



Migración de trabajos ETL de SSIS a AWS

AWS Orientación prescriptiva



AWS Orientación prescriptiva: Migración de trabajos ETL de SSIS a AWS

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Las marcas comerciales y la imagen comercial de Amazon no se pueden utilizar en relación con ningún producto o servicio que no sea de Amazon, de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas registradas que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

Introducción	1
Resultados empresariales específicos	1
Fases de migración	2
Fase de detección	2
Fase de análisis	4
Determinar el enfoque de migración	5
Fase de implementación	7
AWS SCT	8
AWS Glue	8
Amazon EMR	10
Testeo	10
Prácticas recomendadas	12
Preguntas frecuentes	14
¿Debo mejorar los trabajos de SSIS durante la migración?	14
¿Cómo puedo supervisar los trabajos? AWS	14
¿Cuándo puedo retirar mis trabajos de SSIS locales?	14
Pasos a seguir a continuación	15
Recursos relacionados	16
Historial de documentos	17
.....	xviii

Migración de trabajos ETL de SSIS a AWS

Durga Prasad y Harpreet Singh, Amazon Web Services (AWS)

Diciembre de 2021 ([historial del documento](#))

Amazon Web Services (AWS) proporciona servicios para desarrollar y ejecutar trabajos de extracción, transformación y carga (ETL) o extracción, carga y transformación (ELT) y cargas de trabajo de macrodatos. Microsoft SQL Server Integration Services (SSIS) es una herramienta ETL local que tiene una interfaz gráfica para crear trabajos de ETL. Según la arquitectura del estado de destino, puede utilizar la interfaz AWS Glue gráfica o el marco Apache Spark con bibliotecas de Python para crear trabajos de ETL en la AWS nube.

Esta guía describe los pasos para migrar los trabajos de ETL/ELT de SSIS de un entorno local a otro. AWS En ella se analizan las fases de migración, las mejores prácticas y las recomendaciones para reducir el esfuerzo de migración y mejorar la experiencia. La información es aplicable a cualquier AWS arquitectura de destino.

La guía está destinada a directores de programas o proyectos, propietarios de productos, arquitectos de soluciones y desarrolladores que:

- La migración del SSIS al SSIS se AWS debe a varios motivos, como la modernización o el ahorro de costes.
- Utilizan principalmente AWS servicios para cargas de trabajo de ETL y macrodatos.
- Está buscando una solución basada en la nube para aprovechar un modelo sin servidor y unos precios flexibles.

Resultados empresariales específicos

La migración de sus trabajos de SSIS a la AWS nube le ayuda a lograr estos resultados principales:

- Modernice los procesos ETL locales existentes de forma rentable
- Responda a las necesidades de información de la organización con mayor rapidez
- Combine los procesos existentes y retire los procesos no utilizados para reducir los gastos operativos y de mantenimiento
- Cree una base para cumplir con los requisitos de datos futuros de su organización

Fases de migración

La migración de los procesos de ETL del SSIS a la AWS nube consta de cuatro fases principales: descubrir, analizar, determinar, abordar e implementar.



Estas fases se describen en las siguientes secciones:

- [Fase de descubrimiento](#)
- [Fase de análisis](#)
- [Determinar el enfoque de migración](#)
- [Fase de implementación](#)

Fase de detección

En la fase de descubrimiento, se crea una lista de los paquetes de SSIS a los que se quiere migrar. AWS Los diferentes equipos de desarrollo siguen diferentes estilos, estándares y patrones para desarrollar trabajos de ETL. Le recomendamos que revise los documentos existentes de su organización para comprender estos patrones. Sin embargo, la documentación suele estar incompleta. Puede automatizar la extracción de información importante de los scripts de ETL. Esto ahorra tiempo y esfuerzo manual, reduce los errores humanos y estandariza el enfoque de migración. Estos son algunos de los detalles importantes que querrá extraer:

- Número total de tareas de flujo de control
- Detalles de las tareas del flujo de control
- Número total de tareas de flujo de datos
- Transformaciones de flujo de datos utilizadas
- Controladores de eventos
- Administradores de conexiones

Utilice esta información para comprender los patrones de ETL que se utilizan en su organización, evaluar su complejidad e identificar el AWS servicio adecuado al que migrar esta información.

