

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



Capítulo 13: Supervisión del almacenamiento y procesamiento de datos	3
13.1. Requisitos técnicos.....	3
13.2. Implementing logging used by Azure Monitor.....	4
13.3. Configuración de los servicios de monitorización.....	8
13.4. Entender las opciones de custom logging.....	12
13.5. Interpretación de las métricas y logs del Azure Monitor	16
13.5.1. Interpretación de las métricas de Azure Monitor	16
13.5.2. Interpretación de los logs de Azure Monitor	16
13.6. Medición del rendimiento del movimiento de datos	19
13.7. Supervisión del rendimiento de los data pipeline.....	20
13.8. Monitorización y actualización de estadísticas sobre datos en un sistema.....	22
13.8.1. Creación de estadísticas para los Synapse dedicated pools	22
13.8.2. Actualización de las estadísticas de Synapse dedicated pools.....	23
13.8.3. Creando estadísticas para Synapse serverless pools	23
13.8.4. Actualizar las estadísticas de Synapse serverless pools	24
13.9. Medición del rendimiento de las consultas	25
13.9.1. Supervisar el rendimiento de Synapse SQL pool.....	25
Consulta de las tablas del sistema	26
Usando Query Store	26
13.9.2. Monitorización del rendimiento de las consultas de Spark	28
13.10. Interpretando un DAG de Spark.....	29
13.11. Monitorización del rendimiento del clúster.....	32
13.11.1. Supervisión del rendimiento general del clúster	32
13.11.2. Monitorización del rendimiento por nodo.....	33
13.11.3. Monitorización de YARN queue/scheduler performance	34
13.11.4. Supervisión del storage throttling.....	35
13.12. Programar y supervisar las pruebas de los pipelines	36
Resumen.....	41

Capítulo 13: Supervisión del almacenamiento y procesamiento de datos

Bienvenido al siguiente capítulo. Nos encontramos en el tramo final de nuestra formación de certificación. Esta es la última sección de la certificación: Supervisión y optimización del almacenamiento y el procesamiento de datos. Esta sección contiene dos capítulos: el actual, Supervisión del almacenamiento y el procesamiento de datos, y el siguiente, Optimización y resolución de problemas del almacenamiento y el procesamiento de datos. Como sugiere el título de este capítulo, nos centraremos en el aspecto de la monitorización del almacenamiento de datos y los pipelines. Una vez que hayas completado este capítulo, deberías ser capaz de configurar la monitorización de cualquiera de tus servicios de datos de Azure, configurar registros personalizados, procesar registros utilizando herramientas como Azure Log Analytics, y entender cómo leer los gráficos acíclicos dirigidos (DAGs) de Spark. Al igual que en los capítulos anteriores, me he tomado la libertad de reordenar la secuencia de temas para hacer más cómoda la lectura, sin demasiados cambios de contexto.

En este capítulo, cubriremos los siguientes temas:

- Implementing logging used by Azure Monitor
- Configuración de los servicios de monitorización
- Understanding custom logging options
- Interpretación de las métricas y registros de Azure Monitor
- Medición del rendimiento del movimiento de datos
- Monitorización del rendimiento de los data pipelines
- Monitorización y actualización de estadísticas sobre datos en un sistema
- Medición del rendimiento de las consultas
- Interpretación de un DAG de Spark
- Monitorización del rendimiento del cluster
- Programación y supervisión de las pruebas del pipeline

13.1. Requisitos técnicos

Para este capítulo, necesitarás lo siguiente:

- Una cuenta de Azure (gratuita o de pago)
- Un área de trabajo de Azure Data Factory (ADF) activa

Comencemos.

13.2. Implementing logging used by Azure Monitor

Azure Monitor es el servicio que utilizamos para supervisar la infraestructura, los servicios y las aplicaciones. Azure Monitor registra dos tipos de datos: métricas y logs. Las métricas son valores numéricos que describen una entidad o un aspecto de un sistema en diferentes instancias de tiempo, por ejemplo, el número de gigabytes (GB) de datos almacenados en una cuenta de almacenamiento en cualquier momento, el número actual de pipelines activos en ADF, etc. Las métricas se almacenan en bases de datos de series temporales y pueden agregarse fácilmente para fines de alerta, informes y auditoría.

Los logs, por otro lado, suelen ser detalles de texto de lo que está ocurriendo en el sistema. A diferencia de las métricas, que se registran a intervalos regulares, los logs suelen estar basados en eventos. Por ejemplo, un usuario que inicie sesión en un sistema, una aplicación web que reciba una solicitud REpresentational State Transfer (REST) y la activación de un pipeline en ADF podrían generar logs.

Dado que Azure Monitor es un servicio independiente, puede agregar registros de múltiples servicios diferentes y múltiples instancias del mismo servicio en Azure para dar una perspectiva global de lo que está sucediendo con todos nuestros servicios de Azure.

Aquí hay un diagrama de arquitectura reproducido de la documentación de Azure que muestra los componentes principales de Azure Monitor y las fuentes y destinos de los datos de métricas y logs:

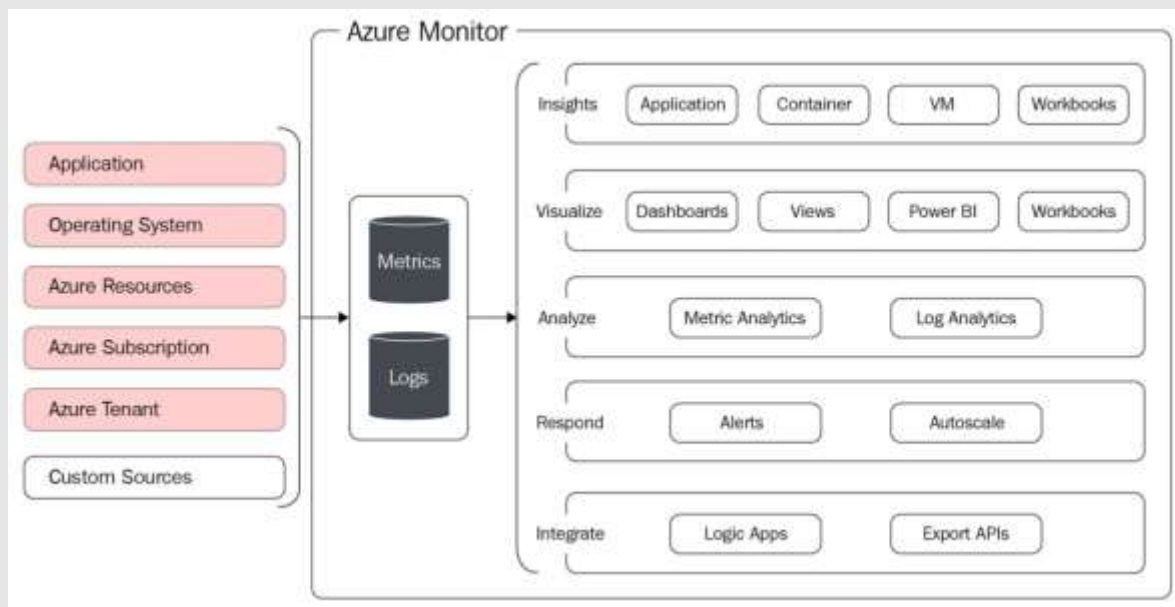


Figura 13.1 - Arquitectura de Azure Monitor

Veamos un ejemplo sencillo de cómo añadir Activity logs en Azure Monitor. Para empezar, necesitamos entender un sub-servicio de Azure Monitor llamado Log Analytics workspaces. Un workspace de Log Analytics es un componente de Azure Monitor que se especializa en procesar y explorar los mensajes log. Puedes acceder a Log Analytics desde la pestaña Logs de cualquier servicio de Azure o desde la pestaña Logs del menú de Azure Monitor. Log Analytics soporta un potente

lenguaje de consulta llamado Kusto Query Language (KQL) que se puede utilizar para realizar análisis sobre los datos de registro. Veremos algunos ejemplos de KQL en este capítulo.

NOTA

Log Analytics es una de las opciones para almacenar logs, y puede incurrir en cargos adicionales basados en el coste de almacenamiento y el número de consultas de Kusto ejecutadas. También puedes optar por almacenar los logs en las opciones habituales de almacenamiento de Azure, como Blob y Data Lake, o utilizar servicios como Event Hubs para realizar el procesamiento de los logs en tiempo real.

Veamos los pasos necesarios para registrar datos en un área de trabajo de Log Analytics.

1. En primer lugar, tenemos que crear una nueva área de trabajo de Log Analytics. Simplemente busque Log Analytics workspaces en el portal de Azure y haga clic en el resultado. Haz clic en el botón + Crear en la página de inicio de Log Analytics. Debería ver una pantalla como la que se muestra a continuación. Rellene los detalles, como los campos Grupo de recursos, Nombre y Región para la instancia de Log Analytics y, a continuación, haga clic en Revisar + Crear para crear una nueva área de trabajo de Log Analytics:

Home > Log Analytics workspaces >

Create Log Analytics workspace

Basics Tags Review + Create

Information A Log Analytics workspace is the basic management unit of Azure Monitor Logs. There are specific considerations you should take when creating a new Log Analytics workspace. [Learn more](#)

With Azure Monitor Logs you can easily store, retain, and query data collected from your monitored resources in Azure and other environments for valuable insights. A Log Analytics workspace is the logical storage unit where your log data is collected and stored.

Project details

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

Subscription * ⓘ Azure subscription 1

Resource group * ⓘ [Create new](#)

Instance details

Name * ⓘ

Region * ⓘ East US

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

Figura 13.2 - Creación de un área de trabajo de Log Analytics

2. A continuación, seleccione el servicio Azure Monitor en el portal de Azure y haga clic en la pestaña Activity log, como se ilustra en la siguiente captura de pantalla:

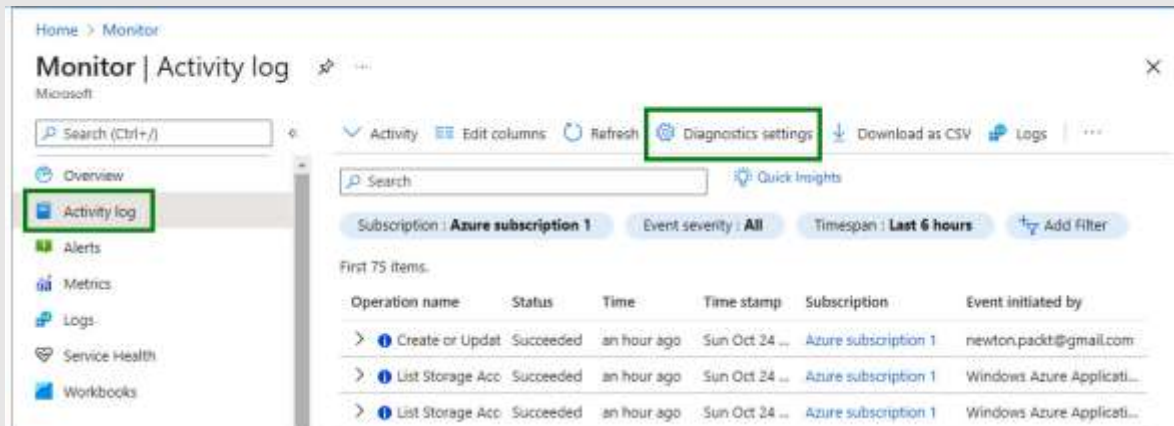


Figura 13.3 - Pantalla de Activity log de Azure Monitor

3. Haga clic en el botón Diagnostics settings y seleccione todos los logs que desee enviar al workspace de Log Analytics (el workspace que creó en el Paso 1), como se ilustra en la siguiente captura de pantalla:

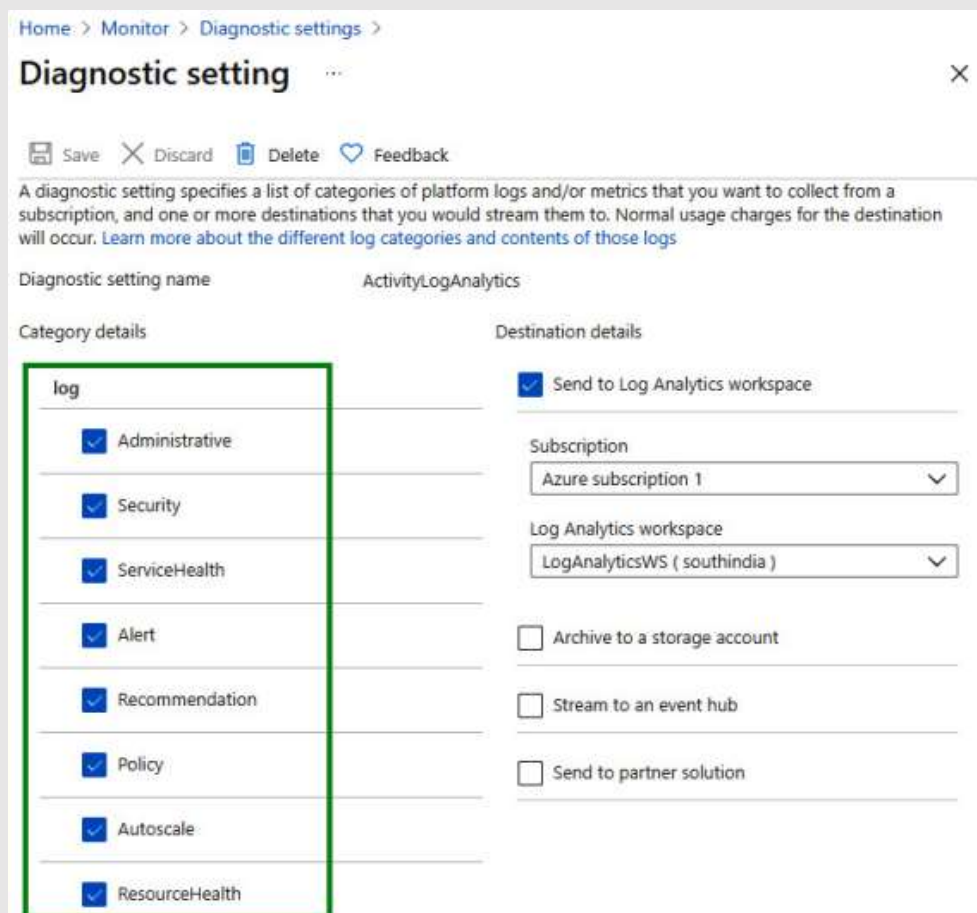


Figura 13.4 - Configuración de la pantalla de diagnóstico

Después de unos minutos, verá una tabla AzureActivity creada en el área de trabajo de Log Analytics, como se muestra en la siguiente captura de pantalla:

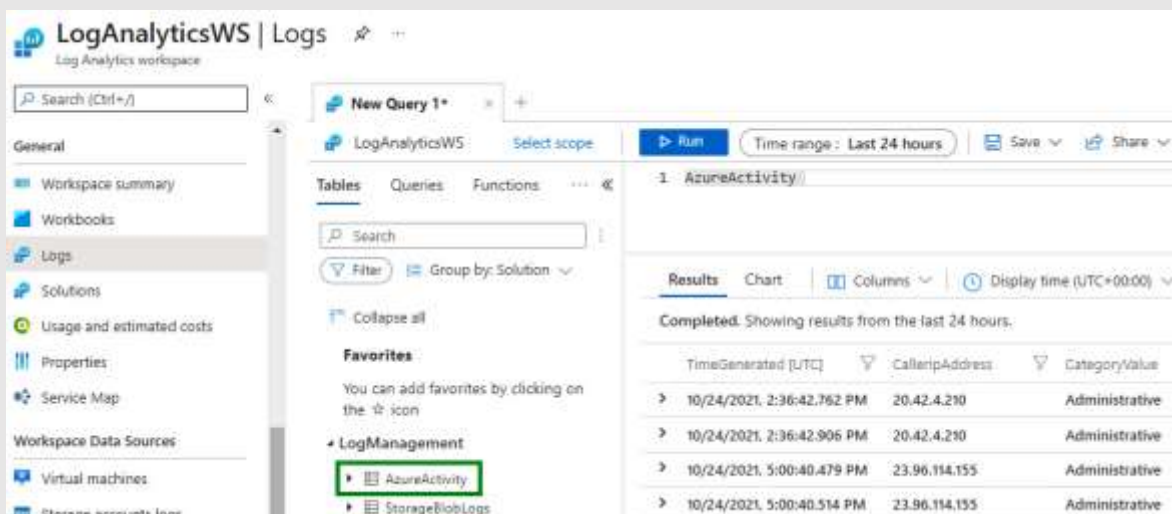


Figura 13.5 - Tabla AzureActivity en el área de trabajo de Log Analytics

A partir de ahora, todos los registros de actividad comenzarán a poblarse en esta tabla AzureActivity. Puede consultar la tabla utilizando KQL para obtener información sobre el registro.

