



Migração de trabalhos SSIS ETL para AWS

# AWS Recomendações



# AWS Recomendações: Migração de trabalhos SSIS ETL para AWS

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

As marcas comerciais e imagens de marcas da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre os clientes ou que deprecie ou desprestige a Amazon. Todas as outras marcas comerciais que não pertencem à Amazon pertencem a seus respectivos proprietários, que podem ou não ser afiliados, patrocinados pela Amazon ou ter conexão com ela.

# Table of Contents

|                                                                    |       |
|--------------------------------------------------------------------|-------|
| Introdução .....                                                   | 1     |
| Resultados de negócios desejados .....                             | 1     |
| Fases de migração .....                                            | 2     |
| Fase de descoberta .....                                           | 2     |
| Fase de análise .....                                              | 4     |
| Determinando a abordagem de migração .....                         | 5     |
| Fase de implementação .....                                        | 7     |
| AWS SCT .....                                                      | 8     |
| AWS Glue .....                                                     | 8     |
| Amazon EMR .....                                                   | 10    |
| Teste .....                                                        | 10    |
| Práticas recomendadas .....                                        | 12    |
| Perguntas frequentes .....                                         | 14    |
| Devo melhorar os trabalhos do SSIS durante a migração? .....       | 14    |
| Como posso monitorar os trabalhos em andamento AWS? .....          | 14    |
| Quando posso descomissionar minhas tarefas do SSIS no local? ..... | 14    |
| Próximas etapas .....                                              | 15    |
| Recursos relacionados .....                                        | 16    |
| Histórico do documento .....                                       | 17    |
| .....                                                              | xviii |

# Migrando trabalhos SSIS ETL para AWS

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

Dezembro de 2021 ([histórico do documento](#))

A Amazon Web Services (AWS) fornece serviços para desenvolver e executar trabalhos de extração, transformação e carga (ETL) ou de extração, carga e transformação (ELT) e cargas de trabalho de big data. O Microsoft SQL Server Integration Services (SSIS) é uma ferramenta de ETL local que tem uma interface gráfica para criar trabalhos de ETL. Dependendo da arquitetura do estado de destino, você pode usar a interface AWS Glue gráfica ou a estrutura Apache Spark com bibliotecas Python para criar trabalhos de ETL na nuvem. AWS

Este guia descreve as etapas para migrar trabalhos SSIS ETL/ELT do local para o AWS. Ele discute as fases da migração, as melhores práticas e as recomendações para reduzir o esforço de migração e melhorar a experiência. As informações são aplicáveis a qualquer AWS arquitetura de destino.

O guia é para gerentes de programas ou projetos, proprietários de produtos, arquitetos de soluções e desenvolvedores que são:

- Migração do SSIS para AWS por vários motivos, como modernização ou economia de custos.
- Usando principalmente AWS serviços para cargas de trabalho de ETL e big data.
- Procurando uma solução baseada em nuvem para aproveitar as vantagens de um modelo sem servidor e preços flexíveis.

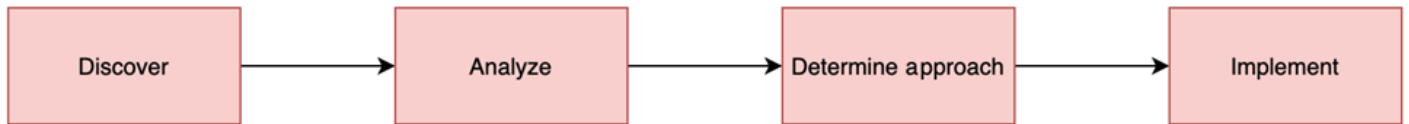
## Resultados de negócios desejados

A migração de suas tarefas do SSIS para a AWS nuvem ajuda você a alcançar esses resultados principais:

- Modernize os processos de ETL locais existentes de forma econômica
- Responda às necessidades de informação da organização com mais rapidez
- Combine processos existentes e elimine processos não utilizados para reduzir a sobrecarga operacional e de manutenção
- Crie uma base para atender aos requisitos futuros de dados da sua organização

# Fases de migração

A migração dos processos de ETL do SSIS para a AWS nuvem consiste em quatro fases principais: descobrir, analisar, determinar a abordagem e implementar.