La migración de estos detalles de ETL desde el SSIS constituye la mayor parte del esfuerzo de migración. Sin embargo, las propiedades adicionales pueden proporcionar información sobre las decisiones de diseño y arquitectura. Algunas de estas propiedades del SSIS son:

- [Puntos de control](#), que se utilizan en el SSIS para reiniciar los trabajos desde puntos de fallo
- [Propague variables](#), que ayudan a que un paquete de SSIS tenga éxito en casos de uso específicos, incluso cuando se produce un error
- [Niveles de aislamiento de transacciones](#), que controlan la calidad de los datos que se leen de las bases de datos
- [Registro](#), para comprender los tipos de registros que captura el diseño actual y sus ubicaciones de almacenamiento

El resultado de la fase de descubrimiento puede ser un inventario, como se muestra en la siguiente tabla.

inventory

count	Package	Flow	Task
1	SSIS_Package_1	control_flow	Microsoft.ExecuteSQLTask
1	SSIS_Package_1	data_flow	Microsoft.BDD
1	SSIS_Package_1	data_flow	Microsoft.ConditionalSplit
2	SSIS_Package_1	data_flow	Microsoft.OLEDBDestination
1	SSIS_Package_1	data_flow	Microsoft.OLEDBSource
1	SSIS_Package_2	control_flow	Microsoft.DbMaintenanceFileCleanupTask
1	SSIS_Package_2	control_flow	Microsoft.ExecuteSQLTask
1	SSIS_Package_2	control_flow	Microsoft.ScriptTask
1	SSIS_Package_2	control_flow	STOCK:FOREACHLOOP
1	SSIS_Package_2	control_flow	STOCK:FORLOOP

Este inventario puede incluir la siguiente información:

- Paquete: Nombre del paquete SSIS que se va a migrar
- Flujo: flujo [de control o flujo de datos](#)
- Tarea: nombre de la tarea del flujo de control o del componente del flujo de datos
- Recuento: número de veces que se utilizó una tarea en el paquete SSIS

Fase de análisis

El análisis de la información de la fase de descubrimiento le ayuda a comprender los patrones y a diseñar mejor la arquitectura de destino. Siga estos consejos para mejorar el resultado del análisis:

- La opción de registro a [nivel de paquete](#) registra los eventos y captura la información de tiempo de ejecución de cada componente. Utilice el registro personalizado para poder configurar los archivos de registro para que incluyan información adicional, como los identificadores de ejecución y otros valores de tiempo de ejecución.
- Redirija los registros incorrectos a una ubicación de almacenamiento diferente para su análisis, corrección y reprocesamiento.
- Conozca las estrategias de archivado y depuración de datos implementadas en su entorno SSIS local.
- [Envíe alertas y notificaciones mediante tareas incluidas en el SSIS \(como la tarea Enviar correo\) o scripts personalizados \(como la tarea Script\).](#)
- Comprenda el comportamiento de [las transformaciones](#) en el ámbito de aplicación para evitar que los datos se corrompan. Por ejemplo, la [transformación de búsqueda](#) es una unión equivalente entre dos conjuntos de datos y su comparación distingue entre mayúsculas y minúsculas.

Puede asignar cada entrada del inventario a una complejidad estimada, ya sea en términos de duración (minutos u horas) o de nivel (simple, media o compleja). Este inventario ampliado, que se muestra en la siguiente tabla, es el resultado de la fase de análisis.

inventory

count	Package	Flow	Task	Effort	Complexity
1	SSIS_Package_1	control_flow	Microsoft.ExecuteSQLTask	30	S
1	SSIS_Package_1	data_flow	Microsoft.BDD	0	N/A
1	SSIS_Package_1	data_flow	Microsoft.ConditionalSplit	30	S
2	SSIS_Package_1	data_flow	Microsoft.OLEDBDestination	60	M
1	SSIS_Package_1	data_flow	Microsoft.OLEDBSource	30	S
1	SSIS_Package_2	control_flow	Microsoft.DbMaintenanceFileCleanupTask	60	M
1	SSIS_Package_2	control_flow	Microsoft.ExecuteSQLTask	30	S
1	SSIS_Package_2	control_flow	Microsoft.ScriptTask	120	C
1	SSIS_Package_2	control_flow	STOCK:FOREACHLOOP	30	S
1	SSIS_Package_2	control_flow	STOCK:FORLOOP	30	S

