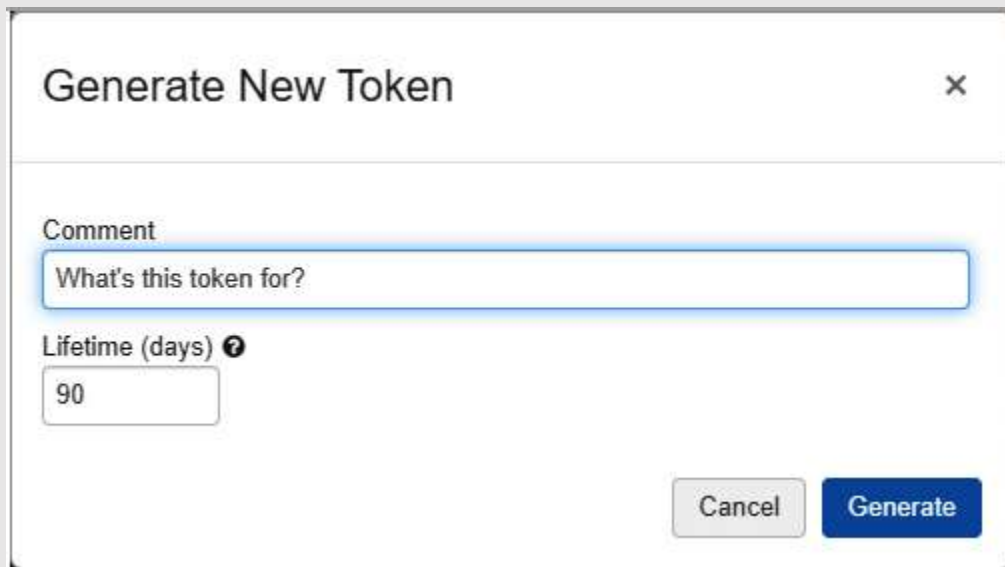


Figura 12.34 - Creación de un nuevo PAT de Azure Databricks

#### NOTA

Al hacer clic en Generar, aparecerá una pantalla con el token. Tendrás que copiarlo y almacenarlo de forma segura en ese momento. No podrás volver a copiar ese token una vez que se cierre esa pantalla. Si pierdes el token, tendrás que borrarlo y generar uno nuevo.

4. También puede crear un PAT utilizando las API, como se muestra aquí. Por ejemplo, la siguiente solicitud es para un token que será válido durante 1 día (86.400 segundos):

```
curl --netrc --request POST \  
https://<databricks-instance>/api/2.0/token/create \  
--data '{ "comment": "ADB PAT token", "lifetime_seconds": 86400 }'
```

Una vez que tenga un PAT, puede utilizarlo en las APIs, como se muestra en el siguiente bloque de código:

```
export DATABRICKS_TOKEN=<INSERT YOUR TOKEN>  
curl -X GET --header "Authorization: Bearer $DATABRICKS_TOKEN"  
https://<ADB-INSTANCE>.azuredatabricks.net/api/2.0/clusters/list
```

#### NOTA

El número máximo de PATs por área de trabajo de Azure Databricks es de 600.

Puede obtener más información sobre las PAT aquí: <https://docs.microsoft.com/en-us/azure/databricks/administration-guide/access-control/tokens>.

Al igual que los PATs de Azure Databricks, también se pueden utilizar tokens AAD normales para la autorización. Si estás interesado, puedes leer sobre ello aquí: <https://docs.microsoft.com/en-us/azure/databricks/dev-tools/api/latest/aad/service-prin-aad-token>.

Veamos a continuación cómo manejar la información sensible dentro de Spark DataFrames.