Essas fases são discutidas nas seções a seguir:

- [Fase de descoberta](#)
- [Fase de análise](#)
- [Determinando a abordagem de migração](#)
- [Fase de implementação](#)

## Fase de descoberta

Na fase de descoberta, você cria uma lista dos pacotes SSIS para os quais deseja migrar. AWS Diferentes equipes de desenvolvimento seguem diferentes estilos, padrões e padrões para desenvolver trabalhos de ETL. Recomendamos que você revise os documentos existentes da sua organização para entender esses padrões. No entanto, a documentação geralmente está incompleta. Você pode automatizar a extração de informações importantes dos scripts ETL. Isso economiza tempo e esforço manual, reduz os erros humanos e padroniza a abordagem de migração. Aqui estão alguns dos detalhes importantes que você desejará extrair:

- Número total de tarefas de controle de fluxo
- Detalhes das tarefas de controle de fluxo
- Número total de tarefas de fluxo de dados
- Transformações de fluxo de dados usadas
- Manipuladores de eventos
- Gerenciadores de conexão

Use essas informações para entender os padrões de ETL usados em sua organização, avaliar sua complexidade e identificar o AWS serviço apropriado para o qual migrar essas informações.

A migração desses detalhes de ETL do SSIS constitui a maior parte do esforço de migração. No entanto, propriedades adicionais podem fornecer informações sobre decisões de design e arquitetura. Algumas dessas propriedades do SSIS são:

- [Pontos de verificação](#), que são usados no SSIS para reiniciar trabalhos a partir de pontos de falha
- [Propague variáveis](#), que ajudam um pacote SSIS a ter sucesso em casos de uso específicos, mesmo quando há um erro
- [Níveis de isolamento de transações](#), que controlam a qualidade dos dados lidos nos bancos de dados
- [Registro](#), para entender os tipos de registros que estão sendo capturados pelo design atual e seus locais de armazenamento

O resultado da fase de descoberta pode ser um inventário, como mostra a tabela a seguir.

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                          |

Esse inventário pode incluir as seguintes informações:

- **Package:** Nome do pacote SSIS a ser migrado
- **Fluxo:** [controle o fluxo](#) ou o [fluxo de dados](#)
- **Tarefa:** Nome da tarefa do fluxo de controle ou do componente do fluxo de dados
- **Contagem:** Número de vezes que uma tarefa foi usada no pacote SSIS

## Fase de análise

Analisar as informações da fase de descoberta ajuda você a entender os padrões e a projetar melhor a arquitetura de destino. Siga estas dicas para melhorar o resultado da análise:

- A opção de registro no [nível do pacote](#) registra eventos e captura informações de tempo de execução de cada componente. Use o registro personalizado para que você possa configurar arquivos de log para incluir informações adicionais, como IDs de execução e outros valores de tempo de execução.
- Redirecione registros incorretos para um local de armazenamento diferente para análise, correção e reprocessamento.
- Entenda as estratégias de arquivamento e limpeza de dados implementadas em seu ambiente SSIS local.
- Envie alertas e notificações usando tarefas incluídas no SSIS (como a [tarefa Enviar e-mail](#)) ou scripts personalizados (como a [tarefa Script](#)).
- Entenda o comportamento das [transformações](#) no escopo para evitar dados corrompidos. Por exemplo, a [transformação Lookup](#) é uma junção equivalente entre dois conjuntos de dados e sua comparação diferencia maiúsculas de minúsculas.

Você pode mapear cada entrada no inventário para uma complexidade estimada, seja em termos de duração (minutos ou horas) ou nível (simples, médio ou complexo). Esse inventário expandido, mostrado na tabela a seguir, é um resultado da fase de análise.