Determinar el enfoque de migración

Para decidir el enfoque de migración, utilice el análisis que realizó sobre los patrones existentes en la fase anterior. Las necesidades futuras de datos y análisis de su organización son consideraciones igualmente importantes. Las herramientas ETL locales tradicionales se ocupan de modelos de datos relacionales y datos estructurados. Si tiene que procesar datos semiestructurados y no estructurados, puede utilizar AWS servicios como AWS Glue Amazon EMR para la migración. Otros factores que pueden influir en el enfoque de migración son:

- Si desea utilizar una interfaz gráfica (como [AWS Glue Studio](#)) o un marco personalizado (como las bibliotecas Spark/Python)
- Si tiene acceso seguro a las fuentes y los destinos locales AWS
- Habilidades y formación necesarias para el equipo
- Requisitos de auditoría y cumplimiento

Puede seleccionar entre tres enfoques de migración: big bang, gradual y lift and shift. En la siguiente tabla se comparan estos tres enfoques.

Enfoque	Descripción	Caso de uso	Ventajas y desventajas
Big bang	Migre todos los paquetes de SSIS en un período de tiempo específico.	• La complejidad, el alcance y la arquitectura objetivo son claros. • El equipo tiene las habilidades requeridas o la curva de aprendizaje es superficial.	• Riesgo alto. • Lleva menos tiempo que el enfoque gradual. • Puede utilizar AWS Glue Amazon EMR o marcos personalizados.
En fases	Identifique un paquete de SSIS para cada patrón y complejidad distintos. Migre el paquete a la arquitectura existente AWS, pruebe y compare los resultados con ella.	• El tiempo no es una limitación. • Quiere diferentes diseños para diferentes patrones de ETL.	• Es menos arriesgado que el enfoque del big bang, pero requiere más tiempo y esfuerzo. • Puede utilizar AWS Glue Amazon EMR o marcos personalizados.
Levante y cambie de posición	Migre la arquitectura actual tal cual AWS.	• Su hardware local ya no es compatible. • No tiene los recursos para planificar una migración de forma inmediata.	• La cantidad mínima de esfuerzo y tiempo de migración necesarios. • Los problemas con la solución existente persisten en AWS. • Los paquetes SSIS se ejecutan tal cual. No se necesitan

Enfoque	Descripción	Caso de uso	Ventajas y desventajas
			otras herramientas o marcos de ETL.

La comparación de los datos de los sistemas de origen y destino es fundamental para una migración exitosa. Dado que el sistema de producción existente recibe actualizaciones periódicas desde los sistemas de origen, esta comparación puede resultar confusa. Por este motivo, al determinar el enfoque de migración, le recomendamos que también decida su estrategia de validación de datos.

- Realice copias de seguridad de todas las bases de datos y archivos aplicables del entorno de producción en el sistema de origen en una fecha y hora específicas.
- Realice copias de seguridad de todas las bases de datos del entorno de producción en el sistema de destino una vez que todos los trabajos hayan cargado correctamente los datos de los datos de origen respaldados.
- Restaure los datos de origen en un entorno de pruebas y ejecute los nuevos trabajos.
- Acuerde un porcentaje de diferencias válidas entre las bases de datos de origen y destino (antiguas y nuevas). Por ejemplo, puede decidir que una diferencia inferior al 1% es aceptable.
- Enumere todas las reglas de validación que se van a cubrir.
- Automatice la comparación en la medida de lo posible y cubra todas las reglas.