### 12.13. Carga de un DataFrame con información sensible

Anteriormente en este capítulo, aprendimos sobre técnicas como el enmascaramiento de datos y la seguridad a nivel de filas y columnas para Azure Synapse SQL. Spark, en el momento de escribir este libro, no disponía de estas técnicas para manejar información sensible. En esta sección, veremos un ejemplo de la mejor manera de emular el manejo de información sensible, como la Personally Identifiable Information (PII), utilizando el cifrado y el descifrado:

1. Vamos a crear una tabla simple que contenga información PII como números de seguridad social (SSNs) usando PySpark:

```
from pyspark.sql.types import StructType, StructField, StringType, IntegerType
cols = StructType([ \
    StructField("Name",StringType(),True), \
    StructField("SSN",StringType(),True), \
    StructField("email",StringType(),True)
])
data = [("Adam Smith","111-11-1111","james@james.com"),
        ("Brenda Harman","222-22-2222","brenda@brenda.com"),
        ("Carmen Pinto","333-33-3333","carmen@carmen.com")
]
piidf = spark.createDataFrame(data=data,schema=cols)
display(piidf)
```

La salida será algo parecido a lo siguiente:

	Name ▲	SSN ▲	email ▲
1	Adam Smith	111-11-1111	james@james.com
2	Brenda Harman	222-22-2222	brenda@brenda.com
3	Carmen Pinto	333-33-3333	carmen@carmen.com

Figura 12.35 - Tabla de ejemplo con PII

2. A continuación, vamos a importar la librería de encriptación Fernet, que proporciona la capacidad de encriptar y desencriptar texto. Puedes descargar la librería Fernet de <https://cryptography.io/en/latest/fernet/>:

```
from cryptography.fernet import Fernet
encryptionKey = Fernet.generate_key()
```

3. Define la función definida por el usuario (UDF) encrypt:

```
def encryptUdf(plaintext, KEY):
    from cryptography.fernet import Fernet
    f = Fernet(KEY)
```

```

encryptedtext = f.encrypt(bytes(plaintext, 'utf-8'))
return str(encryptedtext.decode('ascii'))
encrypt = udf(encryptUdf, StringType())

```

4. Definir el UDF decrypt:

```

def decryptUdf(encryptedtext, KEY):
    from cryptography.fernet import Fernet
    f = Fernet(KEY)
    plaintext=f.decrypt( encryptedtext.encode()).decode()
    return plaintext
decrypt = udf(decryptUdf, StringType())

```

5. Encrypt the SSN column DataFrame:

```

df = piidf.withColumn("SSN", encrypt("SSN", lit(encryptionKey)))
display(encrypteddf)

```

La salida estará ahora encriptada.

	Name	SSN
1	Adam Smith	gAAAAABhYbAciaaO-twCsrR2cRSxv8i5HSQcQl5nLDQPGXrDabn5UW5a5hGNkzooEEilPmqmrnlvxq8niDF1
2	Brenda Harman	gAAAAABhYbAdEYKEYdqSKyr3DG87EvVre2SMXK2_dfB2zZM4pt1Wlm2DKB-YI1kinKsdSVP4Fz_EshH7HdU6EHYvj5JU1OGBDA==
3	Carmen Pinto	gAAAAABhYbAcMjc-lj0rkAcMbB5t4A3dLouJxVKj4o_DbxTLnPENuaM6JsnKRMwnqKhZs3mNzdxjHgrxXDqQUfugDYdmg==

Figura 12.36 - Salida con la información PII encriptada

6. Ahora, podemos seguir adelante y guardarlo como una tabla:

```

df.write.format("delta").mode("overwrite").option("overwriteSchema",
"true").saveAsTable("PIIEncryptedTable")

```

7. Alternativamente, también puedes escribir el archivo encriptado en Parquet, como se muestra aquí:

```

encrypted.write.mode("overwrite").parquet("abfss://path/to/store")

```

8. A partir de ahora, sólo quien tenga la clave de cifrado podrá descifrar y ver la información PII. Si tienes la clave de encriptación, podrías desencriptar la columna, como se muestra aquí:

```

decrypted = encrypteddf.withColumn("SSN", decrypt("SSN",lit(encryptionKey)))
display(decrypted)