Puedes aprender más sobre el envío de logs a un workspace de Log Analytics aquí: <https://docs.microsoft.com/en-us/azure/azure-monitor/essentials/quick-collect-activity-log-portal>

A continuación, veamos cómo configurar la monitorización de los recursos de Azure.

13.3. Configuración de los servicios de monitorización

Azure Monitor se habilita en cuanto creamos un recurso Azure. Por defecto, las métricas y los logs básicos se registran sin requerir ningún cambio de configuración por parte del usuario, pero podemos realizar configuraciones adicionales como el envío de los logs a Log Analytics, como vimos en la sección anterior. Podemos configurar la monitorización a varios niveles, como se indica aquí:

- **Monitorización de aplicaciones**-Métricas y logs sobre las aplicaciones que se han escrito sobre los servicios de Azure.
- **Monitorización del sistema operativo (SO)**: métricas y logs a nivel del SO, como el uso de la CPU, el uso de la memoria, el uso del disco, etc.
- **Supervisión de recursos de Azure**: métricas y logs de servicios de Azure como Azure Storage, Synapse Analytics, Event Hubs, etc.
- **Supervisión a nivel de suscripción**: métricas y logs de las suscripciones de Azure, como el número de personas que utilizan una cuenta concreta, el uso de la cuenta, etc.
- **Tenant-level monitoring**: métricas y logs de los servicios a nivel de tenant, como Azure Active Directory (Azure AD).

En el caso de la supervisión a nivel de recursos, de suscripciones y de tenant, la mayoría de los datos ya están generados y sólo tenemos que activar la configuración de diagnóstico, como se muestra en la Figura 13.4, para trasladar esos datos a cualquiera de los tres destinos de registro siguientes:

- Log Analytics workspace
- Event Hubs
- Storage account

Pero para la monitorización de aplicaciones y sistemas operativos invitados, tendremos que instalar el agente Azure Monitor (AMA) o el agente Log Analytics para empezar a recoger métricas y logs. El agente AMA o Azure Monitor se puede instalar de múltiples maneras. A continuación se presenta una forma sencilla de instalarlo a través de extensiones de máquinas virtuales (VM) en un servidor Linux. Puede iniciar sesión en la interfaz de línea de comandos (CLI) de Azure y ejecutar el siguiente comando:

```
az vm extension set \
  --resource-group <YOUR_RESOURCE_GROUP> \
  --vm-name <VM_NAME> \
  --name OmsAgentForLinux \
  --publisher Microsoft.EnterpriseCloud.Monitoring \
  --protected-settings '{"workspaceKey":"<YOUR_WORKSPACE_KEY>"}' \
  --settings '{"workspaceId":"<YOUR_WORKSPACE_ID>"}'
```


Para conocer otras formas de instalar clientes de monitorización, consulte el siguiente enlace: <https://docs.microsoft.com/en-us/azure/azure-monitor/agents/log-analytics-agent#installation-options>.

Centrémonos ahora en las opciones disponibles para la monitorización de los recursos de Azure, ya que esto es importante desde el punto de vista de la certificación. Tomemos el ejemplo del almacenamiento de Azure.

Las métricas como Ingress, Egress, Blob Capacity, etc. que indican el estado de las operaciones dentro de Azure Storage están directamente disponibles en la pestaña Metrics de Azure Storage. Puedes filtrar, crear nuevos gráficos y crear nuevas alertas utilizando todas las métricas disponibles en esta página, como se muestra en la siguiente captura de pantalla:

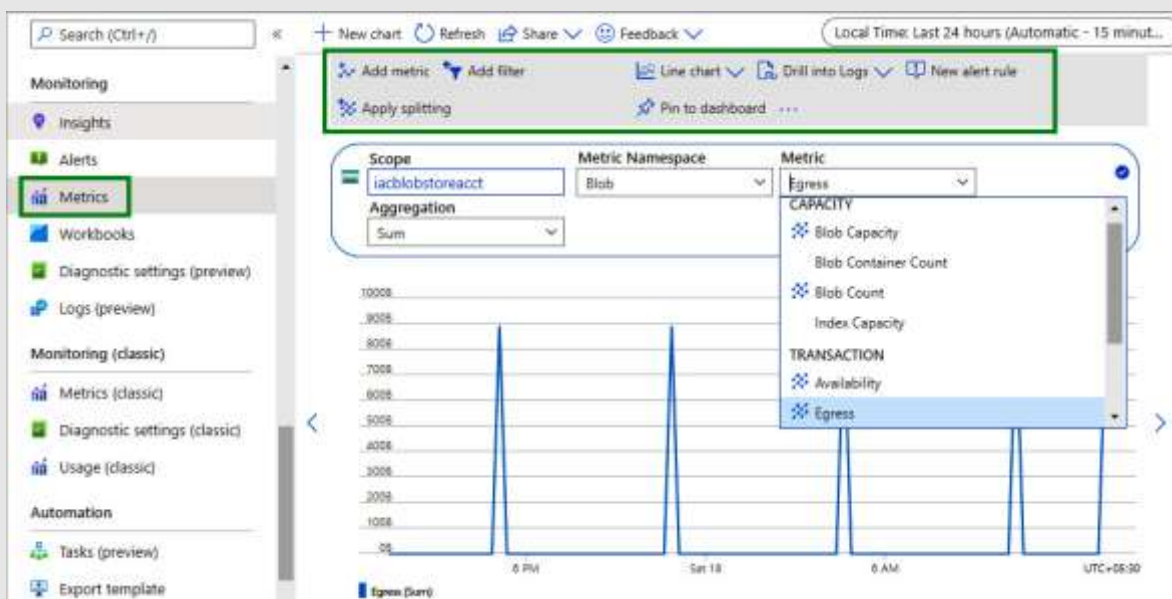


Figura 13.6 - Configuración de métricas para Azure Storage

Pero si lo que necesitamos es conocer el estado del propio servicio de almacenamiento, como por ejemplo la disponibilidad de la cuenta de Blob storage, entonces tendremos que acudir a Azure Monitor. Desde Azure Monitor, selecciona Cuentas de almacenamiento. Deberías poder ver detalles como las Transacciones generales, la Línea de tiempo de las Transacciones, etc., como se muestra en la siguiente captura de pantalla:

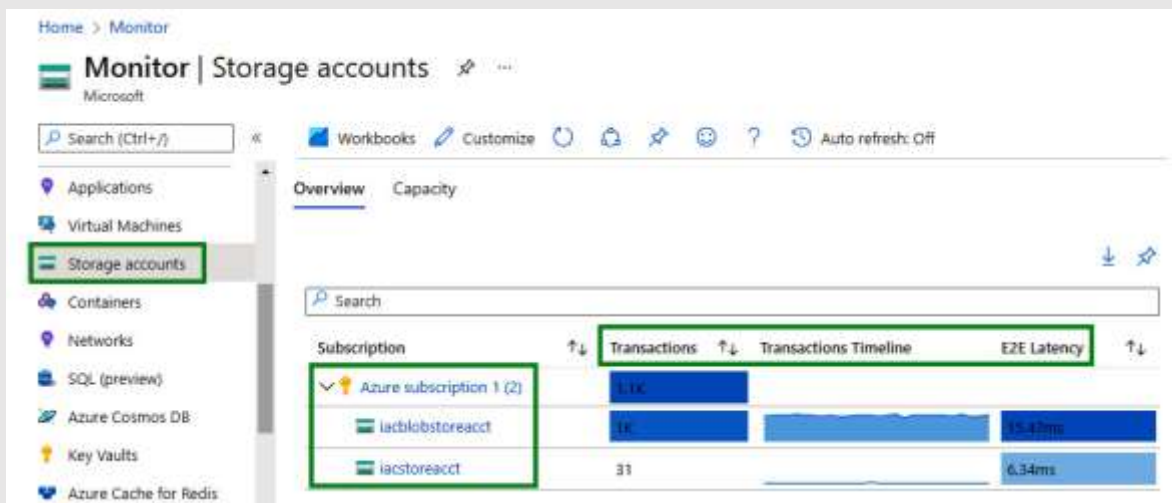


Figura 13.7 - Métricas de almacenamiento desde Azure Monitor

Si haces clic en cualquiera de los enlaces de las cuentas de almacenamiento y seleccionas la pestaña Insights, se muestran detalles de Fallos, Rendimiento, Disponibilidad y Capacidad, como se muestra en la siguiente captura de pantalla:

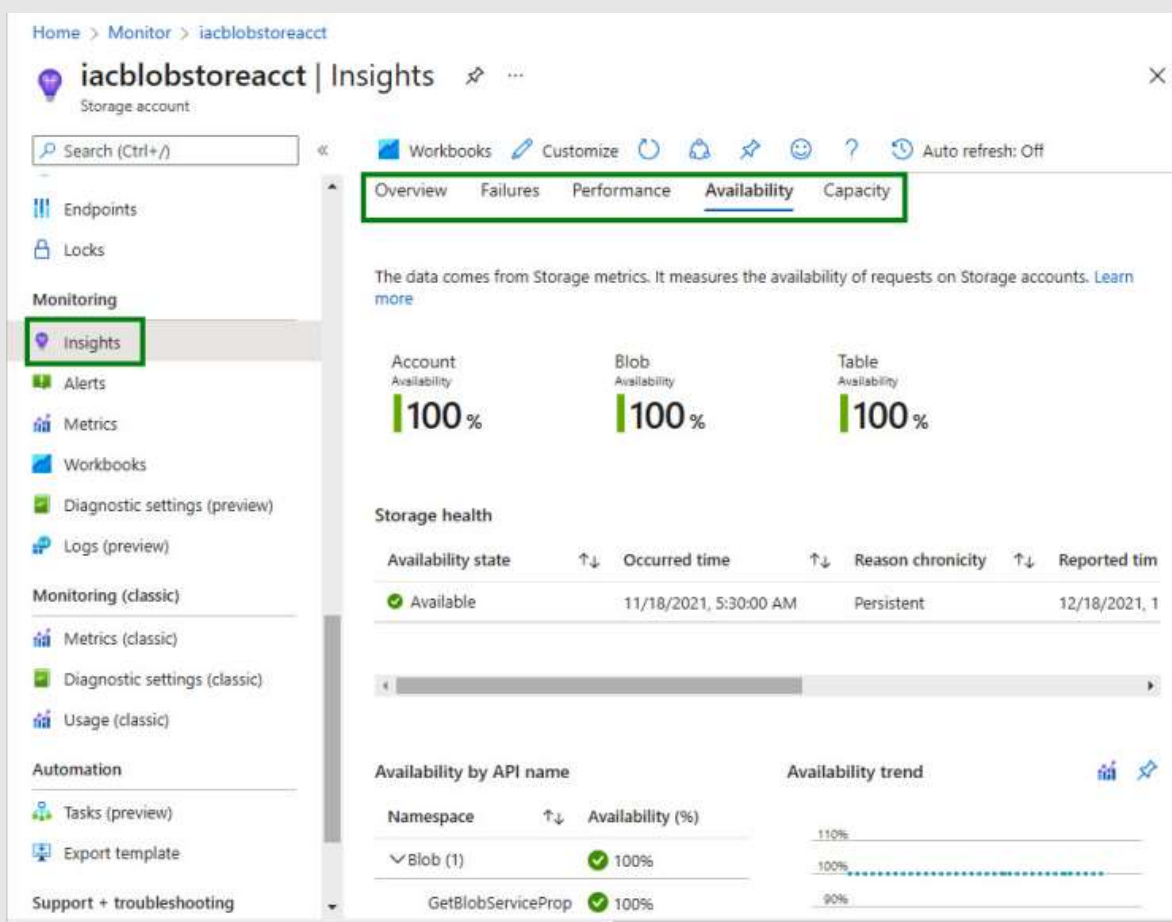


Figura 13.8 - Detalles de la disponibilidad del almacenamiento desde Azure Monitor

Así es como puedes utilizar las características de monitorización disponibles dentro del servicio y con Azure Monitor para obtener una imagen completa de la salud de tu servicio.

A continuación, conozcamos las opciones de registro personalizado disponibles en Azure Monitor.

13.4. Entender las opciones de custom logging

La opción de custom logs en Azure Monitor ayuda a recolectar logs basados en texto que no forman parte de los logs estándar recolectados por Azure Monitor, como los logs del sistema, los logs de eventos en Windows y otros similares en Linux. Para configurar los custom logs, la host machine debe tener el agente de Log Analytics o el más reciente AMA instalado en ella. Acabamos de ver cómo instalar los agentes en la sección anterior.