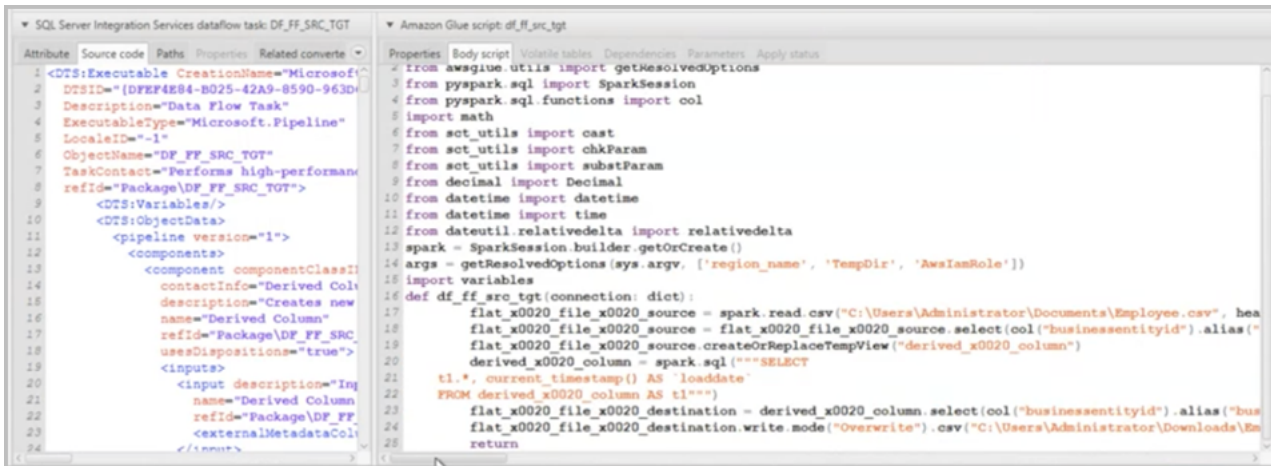
Fase de implementación

La migración que sigue un enfoque gradual o gradual requiere nuevos desarrollos y pruebas. El AWS Schema Conversion Tool (AWS SCT) puede generar automáticamente AWS Glue trabajos a partir de los paquetes del SSIS. Esto reduce significativamente el tiempo y el esfuerzo de migración. O bien, puedes usar AWS Glue Studio para el desarrollo basado en una interfaz gráfica o crear bibliotecas de Spark que puedas ejecutar en Amazon EMR o en AWS Glue Amazon EMR.

En las siguientes secciones se proporcionan consejos útiles sobre el uso de AWS SCT AWS Glue, y Amazon EMR.

AWS SCT

La siguiente ilustración de pantalla muestra un script de AWS Glue trabajo que fue convertido por AWS SCT