```

decrypted: pyspark.sql.dataframe.DataFrame = [Name: string, SSN: string, email: string]

	Name ▲	SSN ▲	email ▲
1	Adam Smith	111-11-1111	james@james.com
2	Brenda Harman	222-22-2222	brenda@brenda.com
3	Carmen Pinto	333-33-3333	carmen@carmen.com

Figura 12.37 - Tabla desencryptada con PII de nuevo

Espero que ahora tenga una idea de cómo realizar el cifrado y descifrado a nivel de columna utilizando DataFrames. Esta técnica funcionaría bien tanto con Synapse Spark como con Databricks Spark.

A continuación, vamos a ver cómo escribir datos encriptados en tablas y archivos.

## 12.14. Escribir datos encriptados en tablas o archivos Parquet

En realidad acabamos de ver cómo escribir datos encriptados en tablas y archivos Parquet en el ejemplo anterior. Aquí está de nuevo, escribiendo en tablas:

```
df.write.format("delta").mode("overwrite").option("overwriteSchema",  
"true").saveAsTable("PIIEncryptedTable")
```

Aquí se escribe en archivos Parquet:

```
encrypted.write.mode("overwrite").parquet("abfss://path/to/store")
```

Veamos a continuación algunas pautas para gestionar la información sensible.

## 12.15. Diseño de la privacidad de los datos y gestión de la información sensible

Cualquier organización que maneje información sensible suele estar obligada por las leyes de su país o estado y otras normas de cumplimiento a mantener los datos seguros y confidenciales. Aparte de las razones legales, mantener los datos sensibles protegidos es muy importante para la reputación de una organización y para reducir el riesgo de robo de identidad de sus clientes. Las normas de seguridad de Azure recomiendan las siguientes técnicas para mantener los datos sensibles a salvo:

- **Identificar y clasificar los datos sensibles** - El primer paso es analizar e identificar todos los datos sensibles en sus almacenes de datos. Algunos pueden ser sencillos, como las tablas SQL o los archivos estructurados, y otros pueden no serlo tanto, como los datos PII que se registran en los archivos de registro. Azure también proporciona herramientas que pueden ayudar a identificar y clasificar los datos. Por ejemplo, el portal de Synapse SQL proporciona una función para el descubrimiento y la clasificación de datos, que sugiere automáticamente las columnas sensibles. Aquí hay una captura de pantalla de muestra de la página de Descubrimiento y Clasificación de Datos:

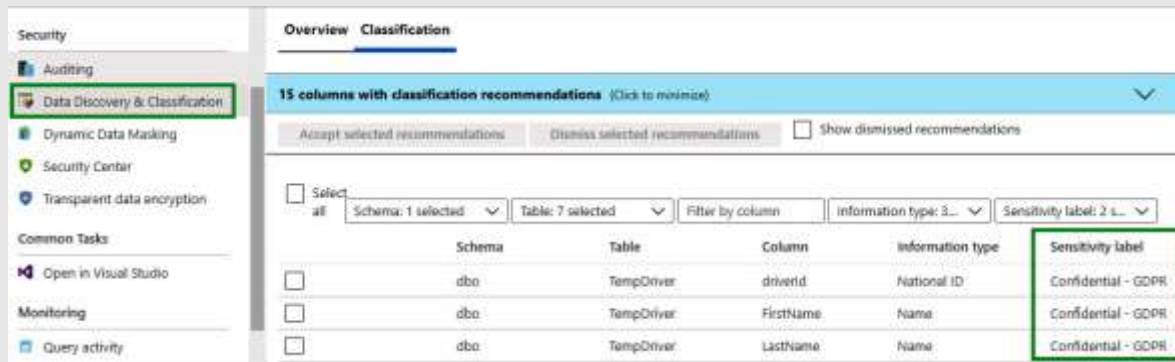


Figura 12.38 - Descubrimiento y clasificación de datos

- **Proteger los datos sensibles** - Una vez que hemos catalogado todos los datos sensibles en nuestros almacenes de datos, el siguiente paso es tomar las medidas necesarias para protegerlos. Esto incluye todas las técnicas que hemos discutido en este capítulo y en los capítulos de almacenamiento, como separar los datos sensibles en diferentes cuentas, particiones o carpetas, restringir el acceso usando RBAC y ACLs, encriptar los datos en reposo, encriptar los datos en tránsito, enmascarar los datos, y seguridad a nivel de filas y columnas.
- **Supervisión y auditoría del consumo de datos sensibles**: la mejor política de seguridad es no confiar en nadie, ni siquiera en las personas que oficialmente tienen acceso a los datos sensibles. Por lo tanto, la adición de fuertes capacidades de monitoreo y la habilitación de pistas de auditoría ayuda a rastrear activa y pasivamente cualquier acceso malicioso a los datos.

Veamos también brevemente los demás servicios disponibles en Azure para ayudar a la gestión de la seguridad y las amenazas.

## 12.15. Microsoft Defender

Para reforzar aún más la capacidad de gestión de la seguridad y las amenazas de su aplicación de Azure, Azure proporciona un servicio llamado Microsoft Defender. Microsoft Defender proporciona las herramientas y los servicios necesarios para supervisar, alertar y mitigar continuamente las amenazas a los servicios de Azure. Microsoft Defender está integrado de forma nativa en la mayoría de los servicios de Azure, por lo que se puede habilitar fácilmente sin necesidad de realizar grandes cambios en sus aplicaciones.

### Microsoft Defender para el almacenamiento

Microsoft Defender for Storage puede ayudar a identificar amenazas como el acceso anónimo, el contenido malicioso, las credenciales comprometidas, el abuso de privilegios, etc.

Puede obtener más información sobre Microsoft Defender for Storage aquí: <https://docs.microsoft.com/en-us/azure/defender-for-cloud/defender-for-storage-introduction>.

### Microsoft Defender para SQL

Microsoft Defender para SQL puede ayudar a identificar amenazas como SQL injection, los ataques de fuerza bruta y el abuso de privilegios.

Puede obtener más información sobre Microsoft Defender para SQL aquí: <https://docs.microsoft.com/en-us/azure/defender-for-cloud/defender-for-sql-introduction>.

Con esto llegamos al final de esta sección. Puede encontrar más información sobre el manejo de información sensible y las directrices de protección de datos en los siguientes enlaces:

- <https://docs.microsoft.com/en-us/security/benchmark/azure/security-control-data-protection>
- <https://docs.microsoft.com/en-us/security/benchmark/azure/security-controls-v2-data-protection>

## Resumen

Con esto, hemos llegado al final de este capítulo. Este capítulo es uno de los más largos, pero por suerte no es tan complicado. Comenzamos con el diseño de los requisitos de seguridad de nuestro ejemplo de IAC y luego usamos eso como nuestra guía para explorar los diversos temas de seguridad y cumplimiento. Aprendimos sobre el cifrado en reposo y en tránsito, la habilitación de la auditoría para Azure Data Lake Storage y Synapse SQL, la implementación del enmascaramiento de datos, las reglas RBAC y ACL, la seguridad a nivel de filas y columnas, y tuvimos una recapitulación sobre la retención y purga de datos. Después, continuamos con temas como AAD y Key Vault para gestionar claves, secretos y certificados, aprendimos sobre endpoints seguros y detalles de tokens y encriptación en Spark. Por último, terminamos con los conceptos a seguir para diseñar la privacidad de los datos. Ahora has cubierto más o menos todos los temas importantes en el manejo de la seguridad y la privacidad. Ahora deberías ser capaz de diseñar e implementar una solución completa de seguridad y privacidad sobre Azure Data Lake.

En el próximo capítulo exploraremos las técnicas de monitorización y optimización para el almacenamiento y procesamiento de datos.