Una vez que nos hemos asegurado de que los agentes están instalados, es un proceso muy fácil configurar los custom logs. Estos son los pasos:

1. En el área de trabajo de Log Analytics, seleccione la sección Custom logs y haga clic en la opción + Add custom log, como se ilustra en la siguiente captura de pantalla:

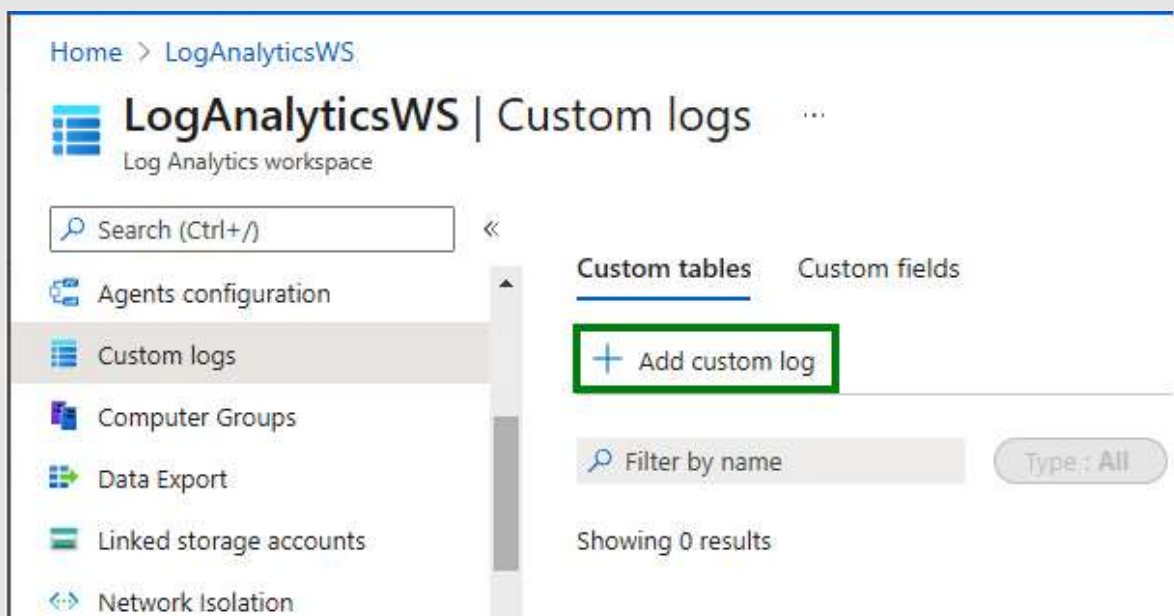


Figura 13.9 - Configuración de un nuevo custom log mediante Log Analytics

2. En el asistente que sigue, cargue un archivo de log de muestra para que la herramienta pueda analizarlo y entender el formato del log. Este es un ejemplo de un archivo de log de muestra:

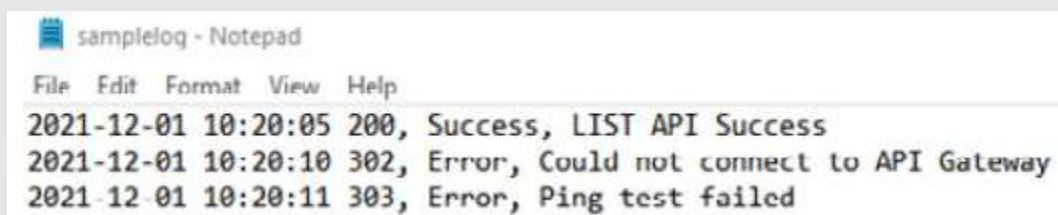


Figura 13.10 - Archivo de log de muestra

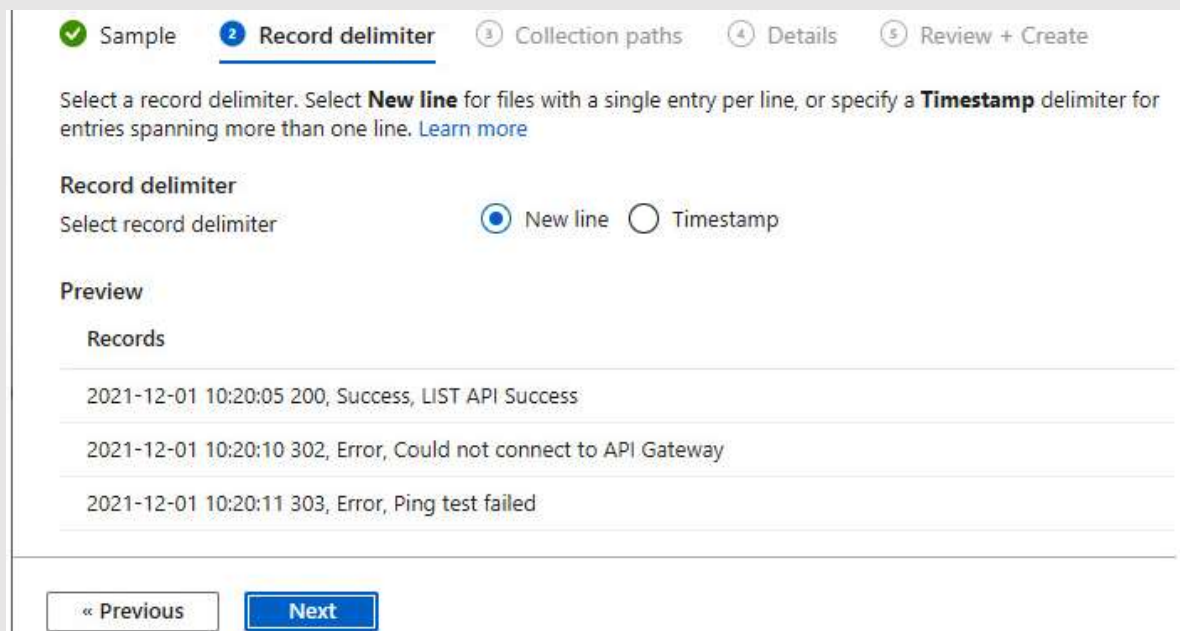
3. Busque la ubicación del archivo de muestra y cárguelo, como se muestra en la siguiente captura de pantalla:



The screenshot shows the 'Sample' step of a wizard. At the top, there are five steps: 1. Sample (active), 2. Record delimiter, 3. Collection paths, 4. Details, and 5. Review + Create. Below the steps, there is a text input field labeled 'Sample log' with the value 'samplelog.txt' and a file selection icon. Below the input field, there are two buttons: '« Previous' and 'Next'.

Figura 13.11 - Carga de un archivo de log de muestra en el asistente de logs personalizados

4. Haga clic en Siguiente, y en la siguiente pantalla, verá sus líneas de log de muestra, como se muestra en la siguiente captura de pantalla:



The screenshot shows the 'Record delimiter' step of a wizard. At the top, there are five steps: 1. Sample, 2. Record delimiter (active), 3. Collection paths, 4. Details, and 5. Review + Create. Below the steps, there is a text input field labeled 'Record delimiter' with the value 'New line' and a radio button. Below the input field, there are two radio buttons: 'New line' (selected) and 'Timestamp'. Below the radio buttons, there is a section titled 'Preview' with a sub-header 'Records'. Below the sub-header, there are three lines of log data: '2021-12-01 10:20:05 200, Success, LIST API Success', '2021-12-01 10:20:10 302, Error, Could not connect to API Gateway', and '2021-12-01 10:20:11 303, Error, Ping test failed'. Below the preview, there are two buttons: '« Previous' and 'Next'.

Figura 13.12 - Líneas de log de muestra mostradas en el asistente

5. En la siguiente pestaña, introduzca el tipo de sistema operativo y la ruta de los archivos de log, como se muestra en la siguiente captura de pantalla:

✓ Sample
✓ Record delimiter
3 Collection paths
4 Details
5 Review + Create

Define one or more paths on the agent where it can locate the custom log. [Learn more](#)

Collection paths

Type	Path
Linux	/var/log/app/app.log ✓
Select type	

« Previous
Next

Figura 13.13 - Configuración de las rutas de logs

- En la pestaña Detalles, introduzca un valor de Nombre de Custom log y una descripción para su custom log, como se muestra en la siguiente captura de pantalla, y haga clic en Siguiente:

Create a custom log

✓ Sample
✓ Record delimiter
✓ Collection paths
4 Details
5 Review + Create

Add a name and description to the custom log.

This name will be used for the log type, and will always end with _CL to distinguish it as a custom log. [Learn more](#)

Details

Custom log name * ✓
_CL

Description

« Previous
Next

Figura 13.14 - Añadiendo un nombre y una descripción a su Custom log

- Por último, en la pestaña Revisar + Crear, revise las entradas y haga clic en el botón Crear para crear un Custom log. Así de fácil.

Tenga en cuenta que los Custom logs tienen las siguientes restricciones:

- Sólo se permite el siguiente conjunto predefinido de formatos de timestamp:

```
YYYY-MM-DD HH:MM:SS  
M/D/YYYY HH:MM:SS AM/PM  
Mon DD, YYYY HH:MM:SS  
yyMMdd HH:mm:ss  
ddMMyy HH:mm:ss  
MMM d hh:mm:ss  
dd/MMM/yyyy:HH:mm:ss zzz  
yyyy-MM-ddTHH:mm:ssK
```

- No se permiten rotaciones de registros.
- Sólo se proporciona soporte para el Código Estándar Americano para el Intercambio de Información (ASCII) o el Formato de Transformación Unicode 8 (UTF-8).
- La conversión de zonas horarias no está disponible para Linux.

Puede obtener más información sobre los Custom logs aquí: <https://docs.microsoft.com/en-us/azure/azure-monitor/agents/data-sources-custom-logs>.

A continuación, aprendamos a interpretar las métricas y los logs de Azure Monitor.

13.5. Interpretación de las métricas y logs del Azure Monitor

Como hemos visto en la introducción a Azure Monitoring, las métricas y los logs constituyen las dos fuentes principales de datos para la monitorización. Vamos a explorar cómo ver, interpretar y experimentar con estos dos tipos de datos de monitorización.

13.5.1. Interpretación de las métricas de Azure Monitor

Los datos de las métricas recopiladas de los recursos de Azure suelen mostrarse en la página de resumen del propio recurso, y hay más detalles disponibles en la pestaña Métricas. A continuación, volvemos a ver un ejemplo de cómo se ve para una cuenta de almacenamiento:

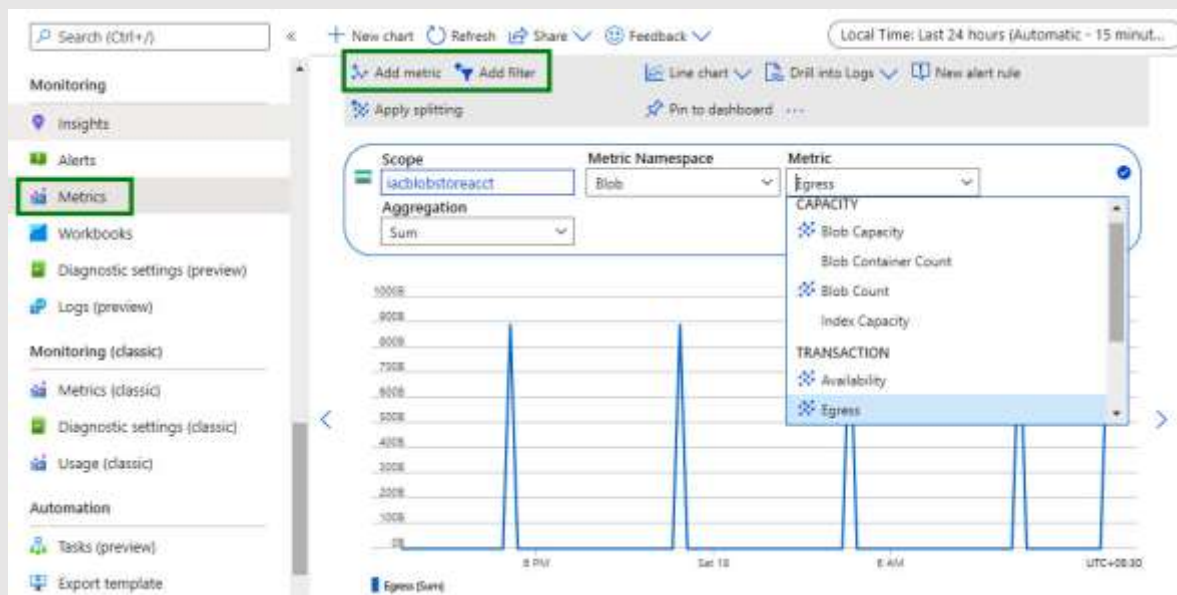


Figura 13.15 - Datos de las métricas de una cuenta de almacenamiento

Para cada una de las métricas, se puede agregar en base a Suma, Promedio, Mínimo y Máximo. La herramienta también ofrece la flexibilidad de superponer métricas adicionales mediante la opción Añadir métrica, filtrar datos no deseados mediante la opción Añadir filtro, etc. Puedes acceder a los datos de hasta 30 días utilizando este explorador de métricas.

Veamos a continuación cómo interpretar los logs.