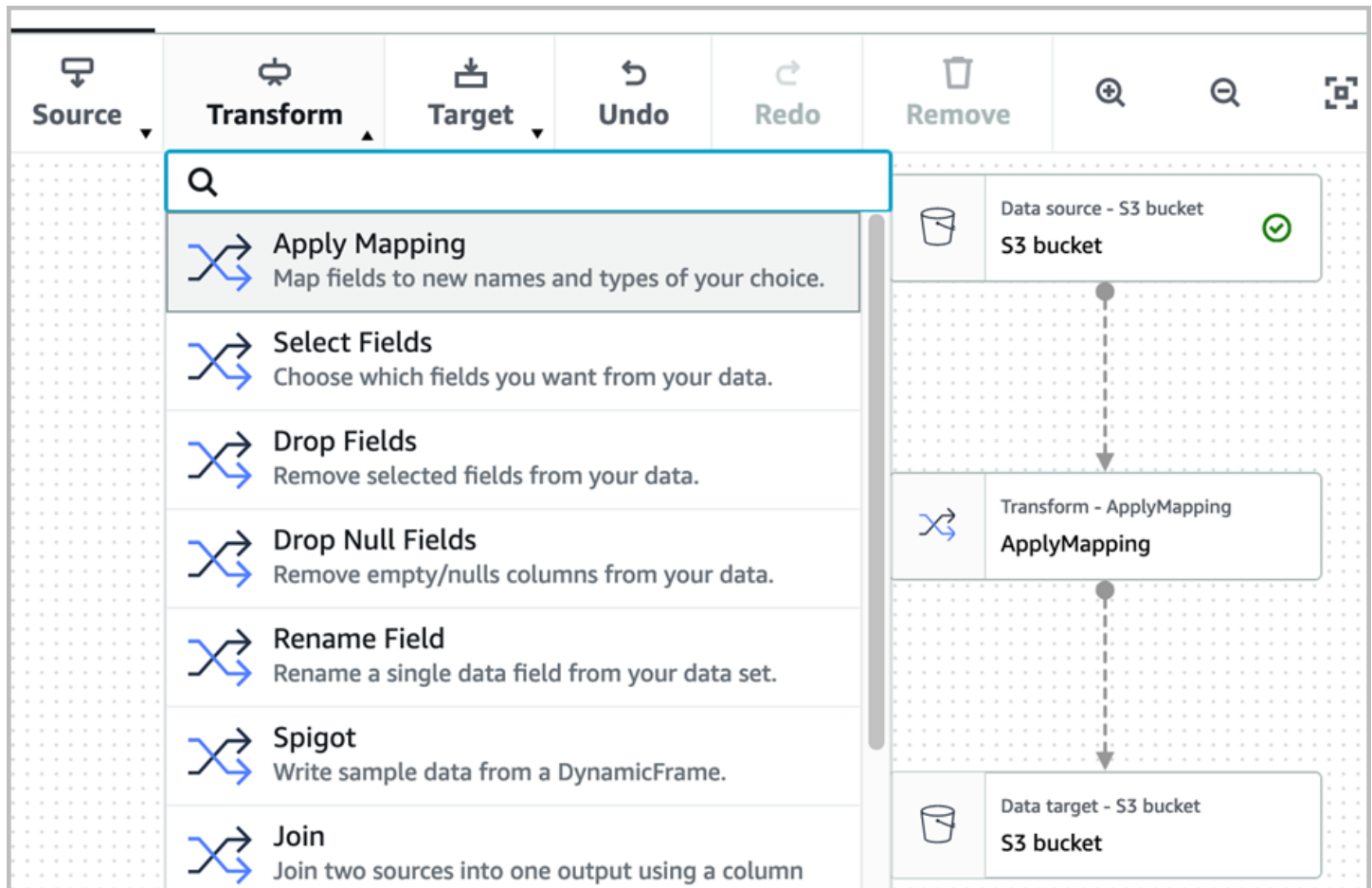


AWS SCT puede convertir paquetes SSIS en AWS Glue trabajos de forma masiva. Puede editar el script para actualizar la lógica existente o añadir una nueva, en función de su nuevo diseño. Le recomendamos que siga las convenciones de nomenclatura de los guiones AWS SCT convertidos para personalizarlos.

Para obtener más información, consulte [Convertir el SSIS para AWS Glue usarlo AWS SCT](#) en la AWS SCT documentación.

AWS Glue

AWS Glue Studio proporciona una interfaz gráfica y una experiencia de desarrollo similar a la de SSIS, como se muestra en la siguiente pantalla.



Si prefiere no utilizar una interfaz gráfica, también puede ejecutar sus scripts personalizados con las bibliotecas de Python necesarias desde la AWS Glue consola. Para obtener más información, consulte [Proporcionar sus propios scripts personalizados](#) en la AWS Glue documentación.

AWS Glue proporciona un conjunto de transformaciones integradas para procesar los datos. Son similares a las transformaciones de flujo de datos del SSIS. Siga estas prácticas recomendadas al migrar sus trabajos de ETL de SSIS mediante: AWS Glue

- Prepare un mapeo desde las AWS Glue transformaciones hasta las transformaciones de SSIS equivalentes.
- Si sus transformaciones no se pueden asignar a AWS Glue transformaciones, créelas mediante un script personalizado de Python o Scala.
- Para el registro personalizado (como filas leídas, filas escritas o registros incorrectos), usa scripts personalizados además de [Amazon CloudWatch](#).
- Añada un [terminal de desarrollo](#) para desarrollar y depurar scripts personalizados de forma local.

Amazon EMR

Puede ejecutar scripts personalizados (escritos en Python o Scala) o bibliotecas de Python compiladas en clústeres de EMR, como ocurre con. AWS Glue Siga estas prácticas recomendadas:

- Comience con tipos de instancias optimizados para memoria y, al mismo tiempo, cree clústeres de EMR con el marco Spark. (SSIS usa búferes de memoria).
- Cree métodos genéricos de Python que sean equivalentes a cada tarea o transformación de SSIS. Por ejemplo, en la siguiente ilustración, un método que toma dos marcos de datos como entrada produce un tercer marco de datos con registros coincidentes de los dos marcos de datos como salida. [Esto funciona como una transformación de fusión y unión.](#)

```
AWS: Add Debug Configuration | AWS: Edit Debug Configuration
def merge_join_spark(spark: SparkSession, df1: DataFrame, df2: DataFrame, join_type: str, join_column:str,
                    merge_fetch_columns: str):
    """Merge/Join directly
    spark: Current Spark session
    df1: First dataframe for merge join
    df2: Second dataframe for merge join
    join_type: Left/Inner/Outer join
    join_column : Column to be merge joined on
    merge_fetch_columns: Columns required in the output dataframe
    """
    try:
        spark.catalog.dropTempView("df1")
        spark.catalog.dropTempView("df2")
        df1.createOrReplaceTempView("df1")
        df2.createOrReplaceTempView("df2")
        merge_sql = f"select src.*, {merge_fetch_columns} from df1 {join_type} join df2 on df1.{join_column} = df2.{join_column}".format(
            merge_fetch_columns=merge_fetch_columns, join_type=join_type, join_column=join_column)
        logger.debug("Merge query is : " + merge_sql)
        output_df = spark.sql(merge_sql)
    except Exception as ex:
        logger.error("Merge Execution Failed for " + str(ex))
        raise Exception("Merge Execution Failed " + str(ex))
    finally:
        spark.catalog.dropTempView("df1")
        spark.catalog.dropTempView("df2")
    return output_df
```

Testeo

Se requiere un marco de pruebas para validar la integridad y exactitud de los datos. Este marco debe cubrir todos los escenarios existentes y cualquier mejora que haya realizado al migrar sus trabajos.

AWS

- Validación de integridad:
 - Todos los trabajos se migran a su estado objetivo.
 - Todas las funciones se migran en cada trabajo.

- Están disponibles todos los tipos de registros, incluidos los detalles de la ejecución de los trabajos, los mensajes de error, los registros incorrectos y el recuento de filas.
- Validación de corrección:
 - La calidad de los datos es uniforme en los entornos nuevos y existentes.
 - Todas las columnas de todas las tablas coinciden o las tablas están mejoradas AWS.
 - Toda la información de auditoría y registro coincide.

También debe comprobar que el rendimiento de los trabajos migrados coincide con el rendimiento de los trabajos actuales.

Prácticas recomendadas

Siga estas prácticas recomendadas generales al migrar sus trabajos de SSIS a: AWS

- Si va a migrar sus bases de datos a un servicio AWS gestionado, como Amazon Relational Database Service (Amazon RDS), elimine las tareas de [mantenimiento](#) (como la copia de seguridad, la restauración, la actualización de las estadísticas o la comprobación de la integridad de la base de datos) que no sean aplicables o no estén gestionadas por él. AWS
- Cree [scripts](#) y métodos reutilizables para estandarizar el desarrollo y reducir el esfuerzo.
- Pruebe siempre los trabajos migrados en volúmenes de datos e infraestructura que coincidan con los recursos de producción.
- Los paquetes SSIS están en formato XML. Cree utilidades de análisis XML para automatizar las actividades de migración siempre que sea posible. El siguiente ejemplo muestra un paquete SSIS en formato XML.

```
DTS:LastModifiedProductVersion="11.0.2100.60"  
DTS:LocaleID="1033"  
DTS:ObjectName="Package"  
DTS:PackageType="5"  
DTS:VersionBuild="1"  
DTS:VersionGUID="{EC16490E-6783-4BE6-92CA-FDD5BC644F88}">  
<DTS:Property  
  DTS:Name="PackageFormatVersion">6</DTS:Property>  
<DTS:ConnectionManagers>  
  <DTS:ConnectionManager
```