13.5.2. Interpretación de los logs de Azure Monitor

Ya hemos aprendido a enviar los logs a Log Analytics. Ahora vamos a explorar cómo leer y experimentar con los logs en Log Analytics. Una vez que comience a enviar los logs a Log Analytics, podrá ver las tablas del recurso de Azure que haya configurado. En la siguiente captura de pantalla, verás las tablas AzureActivity y StorageBlobLogs. Estas tablas se llenarán de logs continuamente:

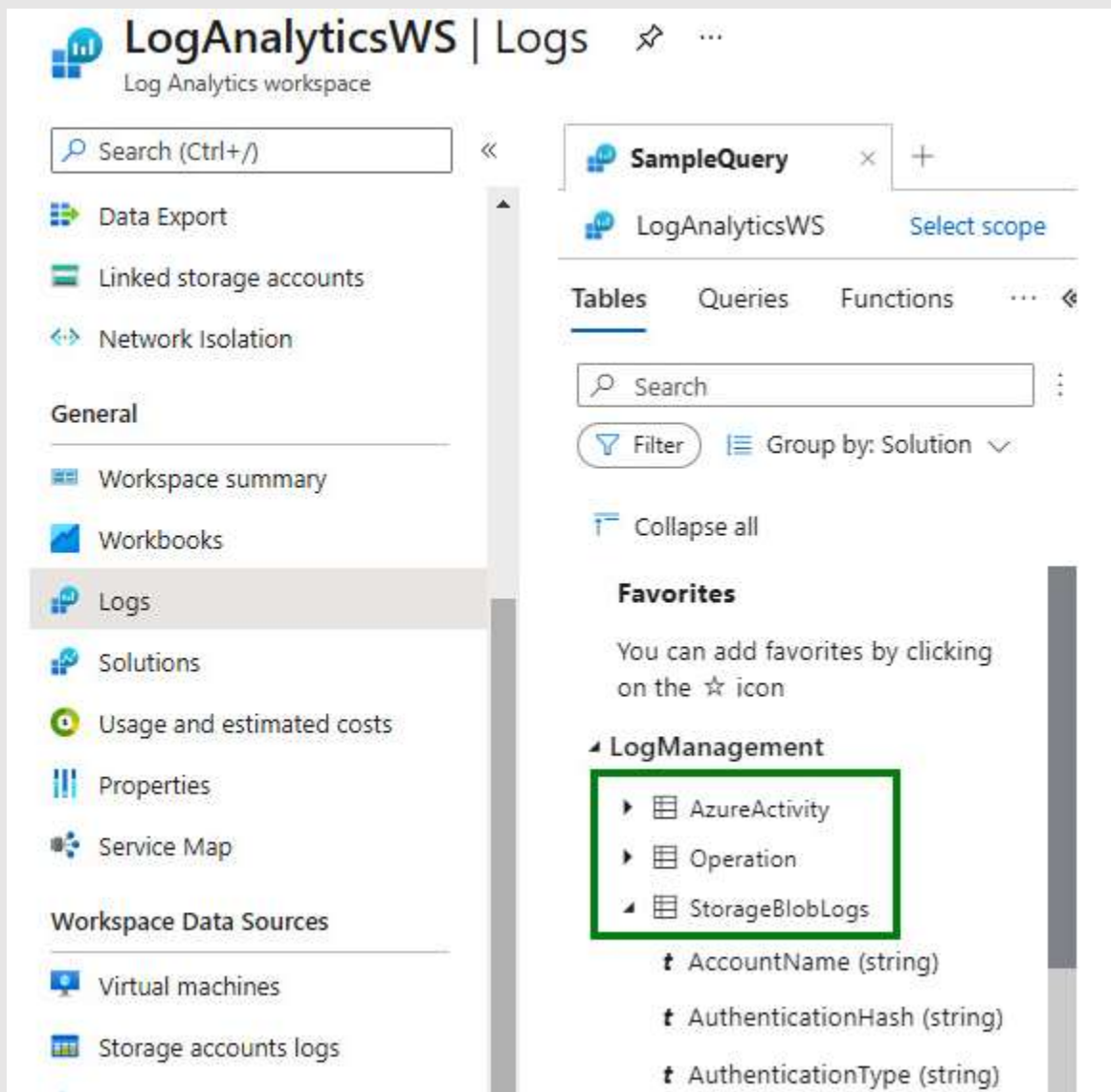


Figura 13.16 - Tablas de Log Analytics

Una vez que tenemos los datos de log en las tablas, podemos utilizar el lenguaje de consulta KQL proporcionado por Azure Monitor para consultar las tablas, tal y como hacemos en Structured Query Language (SQL). Las consultas KQL se pueden escribir directamente en el bloque de consulta, como se muestra en la siguiente captura de pantalla:

The screenshot shows the Azure Data Explorer interface. On the left, the 'LogAnalyticsWS' workspace is selected, and the 'StorageBlobLogs' table is chosen under the 'LogManagement' category. The main pane displays a Kusto query:

```
1 StorageBlobLogs |
2 take 100 | where ServerLatencyMs > 20 |
3 sort by DurationMs |
4 project TimeGenerated, OperationName, DurationMs
```

The query is executed with a time range of 'Last 7 days'. The results are displayed in a table format, showing the top 100 entries where the server latency is greater than 20 ms, sorted by duration in descending order. The table has three columns: TimeGenerated [UTC], OperationName, and DurationMs.

TimeGenerated [UTC]	OperationName	DurationMs
12/22/2021, 11:09:49.078 PM	ListBlobs	1,332
12/22/2021, 3:14:00.013 AM	ListBlobs	857
12/22/2021, 3:13:59.752 AM	GetBlobServiceProper...	716
12/22/2021, 3:14:02.986 AM	ListBlobs	334
12/22/2021, 11:09:47.864 PM	ListBlobs	305
12/22/2021, 3:13:59.680 AM	ListBlobs	248
12/22/2021, 3:14:06.182 AM	ListBlobs	235
12/22/2021, 3:14:29.327 AM	ListBlobs	233
12/22/2021, 3:15:08.102 AM	GetBlob	61

Figura 13.17 - Ejemplo de consulta Kusto

Una consulta Kusto consiste en una secuencia de sentencias que pueden estar delimitadas por punto y coma. Se comienza con una fuente de datos como una tabla y luego se utilizan pipes (|) para transformar los datos de un paso a otro. Por ejemplo, considere la captura de pantalla anterior. Comenzamos con la tabla StorageBlobLogs -sólo con especificar el nombre de la tabla se mostrará el contenido de la misma. A continuación, canalizamos esos datos para tomar 100. Ahora, las 100 entradas se canalizan a través de la siguiente sentencia, donde ServerLatencyMs > 20, que filtra las filas cuya latencia es superior a 20 milisegundos (ms). A continuación, la ordenamos de forma descendente e imprimimos una tabla que contiene tres columnas: TiempoGenerado, NombreDeLaOperación, y DuraciónMs, para saber cuánto duró la operación.

Como ves, Kusto -o KQL- es un lenguaje muy intuitivo que es una mezcla de SQL y Linux pipes (|). Una vez que lo domine, podrá generar información muy poderosa utilizando los datos de log en Log Analytics.

Puedes aprender más sobre KQL aquí: <https://docs.microsoft.com/en-us/azure/data-explorer/kusto/query/>.

Ahora que hemos aprendido a configurar e interpretar las métricas y los logs de los servicios de Azure, vamos a explorar cómo medir el rendimiento de los movimientos de datos dentro de dichos servicios. Aprenderemos a comprobar el rendimiento de los movimientos de datos mientras usamos ADF.

13.6. Medición del rendimiento del movimiento de datos

ADF proporciona un rico conjunto de métricas de rendimiento en su pestaña de Monitorización. En el siguiente ejemplo, tenemos una actividad de ejemplo de Copiar Datos como parte de un pipeline llamado FetchDataFromBlob, que copia datos del almacenamiento Blob en Azure Data Lake Storage Gen2 (ADLS Gen2). Si haces clic en la pestaña de ejecuciones de pipelines bajo la pestaña de monitorización, verás los detalles de cada uno de los pipelines. Si hace clic en cualquiera de los pasos, verá los detalles de diagnóstico:

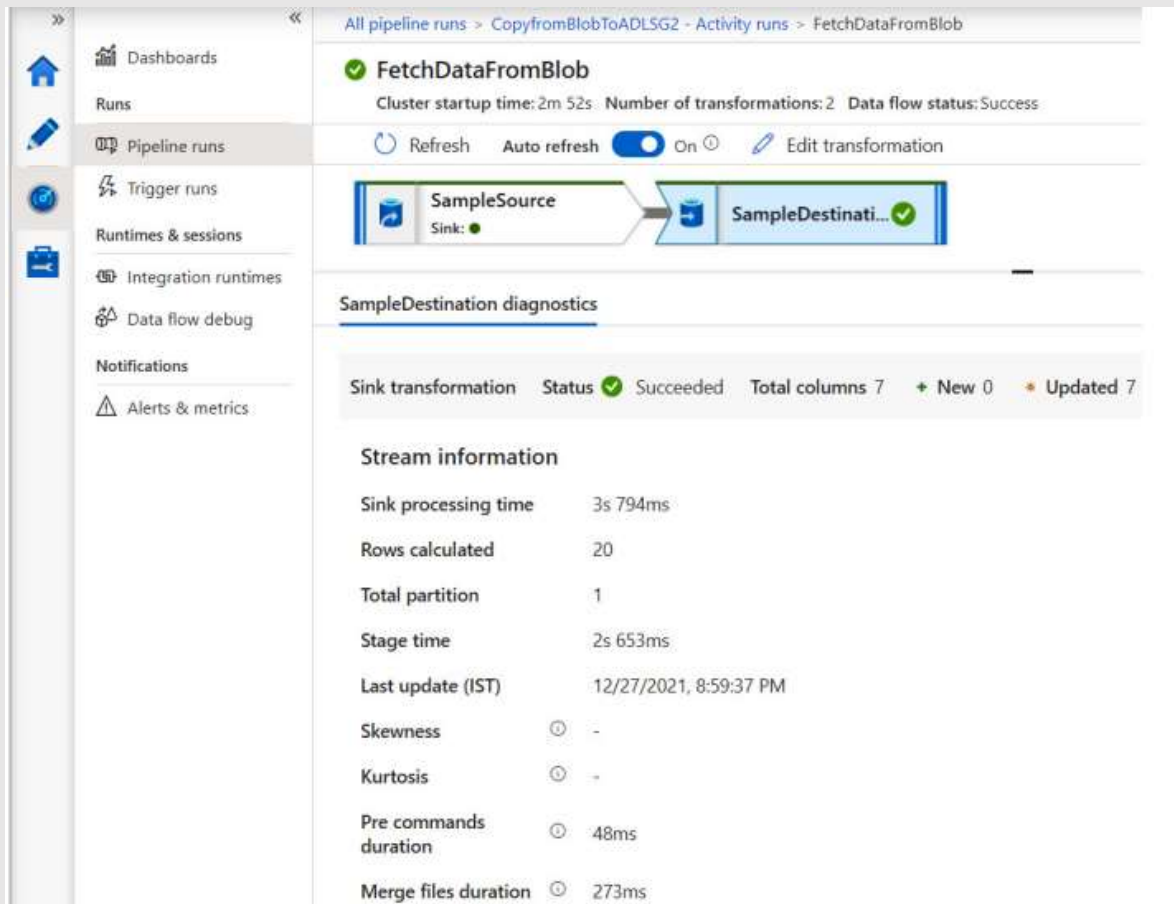


Figura 13.18 - Detalles del rendimiento del movimiento de datos

Así es como se puede monitorizar el rendimiento del movimiento de datos. Puede obtener más información sobre la monitorización de los datos copiados aquí: <https://docs.microsoft.com/en-us/azure/data-factory/copy-activity-monitoring>.

Veamos a continuación cómo supervisar el rendimiento general del pipeline.

13.7. Supervisión del rendimiento de los data pipeline

Al igual que las métricas de movimiento de datos que vimos en la sección anterior, ADF también proporciona métricas para los pipelines en general. En la página de ejecuciones de pipelines en la pestaña de Monitorización, si pasa el ratón por encima de las ejecuciones de pipelines, aparece un pequeño icono de Consumo, como se muestra en la siguiente captura de pantalla:

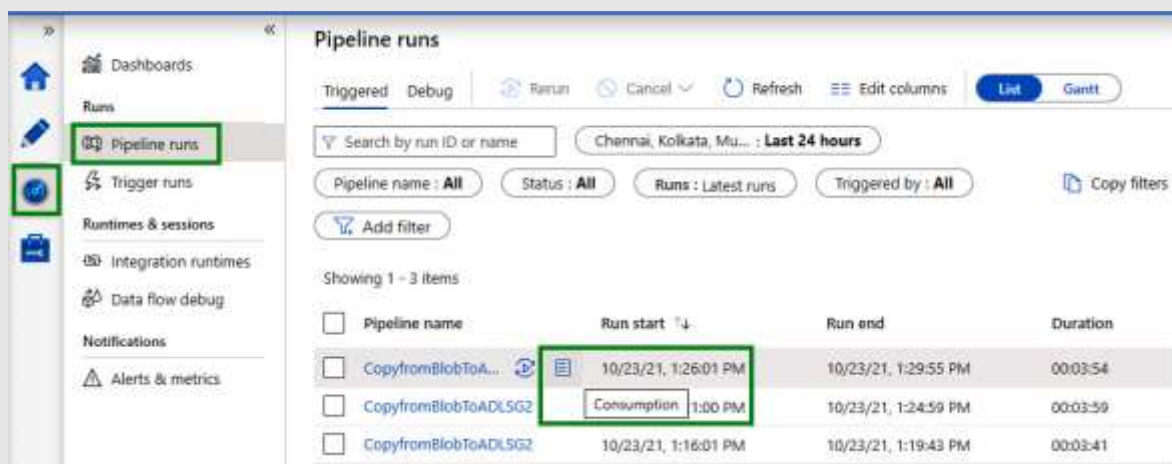


Figura 13.19 - Icono de consumo en la pantalla de ejecuciones de pipeline

Si hace clic en ese icono, el ADF muestra los detalles del consumo del pipeline. A continuación se muestra una pantalla de ejemplo:

Pipeline run consumption			✕
	Quantity	Unit	
Pipeline orchestration			
Activity Runs	1	Activity runs	
Data flow			
General purpose	0.5001	vCore-hour	
Compute optimized	0.0000	vCore-hour	
Memory optimized	0.0000	vCore-hour	
Learn more ⓘ			
Pricing calculator ⓘ			

Figura 13.20 - Pantalla de detalles de consumo de recursos del pipeline

También puede obtener métricas adicionales sobre cada una de las ejecuciones desde la sección del diagrama de Gantt. Puede cambiar la vista de Lista a Gantt, como se muestra en la siguiente captura de pantalla:

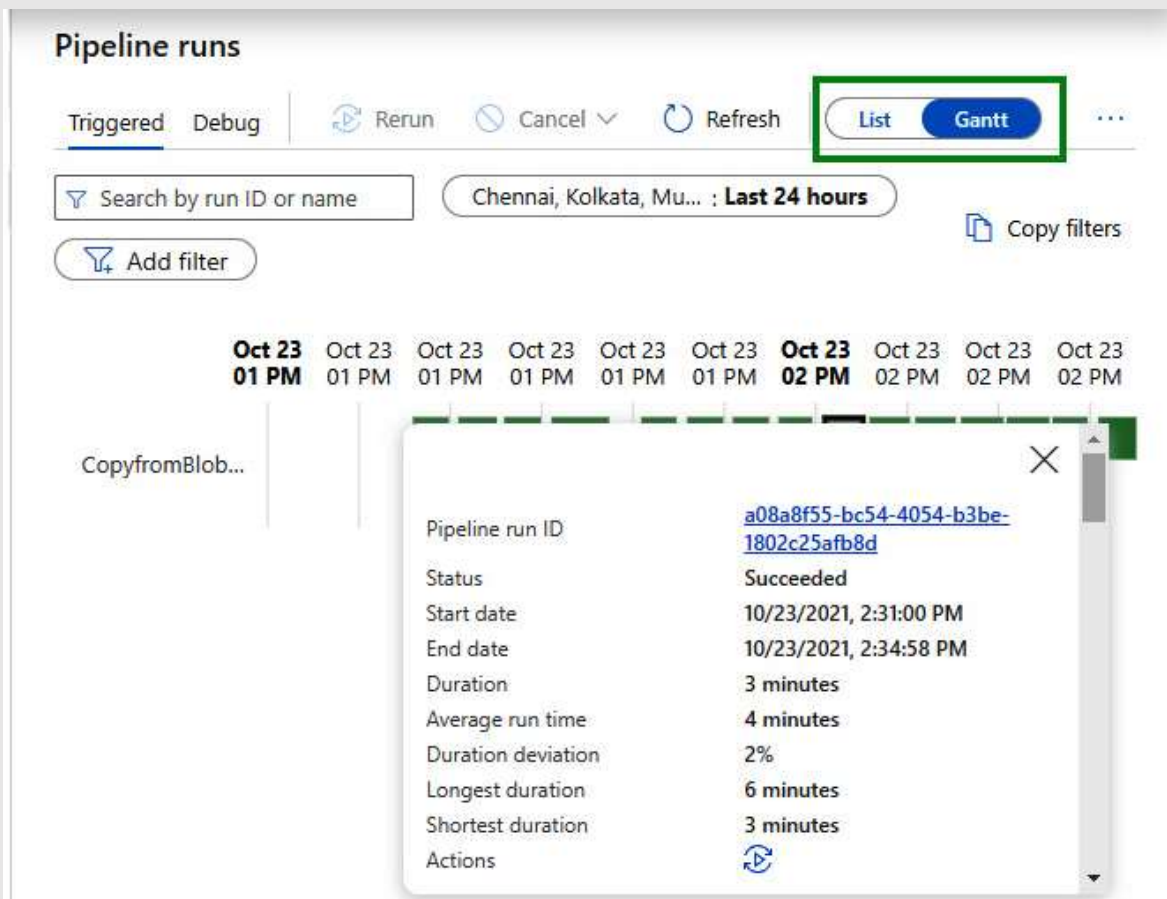


Figura 13.21 - Detalles adicionales del pipeline en la página del diagrama de Gantt

NOTA

El ADF sólo mantiene los detalles de la ejecución de la canalización y las métricas durante 45 días. Si necesita analizar los datos del pipeline durante más de 45 días, tendrá que enviar los datos a Log Analytics y luego utilizar las consultas de Kusto para obtener los detalles de ejecución.

Así es como podemos hacer un seguimiento del rendimiento del pipeline. Veamos a continuación cómo monitorizar y actualizar las estadísticas sobre los datos.

13.8. Monitorización y actualización de estadísticas sobre datos en un sistema

Las estadísticas son un concepto importante en la optimización de consultas. La generación de estadísticas es el proceso de recopilación de metadatos sobre los datos -como el número de filas, el tamaño de las tablas, etc.- que el motor SQL puede utilizar como entradas adicionales para optimizar los planes de consulta. Por ejemplo, si hay que unir dos tablas y una de ellas es muy pequeña, el motor SQL puede utilizar esta información estadística para elegir un plan de consulta que funcione mejor para esas tablas tan sesgadas. El motor SQL pool de Synapse utiliza algo conocido como optimizadores basados en los costes (CBO). Estos optimizadores eligen el plan de consulta menos costoso de un conjunto de planes de consulta que pueden ser generados para un determinado script SQL.

Veamos cómo crear estadísticas para los Synapse dedicated y serverless pools.

13.8.1. Creación de estadísticas para los Synapse dedicated pools

Puede habilitar las estadísticas en los Synapse SQL dedicated pools utilizando el siguiente comando:

```
ALTER DATABASE [DW_NAME] SET AUTO_CREATE_STATISTICS ON
```

Una vez que AUTO_CREATE_STATISTICS esté activado, cualquier sentencia SELECT, INSERT-SELECT, CTAS, UPDATE, DELETE o EXPLAIN desencadenará automáticamente la creación de estadísticas para las columnas implicadas en la consulta, si no están ya presentes.

NOTA

La creación automática de estadísticas no está disponible para las tablas temporales o externas.

Puede crear estadísticas bajo demanda utilizando el siguiente comando:

```
CREATE STATISTICS [statistics_name]  
ON [schema_name].[table_name]([column_name])  
WITH SAMPLE 40 PERCENT;
```

En el ejemplo anterior, estamos utilizando una muestra del 40%. Si no proporciona un valor de muestra, el valor por defecto es el 20%. También puede realizar un análisis completo en lugar de un muestreo utilizando el siguiente comando:

```
CREATE STATISTICS [statistics_name]  
ON [schema_name].[table_name]([column_name])  
WITH FULLSCAN;
```


CONSEJO

Como directriz general, para menos de 1.000 millones de filas, utilice el muestreo por defecto (20%). Con más de 1.000 millones de filas, utilice un muestreo del 2 al 10%.

A continuación, veamos cómo actualizar las estadísticas existentes en Synapse dedicated pools.

13.8.2. Actualización de las estadísticas de Synapse dedicated pools

De forma similar a la creación de estadísticas, también puede actualizarlas periódicamente. Para ello, puede utilizar el siguiente comando:

```
UPDATE STATISTICS [schema_name].[table_name]([stat_name]);
```

Es una buena práctica actualizar las estadísticas después de cada nueva carga de datos.

Para más información sobre las estadísticas en los pools de Synapse SQL, puede consultar el siguiente enlace: <https://docs.microsoft.com/en-us/azure/synapse-analytics/sql/develop-tables-statistics#statistics-in-dedicated-sql-pool>.

A continuación, vamos a ver cómo crear estadísticas para los Synapse serverless pools

13.8.3. Creando estadísticas para Synapse serverless pools

El concepto de estadísticas es el mismo para los dedicated y serverless pools. En el caso de los serverless pools, la creación automática de estadísticas está activada por defecto para los archivos Parquet, pero no para los archivos de valores separados por comas (CSV). Dado que tratamos con tablas externas en serverless pools, tendremos que crear estadísticas para las tablas externas. El comando para las tablas externas se muestra aquí:

```
CREATE STATISTICS [statistics_name]
ON { external_table } ( column )
WITH
    { FULLSCAN
      | [ SAMPLE number PERCENT ] }
, { NORECOMPUTE }
```

NOTA

En las tablas externas sólo son posibles las estadísticas de una columna, mientras que otras permiten las estadísticas de varias columnas.

Veamos a continuación cómo actualizar las estadísticas existentes para los serverless pools.

13.8.4. Actualizar las estadísticas de Synapse serverless pools

Para actualizar las estadísticas de los Synapse serverless pools, primero tendrá que eliminar las estadísticas existentes y luego volver a crearlas. El siguiente comando muestra cómo eliminar la tabla antigua:

```
DROP STATISTICS [OLD_STATISTICS_TABLE_NAME]
```

Y luego, podemos volver a crear las estadísticas de nuevo siguiendo los comandos que se muestran aquí:

```
CREATE STATISTICS [statistics_name]  
on [NEW_STATISTICS_TABLE_NAME] (STATENAME)  
WITH FULLSCAN, NORECOMPUTE
```

NOTA

La recopilación de estadísticas puede causar una ligera degradación del rendimiento si faltan estadísticas para las columnas de una consulta. Pero una vez generadas las estadísticas, las futuras consultas serán mucho más rápidas.

Si quiere saber más sobre la generación de estadísticas, puede consultar el siguiente enlace: <https://docs.microsoft.com/en-us/azure/synapse-analytics/sql/develop-tables-statistics#statistics-in-serverless-sql-pool>.

Veamos ahora cómo medir el rendimiento de las consultas.

13.9. Medición del rendimiento de las consultas

El rendimiento de las consultas es un tema muy interesante en las bases de datos y los motores analíticos como Spark y Hive. Encontrarás toneladas de libros y artículos escritos sobre estos temas. En esta sección, voy a tratar de dar una visión general de cómo supervisar el rendimiento de las consultas en los Synapse dedicated SQL pools y Spark. En el próximo capítulo, nos centraremos en cómo optimizar realmente las consultas. He proporcionado enlaces de lectura adicional en cada sección para que puedas aprender más sobre las técnicas si lo deseas.

Para medir el rendimiento de cualquier consulta basada en SQL, se recomienda configurar las Transaction Processing Performance Council H o DS (TPC-H o TPC-DS) y ejecutarlas de forma regular para identificar cualquier regresión en la plataforma. TPC-H y TPC-DS son conjuntos de pruebas de referencia estándar del sector. Si está interesado en saber más sobre ellos, siga estos enlaces:

Puede obtener más información sobre TPC-H aquí: <http://www.tpc.org/tpch/>.

Puede obtener más información sobre TPC-DS aquí: <http://www.tpc.org/tpcds/>.

Veamos a continuación cómo podemos monitorizar el rendimiento de Synapse SQL pool

13.9.1. Supervisar el rendimiento de Synapse SQL pool

La pagina de Metricas de Synapse SQL tiene una gran cantidad de metricas que pueden ser facilmente usadas para identificar cualquier regresion en el rendimiento. Aqui hay una captura de pantalla de ejemplo de la seccion de Metricas de Synapse SQL:

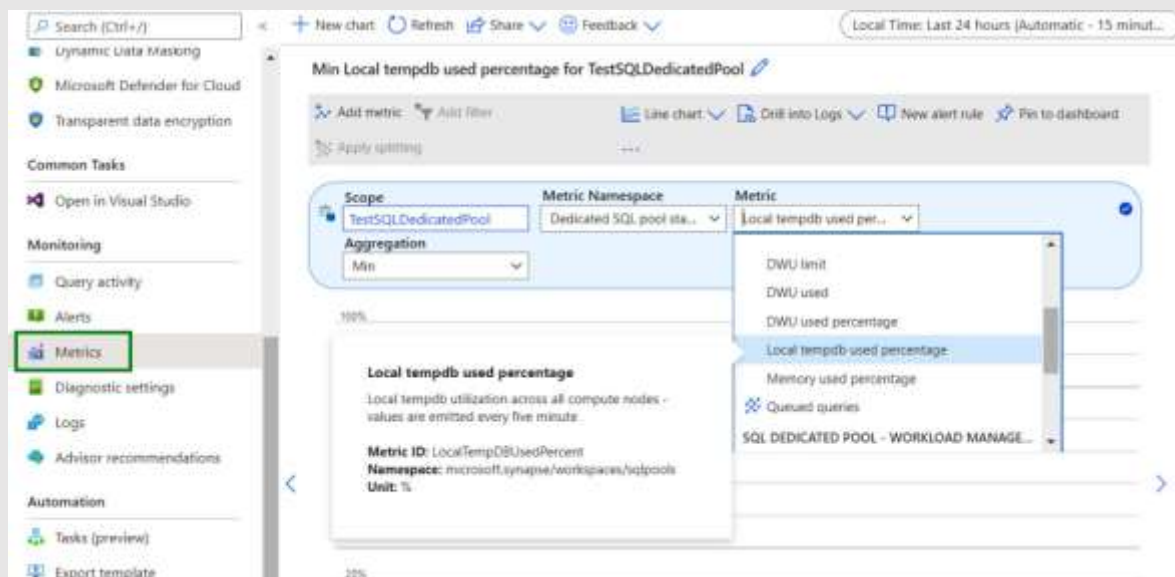


Figura 13.22 - Pantalla de monitorización de consultas de Synapse SQL pool

La pestaña de Métricas en Synapse SQL proporciona métricas para el uso de la Unidad de Almacenamiento de Datos (DWU), el uso de la memoria, etc., pero no proporciona detalles sobre el

rendimiento de la consulta o los tiempos de espera de la consulta. Tendremos que utilizar uno de los dos enfoques siguientes para obtener estos detalles de rendimiento de las consultas.

Consulta de las tablas del sistema

Synapse SQL pool proporciona las siguientes tablas del sistema que pueden utilizarse para supervisar el rendimiento de las consultas:

- **sys.dm_pdw_exec_requests**: contiene todas las solicitudes actuales y recientemente activas en Azure Synapse Analytics. Contiene detalles como `total_elapsed_time`, `submit_time`, `start_time`, `end_time`, `command`, `result_cache_hit`, etc.
- **sys.dm_pdw_waits**: contiene detalles de los estados de espera en una consulta, incluidos los bloqueos y las esperas en las colas de transmisión.

Puede encontrar detalles sobre todas estas tablas del sistema con información de supervisión y gestión aquí:

<https://docs.microsoft.com/en-us/sql/relational-databases/system-dynamic-management-views/sql-and-parallel-data-warehouse-dynamic-management-views>

A continuación, vamos a ver cómo obtener detalles de rendimiento utilizando una función llamada Query Store.

Usando Query Store

Azure SQL y Synapse SQL admiten una función llamada Query Store que puede utilizarse tanto para la supervisión del rendimiento de las consultas como para el ajuste del rendimiento. Se admite en Synapse SQL pool y en otras versiones de SQL en Azure. Query Store se utiliza para almacenar el historial de consultas, planes y estadísticas en tiempo de ejecución. Estos datos históricos pueden utilizarse para ayudar en la optimización de consultas, identificando regresiones en las consultas, monitorizando tendencias en la utilización de recursos como la CPU y la memoria para las consultas, identificando patrones de tiempo de espera en los planes de consulta, etc.

El Almacenamiento de Consultas no está habilitado por defecto para Synapse SQL pool. Puede habilitar Query Store usando el siguiente comando Transact-SQL (T-SQL):

```
ALTER DATABASE <database_name> SET QUERY_STORE = ON;
```

Query Store en Synapse SQL almacena los siguientes tres componentes principales:

- **Query details**-Detalles como los parámetros de la consulta, los tiempos de compilación, los recuentos de compilación, etc. Estas son las tablas específicas que contienen los detalles de las consultas de Query Store:

sys.query_store_query
sys.query_store_query_text

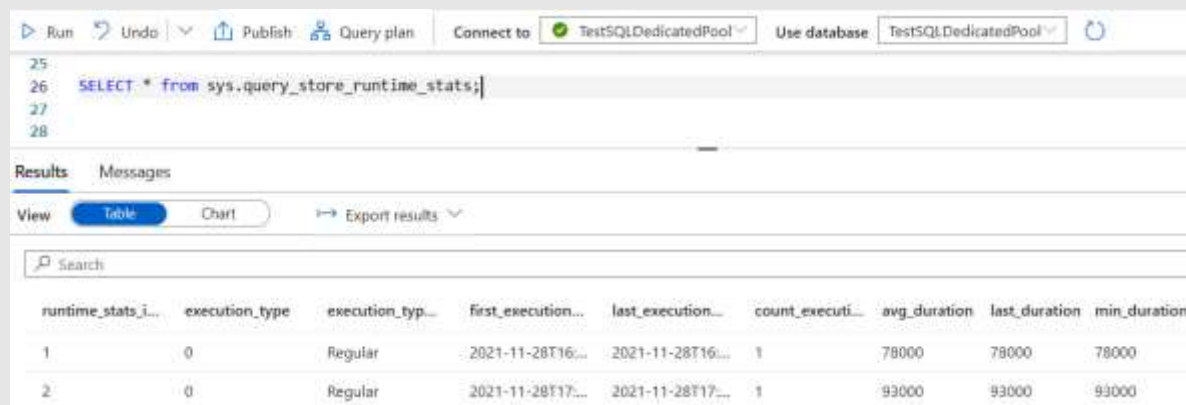
- **Plan details**-Detalles del plan de consulta como el query identifier (ID), query hash, query text, etc. Esta es la tabla específica que contiene los detalles del plan de Query Store:

sys.query_store_plan

- **Runtime statistics**-Detalles de tiempo de ejecución de las consultas, como el tiempo empleado, el tiempo de CPU, la duración media, el recuento de filas, etc. Estas son las tablas específicas que contienen estadísticas de tiempo de ejecución:

sys.query_store_runtime_stats
sys.query_store_runtime_stats_interval

Aquí hay una muestra de la tabla sys.query_store_runtime_stats para que tenga una idea de su aspecto:



runtime_stats_id	execution_type	execution_type_desc	first_execution_time	last_execution_time	count_executions	avg_duration	last_duration	min_duration
1	0	Regular	2021-11-28T16:...	2021-11-28T16:...	1	78000	78000	78000
2	0	Regular	2021-11-28T17:...	2021-11-28T17:...	1	93000	93000	93000

Figura 13.23 - Muestra de las estadísticas de tiempo de ejecución de Query Store

Las estadísticas de espera aún no están disponibles para Azure Synapse SQL, en el momento de escribir este libro. Sin embargo, están disponibles en otros sabores de Azure SQL, como Azure SQL Database.

Puedes aprender más sobre Query Store aquí: <https://docs.microsoft.com/en-us/sql/relational-databases/performance/monitoring-performance-by-using-the-query-store>.

Veamos a continuación las opciones disponibles para monitorizar el rendimiento de Spark.

13.9.2. Monitorización del rendimiento de las consultas de Spark

La monitorización del rendimiento de Spark puede realizarse directamente a través de la interfaz de usuario (UI) de Spark. La interfaz de usuario de Spark, junto con el servidor del historial de Spark, proporciona métricas específicas del job y stage que pueden utilizarse para determinar si los jobs se están ralentizando en comparación con las ejecuciones anteriores. Las técnicas que exploramos en el Capítulo 9, Diseño y desarrollo de una solución de procesamiento por lotes, en la sección Depuración de trabajos de Spark mediante la UI de Spark, serán las mismas para analizar la regresión del rendimiento en los jobs de Spark. Como ya hemos cubierto los detalles en ese capítulo, no los repetiremos aquí. Por favor, echa un vistazo al Capítulo 9, Diseño y desarrollo de una solución de procesamiento por lotes, si tienes alguna duda sobre el uso de la interfaz de usuario de Spark.

Una vez identificadas las consultas lentas a través de la UI de Spark, podemos utilizar técnicas de optimización de consultas para acelerarlas. Exploraremos estas técnicas en el próximo capítulo.

Mientras tanto, vamos a explorar otro tema relacionado con el ajuste del rendimiento: la interpretación de los DAGs de Spark.

13.10. Interpretando un DAG de Spark

Un DAG es un gráfico normal con nodos y aristas pero sin ciclos ni bucles. Para entender un DAG de Spark, primero tenemos que entender dónde entra en escena un DAG durante la ejecución de un Spark job.

Cuando un usuario envía un Spark job, el controlador de Spark primero identifica todas las tareas involucradas en la realización del job. A continuación, averigua cuáles de estas tareas pueden ejecutarse en paralelo y cuáles dependen de otras. Basándose en esta información, convierte el Spark job en un grafo de tareas. Los nodos del mismo nivel indican trabajos que pueden ejecutarse en paralelo, y los nodos de distinto nivel indican tareas que deben ejecutarse después de los nodos anteriores. Este grafo es acíclico, tal y como se denota con A en DAG. Este DAG se convierte en un plan de ejecución físico. En el plan de ejecución físico, los nodos que se encuentran en el mismo nivel se segregan en etapas. Una vez que todas las tareas y etapas se han completado, el Spark job se denomina completado.

Veamos qué aspecto tiene un DAG. Puedes acceder a un Spark DAG desde la interfaz de usuario de Spark. Sólo tienes que hacer clic en cualquiera de los enlaces de los jobs y luego hacer clic en el enlace de Visualización del DAG.

Aquí hay un DAG para un problema simple de recuento de palabras:

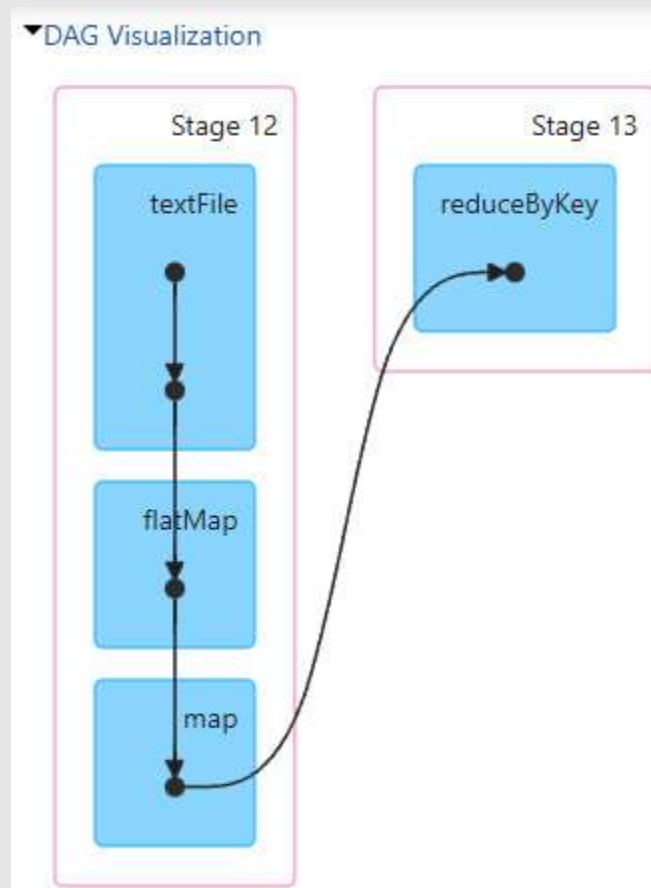


Figura 13.24 - DAG con múltiples etapas

En la primera etapa, vemos que el recuento de palabras tiene tres pasos y un paso de reduce en la siguiente etapa. Ignora los números de etapa, ya que Spark asigna números consecutivos para todos los jobs que se ejecutan en esa sesión de Spark. Por lo tanto, si se ha ejecutado cualquier otro job antes de este job, el número se incrementa secuencialmente. A continuación se ofrece más información sobre cada tarea:

- La **tarea textFile** corresponde a la lectura del archivo desde el almacenamiento.
- La **tarea flatMap** corresponde a la división de las palabras.
- La **tarea map** corresponde a la formación de pares (palabra, 1).
- La **tarea reduceByKey** corresponde a la agregación de todos los pares (palabra, 1) para obtener la suma de cada palabra distinta.

Puede obtener más detalles sobre cada paso haciendo clic en los cuadros de la etapa. A continuación se muestra un ejemplo de una vista detallada de la Etapa 12 de la captura de pantalla anterior:

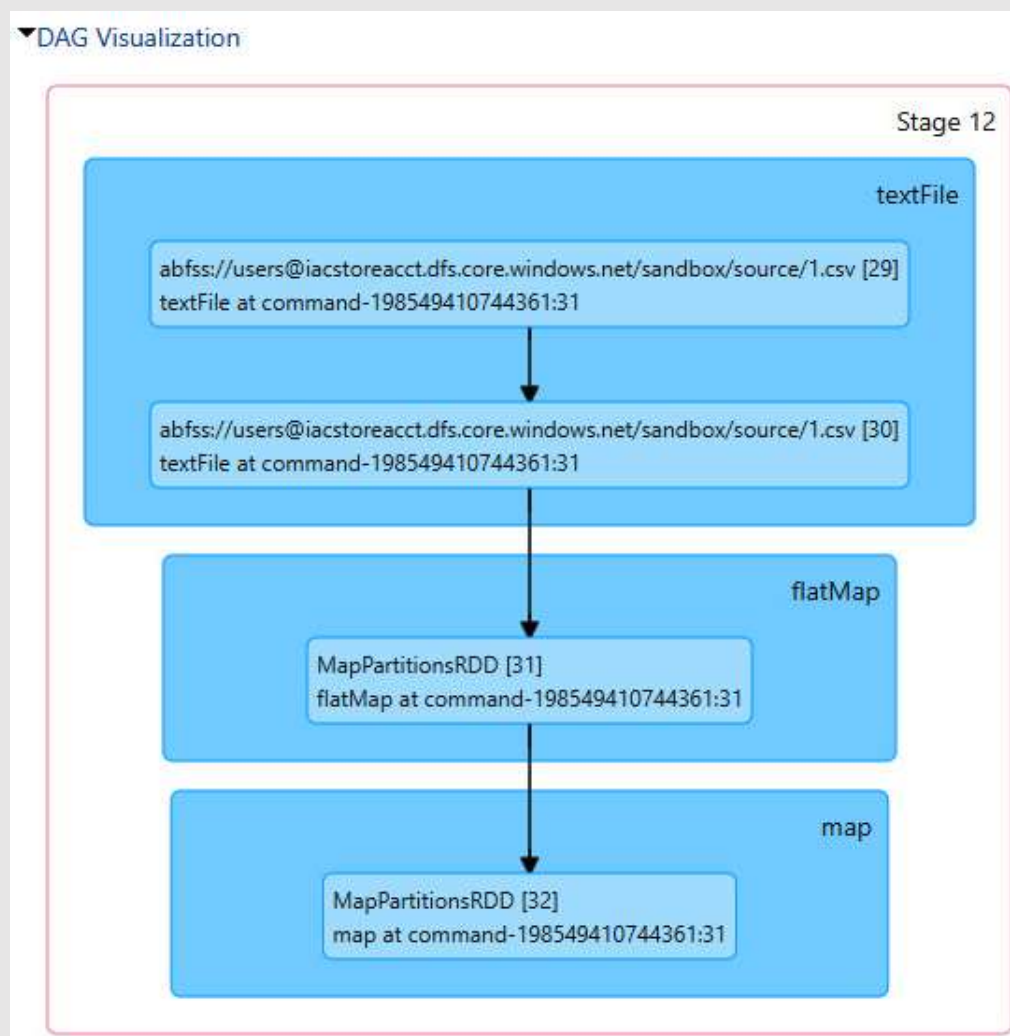


Figura 13.25 - Vista detallada de la etapa

La principal ventaja de aprender a leer Spark DAGs es que te ayudan a identificar los cuellos de botella en tus consultas de Spark. Puedes identificar cuánto movimiento de datos está ocurriendo entre etapas (también conocido como shuffle de datos), si hay demasiadas etapas secuenciales, si hay etapas lentas en la ruta crítica, etc.

Puedes aprender más sobre los Spark DAG aquí: <https://spark.apache.org/docs/3.0.0/web-ui.html>.

Ahora que hemos aprendido un poco sobre la monitorización del rendimiento de las consultas y el análisis de los Spark DAG, veamos también cómo monitorizar el rendimiento general del clúster.

13.11. Monitorización del rendimiento del clúster

Dado que servicios como Synapse y ADF son servicios de plataforma como servicio (PaaS) en los que no tendrás un control explícito sobre los clústeres, el único lugar donde podemos controlar todos y cada uno de los aspectos de un clúster es el servicio Azure HDInsight. En HDInsight, puedes crear tus propios clústeres de Hadoop, Spark, Hive, HBase y otros y controlar cada aspecto del clúster. Puede utilizar Log Analytics para supervisar el rendimiento del clúster, como en los otros ejemplos que vimos anteriormente en el capítulo. Puedes aprender más sobre el uso de Log Analytics en HDInsight aquí: <https://docs.microsoft.com/en-us/azure/hdinsight/hdinsight-hadoop-oms-log-analytics-tutorial>.

Aparte del enfoque de Log Analytics, hay cuatro áreas principales del portal de HDInsight que ayudan a supervisar el rendimiento del clúster. Veámoslas.

13.11.1. Supervisión del rendimiento general del clúster

El panel de control de HDInsight Ambari es el primer lugar para comprobar la salud del clúster. Si ves un uso muy elevado de la pila, del disco o de las latencias de la red, podrían ser indicadores de un uso no óptimo del clúster. Aquí hay una captura de pantalla de ejemplo del panel de control de Ambari:

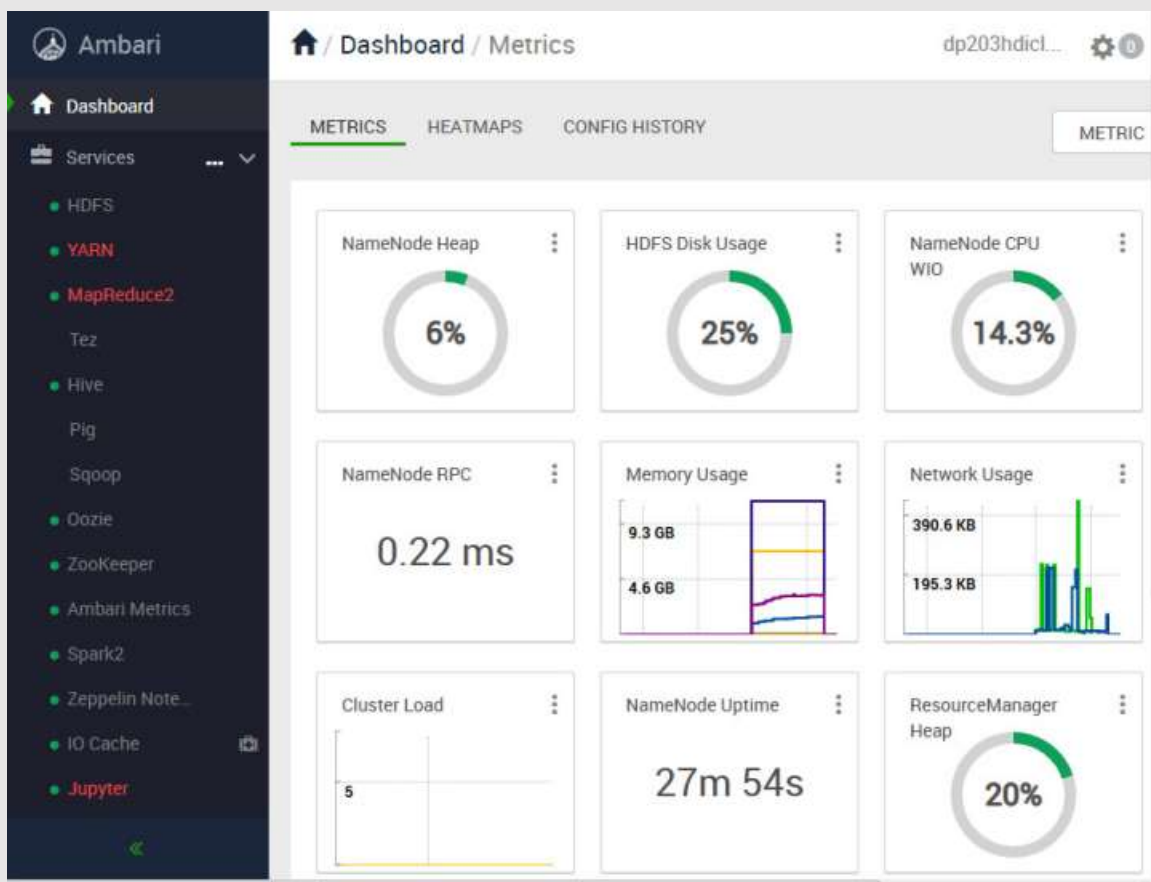


Figura 13.26 - Página de inicio del panel de control de HDInsight Ambari

Pero estos detalles generales del clúster no serán suficientes para aislar los problemas de rendimiento a nivel de nodo o de consulta. Para ello, tendremos que buscar la siguiente opción de monitorización disponible, que es un monitor por nodo.