La siguiente ilustración muestra un ejemplo de analizador XML.

```

def getlistofallexecutables(directory):
    """
    Iterate the root directory with all SSIS packages (.dtsx)
    Get the control flow tasks and data flow task components
    """
    try:
        lst=[]
        for fileName in os.listdir(directory):
            if fileName.endswith('.dtsx'):
                cfgfilename=os.path.basename(fileName).split('.')[0]
                tree = etree.parse(directory+fileName)
                root = tree.getroot()
                prefix = '{www.microsoft.com/|
                exetag = prefix + 'SqlServer/Dts}Executable'
                dataflow='Microsoft.Pipeline'
                childtag=prefix+ 'SqlServer/Dts}ExecutableType'
                objdata=prefix+ 'SqlServer/Dts}ObjectData'
                pipeline='pipeline'
                components= 'components'
                componentclassid="componentClassID"
                for cnt, ele in enumerate(root.xpath(".*/*")):
                    if ele.tag==exetag:
                        attr=ele.attrib[childtag]#controlflow
                        flow=cfgfilename+',control_flow,'
                        if attr!=dataflow:
                            lst.append(flow+attr)
                        if attr==dataflow:
                            for child0 in ele:
                                if child0.tag==objdata:
                                    for child1 in child0:
                                        if child1.tag==pipeline:
                                            for child2 in child1:
                                                if child2.tag==components:
                                                    for child3 in child2:
                                                        df_component=child3.attrib[componentclassid]#
                                                        flow=cfgfilename+',data_flow,'
                                                        lst.append(flow+df_component)
                return lst
    except Exception as e:

```

- Utilice sumas de comprobación y valores hash para comprobar la exactitud y la integridad.
- Si estás creando bibliotecas personalizadas, implementa subprocesos para ejecutar tareas en paralelo. Esto mejora el rendimiento laboral.
- Desactive el entorno existente solo después de que todos los puestos de trabajo se hayan mantenido estables AWS durante un período de tiempo.
- Usa Amazon CloudWatch para el registro, a fin de reducir los esfuerzos de personalización del registro.
- Utilice Amazon Simple Notification Service (Amazon SNS) para sustituir las tareas personalizadas en el envío de notificaciones. Utilice Amazon Simple Email Service (Amazon SES) para reemplazar las tareas de correo electrónico personalizadas.

Preguntas frecuentes

En esta sección se proporcionan respuestas a las preguntas más frecuentes sobre la migración de sus trabajos de ETL de SSIS a AWS.

¿Debo mejorar los trabajos de SSIS durante la migración?

Sí. Implemente todas las mejoras o correcciones de errores importantes en los trabajos nuevos y existentes al mismo tiempo. Los cambios de baja prioridad se pueden implementar después de la migración.

¿Cómo puedo supervisar los trabajos? AWS

Puedes usar [Amazon CloudWatch](#) para supervisar los trabajos y enviar alertas basadas en reglas.

¿Cuándo puedo retirar mis trabajos de SSIS locales?

Le recomendamos que mantenga los trabajos antiguos y nuevos en paralelo durante un período de tiempo hasta que el rendimiento, la precisión y la integridad de los datos sean los mismos en ambos entornos. También puede replicar los sistemas de informes para conectar y comparar los informes de ambos entornos. Puede retirar sus trabajos de SSIS cuando los informes sean idénticos.

Pasos a seguir a continuación

Como siguiente paso, le recomendamos que cree una prueba de concepto para determinar su enfoque de implementación para la arquitectura de destino. Para su prueba de concepto:

- Cree AWS Glue trabajos mediante la interfaz gráfica de AWS Glue Studio.
- Utilice AWS Schema Conversion Tool (AWS SCT) para generar AWS Glue scripts a partir de paquetes SSIS.
- Cree un marco de migración personalizado mediante las bibliotecas de Spark.
- Crea un marco de pruebas.
- Identifique las herramientas y los procesos para la integración y el despliegue continuos (CI/CD).

Recursos relacionados

- [Convertir SSIS en uso AWS GlueAWS SCT](#)
- [Respaldo y restauración de Amazon RDS](#)
- [AWS Glue documentación](#)
- [AWS Glue FAQs](#)
- [Registro continuo de AWS Glue trabajos](#)
- [Documentación de Amazon EMR](#)
- [Amazon EMR FAQs](#)
- [Supervise las métricas con CloudWatch](#)
- [Acelere la validación de migración de datos a gran escala mediante PyDeequ](#)
- [AWS Socios competentes en datos y análisis](#)

Historial de documentos

En la siguiente tabla, se describen cambios significativos de esta guía. Si quiere recibir notificaciones de futuras actualizaciones, puede suscribirse a las [notificaciones RSS](#).

Cambio	Descripción	Fecha
Publicación inicial	—	6 de diciembre de 2021

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.