13.11.2. Monitorización del rendimiento por nodo

Desde la página de inicio del panel de control de Ambari, si selecciona la pestaña Hosts, podrá ver el nivel de rendimiento por nodo, como se ilustra en la siguiente captura de pantalla:

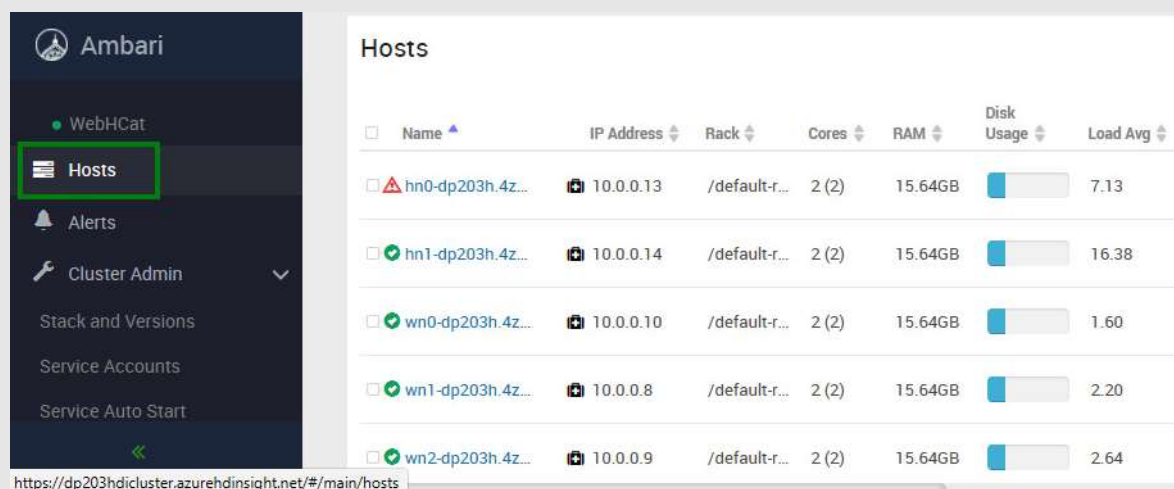


Figura 13.27 - Métricas por host en Ambari

Si haces clic en cada uno de los nombres de host, se mostrarán aún más detalles del host, como se muestra en la siguiente captura de pantalla:

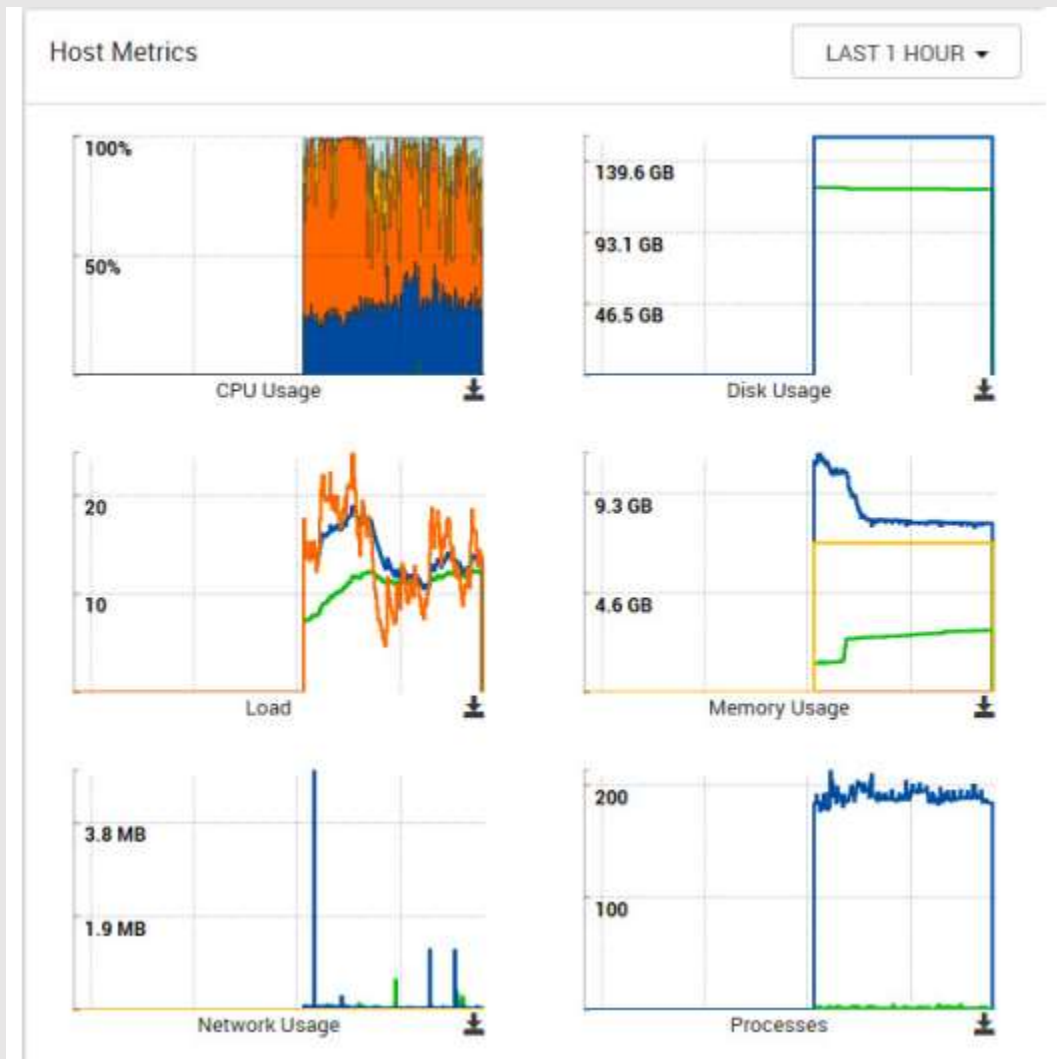


Figura 13.28 - Página de métricas por host en Ambari

Si quieres profundizar en el nivel de consulta, entonces podemos mirar las colas de Yet Another Resource Negotiator (YARN).

13.11.3. Monitorización de YARN queue/scheduler performance

Desde la página del Gestor de Recursos en Ambari, selecciona la opción Colas de Aplicación. Esto mostrará todas las aplicaciones y las métricas relacionadas con las aplicaciones. En el lado izquierdo, tendrás el enlace Scheduler, como se ilustra en la siguiente captura de pantalla. Selecciónelo para ver el estado de las colas de YARN:

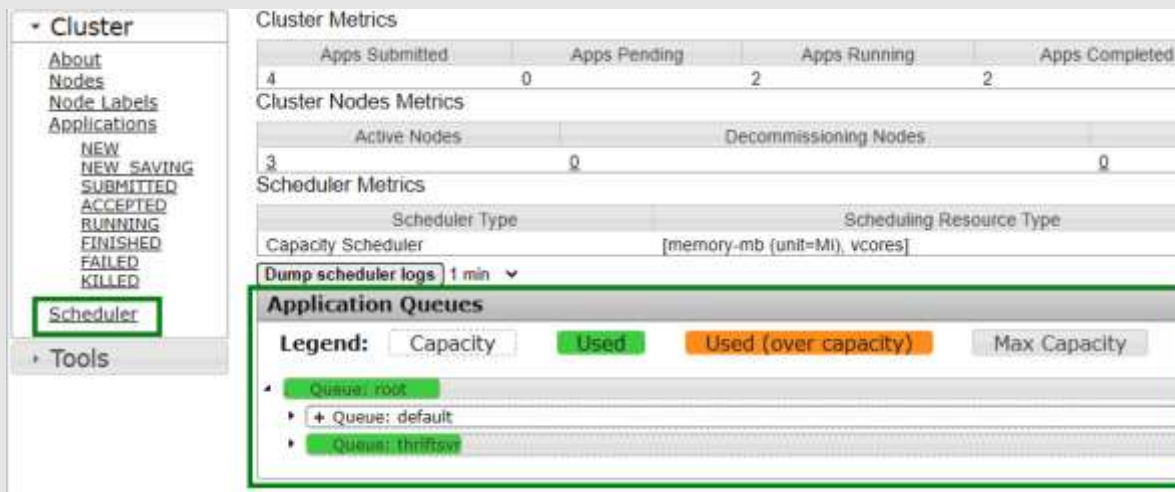


Figura 13.29 - Comprobación del estado de las colas YARN

Basándose en el color de la cola, debería ser capaz de distinguir si una cola está infrautilizada o sobreutilizada.

Hay un aspecto más de los clusters que suele dar lugar a un rendimiento más lento, y es el storage throttling.

13.11.4. Supervisión del storage throttling

Si obtienes un código de error 429 Too many requests en tus aplicaciones, puede que estés encontrando throttling desde el almacenamiento. Compruebe su cuenta de almacenamiento para aumentar el límite o intente distribuir el procesamiento de su aplicación en varias cuentas de almacenamiento para superar este problema.

Puedes aprender más sobre el throttling aquí: <https://docs.microsoft.com/en-us/azure/azure-resource-manager/management/request-limits-and-throttling>.

Espero que tengas una buena idea de cómo monitorizar el rendimiento del cluster. Ahora, veamos cómo programar y monitorizar la integración continua/despliegue continuo (CI/CD) del pipeline de ADF.

13.12. Programar y supervisar las pruebas de los pipelines

En el capítulo 11, Gestión de Batches y Pipelines, presentamos brevemente Azure DevOps para el control de versiones. Azure DevOps proporciona otra característica llamada Azure Pipelines, que puede ser utilizada para crear pipelines CI/CD para desplegar ADF. Si no conoces el CI/CD, es un método para probar y desplegar continuamente aplicaciones en el entorno de producción de forma automatizada. En esta sección, veremos cómo crear, programar y supervisar una canalización CI/CD.

NOTA

Al momento de escribir este libro, el soporte de Azure DevOps Pipelines para Synapse Pipelines no estaba disponible. Sólo está disponible para ADF.

Estos son los pasos de alto nivel para crear un pipeline de CI/CD utilizando pipelines de Azure:

1. Seleccione Azure DevOps en el portal de Azure. En la página de Azure DevOps, seleccione Releases en Pipelines y haga clic en el botón New Pipeline. Esto le llevará a una nueva pantalla, que se muestra en la siguiente captura de pantalla. Elige la opción Empty job:

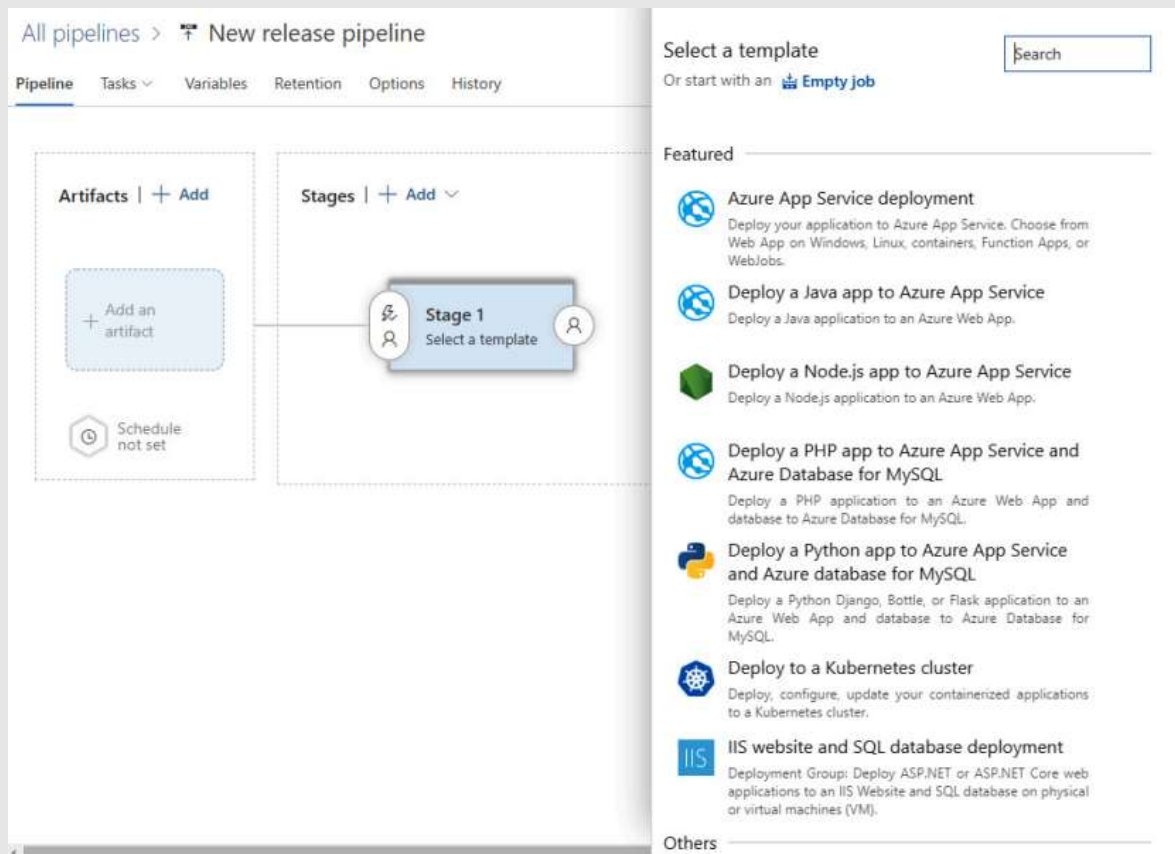


Figura 13.30 - Asistente de creación de pipelines de Azure

2. A continuación, haz clic en la opción Add an artifact y actualiza los detalles de tu repositorio de fuentes ADF, como se muestra en la siguiente captura de pantalla:

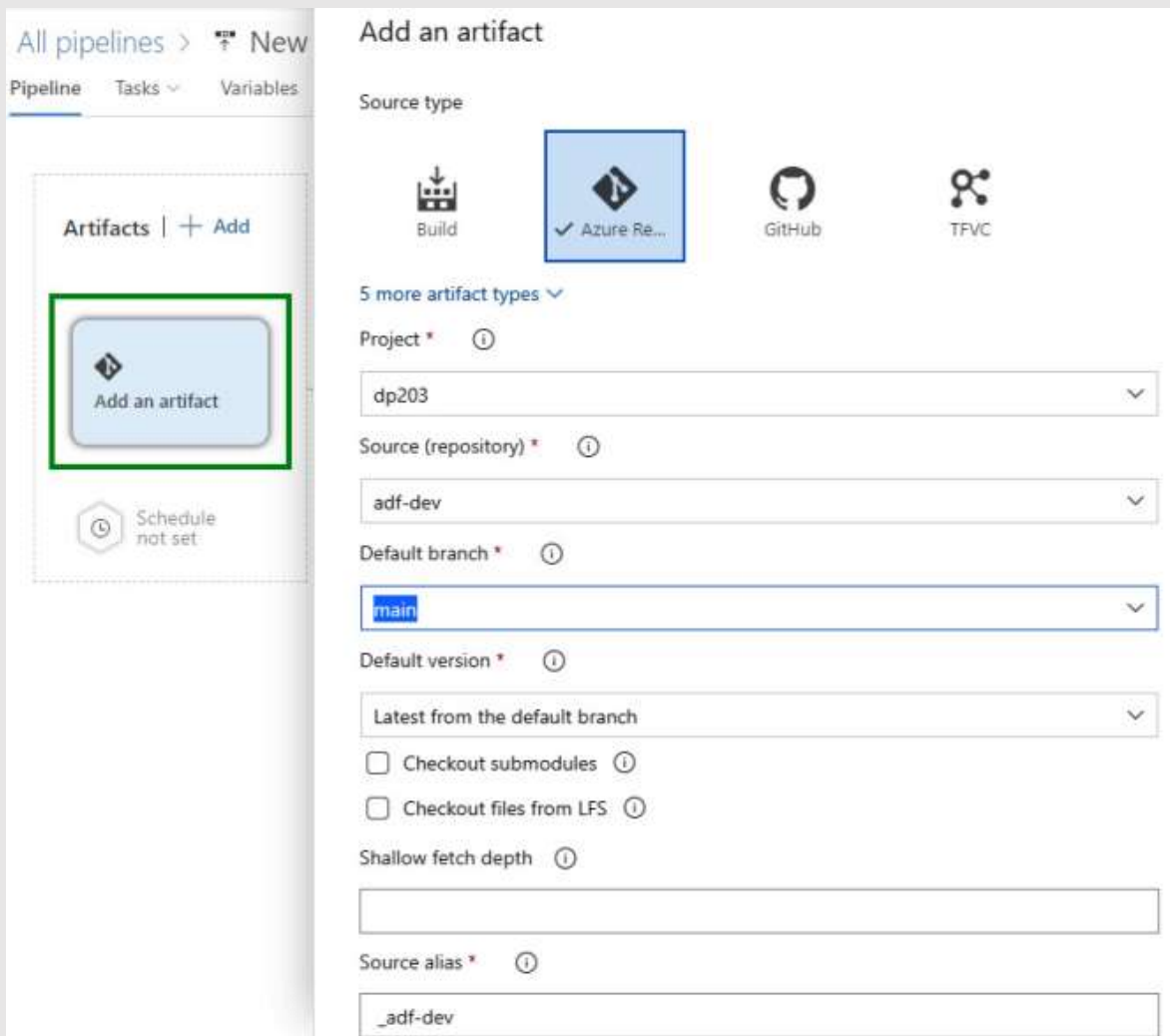


Figura 13.31 - Actualización de la información del artefacto en Azure Pipelines

3. Una vez añadido, haz clic en el enlace View Tasks en la siguiente página, y luego haz clic en el símbolo +. En la lista de tareas, elija la tarea de despliegue de la plantilla ARM, como se ilustra en la siguiente captura de pantalla:

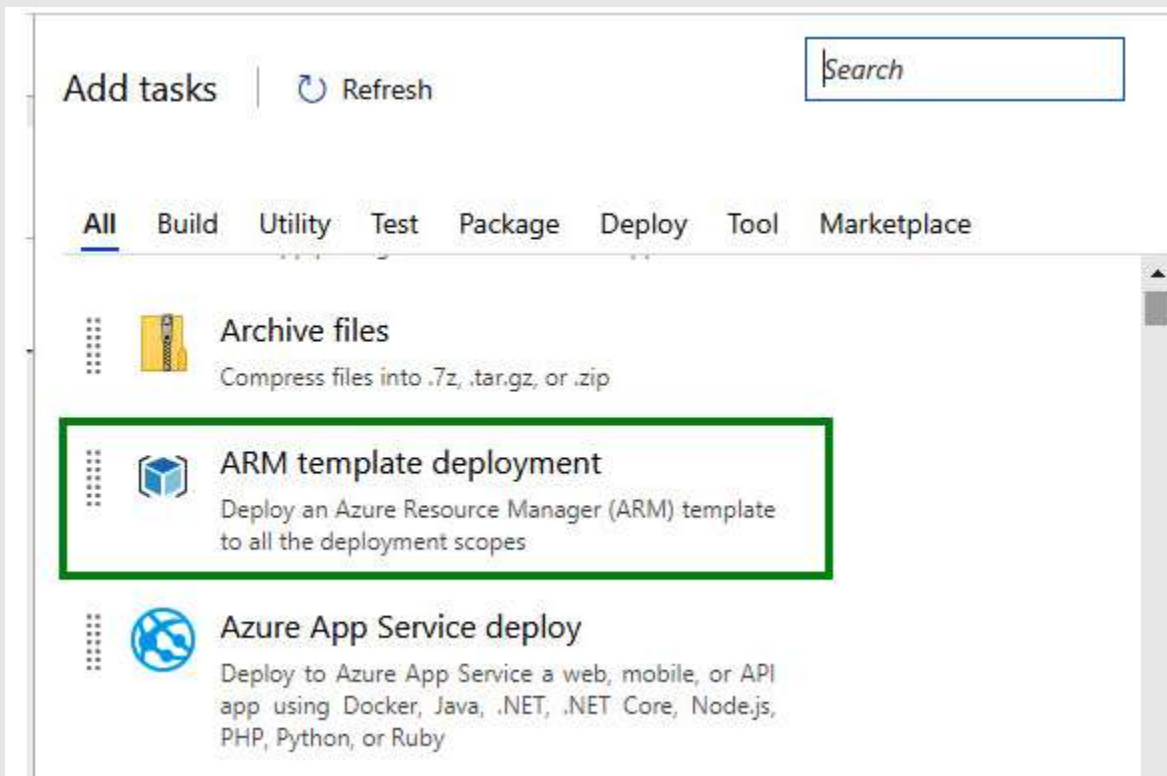


Figura 13.32 - Elección de tareas de Azure Pipeline

4. Esto hará que aparezca la página de configuración de la plantilla de Azure Resource Manager (ARM), como se muestra en la siguiente captura de pantalla:

The image shows the configuration page for the 'ARM template deployment' task. At the top, there is a 'Template' dropdown menu. Below it is the 'Template location' field, which is a dropdown menu currently showing 'Linked artifact'. The next field is 'Template', which is a text input field that is currently empty and has a red border. Below this field, there is a red message that says 'This setting is required.' To the right of the 'Template' field is a '...' button. Below the 'Template' field is the 'Template parameters' field, which is also a text input field. To the right of this field is another '...' button. At the bottom is the 'Override template parameters' field, which is a text input field. To the right of this field is a third '...' button. Each of these three fields (Template parameters and Override template parameters) has an information icon (i) to its right.

Figura 13.33 - Página de especificación de la plantilla ARM y de los parámetros de la plantilla

5. Para el cuadro de texto Plantilla, busque el archivo llamado ARMTemplateForFactory.json en su carpeta ADF de la rama adf_publish.
6. Para el cuadro de texto Parámetros de la plantilla, elija el archivo ARMTemplateParametersForFactory.json en su carpeta ADF de la rama adf_publish.
7. Una vez que haya completado los detalles, guarde la página.
8. Por último, haga clic en el botón Create Release en la parte superior para crear una tubería CI/CD.

Ahora que tenemos un pipeline para desplegar CI/CD, se puede programar de múltiples maneras. A continuación se ofrecen algunos ejemplos:

- **CI triggers**-Para activar un pipeline cuando un usuario empuja un cambio a la rama git
- **Pull request (PR) triggers**-Cuando un PR es levantado o los cambios son enviados al PR
- **Scheduled triggers**-Programación basada en el tiempo
- **Pipeline completion triggers**-Triggering basado en la finalización de pipelines anteriores

Puede obtener más información sobre la implementación de disparadores aquí: <https://docs.microsoft.com/en-us/azure/devops/pipelines/build/triggers?view=azure-devops>.

Una vez que tenemos los triggers configurados, los pipelines continúan siendo probados y desplegados. Podemos monitorizar el progreso desde la pestaña Releases de Azure DevOps. Aquí tenemos una imagen de muestra de la pantalla de Pipelines con un resumen de los pipelines que se han ejecutado:

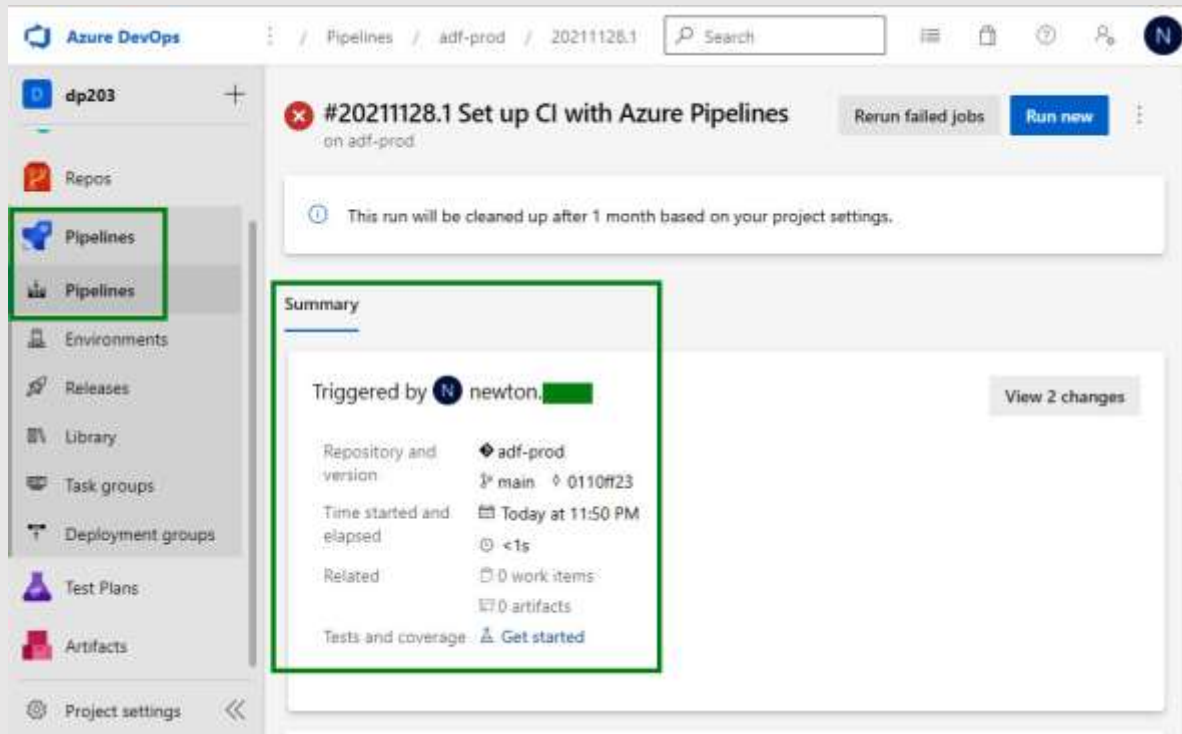


Figura 13.34 - Monitorización de los pipelines de release CI/CD

Azure DevOps y Pipelines son un tema enorme por sí mismos y no podremos hacerles justicia en una pequeña sección aquí, así que estoy proporcionando enlaces de seguimiento para entender más sobre ADF CI/CD, Azure pipelines, y la adición de pruebas a los pipelines.

Puedes aprender más sobre ADF CI/CD aquí: <https://docs.microsoft.com/en-us/azure/data-factory/continuous-integration-delivery>.

Resumen

Este capítulo ha introducido un montón de nuevas tecnologías y técnicas, y espero que las hayas entendido. Aunque el número de tecnologías involucradas es alto, el peso de este capítulo con respecto a la certificación es relativamente bajo, por lo que habrás notado que he mantenido los temas ligeramente en un nivel más alto y he proporcionado más enlaces para que puedas leer más sobre los temas.

En este capítulo, comenzamos introduciendo Azure Monitor y Log Analytics. Aprendimos a enviar datos de log a Log Analytics, a definir opciones de logging personalizadas y a interpretar los datos de las métricas y los logs. Después, nos centramos en la medición del rendimiento de los movimientos de datos, los pipelines, las consultas SQL y las consultas Spark. También aprendimos a interpretar los Spark DAG, antes de pasar a supervisar el rendimiento del clúster y las pruebas de pipelines del clúster. Ahora deberías ser capaz de configurar una solución de monitorización para tus pipelines de datos y ser capaz de saber si tus movimientos de datos, configuraciones de pipelines, configuraciones de cluster y ejecuciones de consultas están teniendo un rendimiento óptimo.

En el próximo capítulo, nos centraremos en la optimización de las consultas y en las técnicas de resolución de problemas.