## 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          |

## Determinando a abordagem de migração

Para decidir sobre uma abordagem de migração, você usa a análise realizada nos padrões existentes na fase anterior. As necessidades futuras de dados e análises de sua organização são considerações igualmente importantes. As ferramentas tradicionais de ETL locais lidam com modelos de dados relacionais e dados estruturados. Se você tiver dados semiestruturados e não estruturados para processar, poderá usar AWS serviços como o AWS Glue Amazon EMR para a migração. Outros fatores que podem influenciar a abordagem de migração incluem:

- Se você quiser usar uma interface gráfica (como o [AWS Glue Studio](#)) ou uma estrutura personalizada (como bibliotecas Spark/Python)
- Se você tem acesso seguro a fontes e AWS destinos locais
- Habilidades e treinamento necessários para a equipe
- Requisitos de auditoria e conformidade

Você pode selecionar entre três abordagens de migração: big bang, phased e lift and shift. A tabela a seguir compara essas três abordagens.

| Abordagem      | Descrição                                                                                                                                            | Caso de uso                                                                                                                                                                                                 | Vantagens e desvantagens                                                                                                                                                                                                                       |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Big bang       | Migre todos os pacotes SSIS dentro de um período específico.                                                                                         | <ul style="list-style-type: none"> <li>A complexidade, o escopo e a arquitetura de destino são claros.</li> <li>A equipe tem as habilidades necessárias ou a curva de aprendizado é superficial.</li> </ul> | <ul style="list-style-type: none"> <li>Alto risco.</li> <li>Leva menos tempo do que a abordagem em fases.</li> <li>Você pode usar AWS Glue o Amazon EMR ou estruturas personalizadas.</li> </ul>                                               |
| Faseado        | Identifique um pacote SSIS para cada padrão e complexidade distintos. Migre o pacote AWS, teste e compare os resultados com a arquitetura existente. | <ul style="list-style-type: none"> <li>O tempo não é uma restrição.</li> <li>Você quer designs diferentes para diferentes padrões de ETL.</li> </ul>                                                        | <ul style="list-style-type: none"> <li>Menos arriscada do que a abordagem do big bang, mas exige mais tempo e esforço.</li> <li>Você pode usar AWS Glue o Amazon EMR ou estruturas personalizadas.</li> </ul>                                  |
| Levante e mude | Migre a arquitetura atual como está. AWS                                                                                                             | <ul style="list-style-type: none"> <li>Seu hardware local não é mais suportado.</li> <li>Você não tem os recursos para planejar uma migração imediatamente.</li> </ul>                                      | <ul style="list-style-type: none"> <li>Menor esforço e tempo de migração necessários.</li> <li>Os problemas com a solução existente permanecem AWS.</li> <li>Os pacotes SSIS são executados como estão. Nenhuma outra ferramenta ou</li> </ul> |

| Abordagem | Descrição | Caso de uso | Vantagens e desvantagens |
|-----------|-----------|-------------|--------------------------|
|-----------|-----------|-------------|--------------------------|

estrutura de ETL é necessária.

A comparação dos dados nos sistemas de origem e de destino é fundamental para uma migração bem-sucedida. Como o sistema de produção existente recebe atualizações regulares dos sistemas de origem, essa comparação pode se tornar confusa. Por esse motivo, ao determinar sua abordagem de migração, recomendamos que você também decida sobre sua estratégia de validação de dados.

- Faça backups de todos os bancos de dados e arquivos aplicáveis do ambiente de produção no sistema de origem em uma data e hora específicas.
- Faça backups de todos os bancos de dados do ambiente de produção no sistema de destino depois que todas as tarefas tiverem carregado com sucesso os dados dos dados de origem do backup.
- Restaure os dados de origem em um ambiente de teste e execute os novos trabalhos.
- Concorde com uma porcentagem de diferenças válidas entre os bancos de dados de origem e de destino (antigos e novos). Por exemplo, você pode decidir que uma diferença de menos de 1% é aceitável.
- Liste todas as regras de validação a serem abordadas.
- Automatize a comparação o máximo possível e cubra todas as regras.

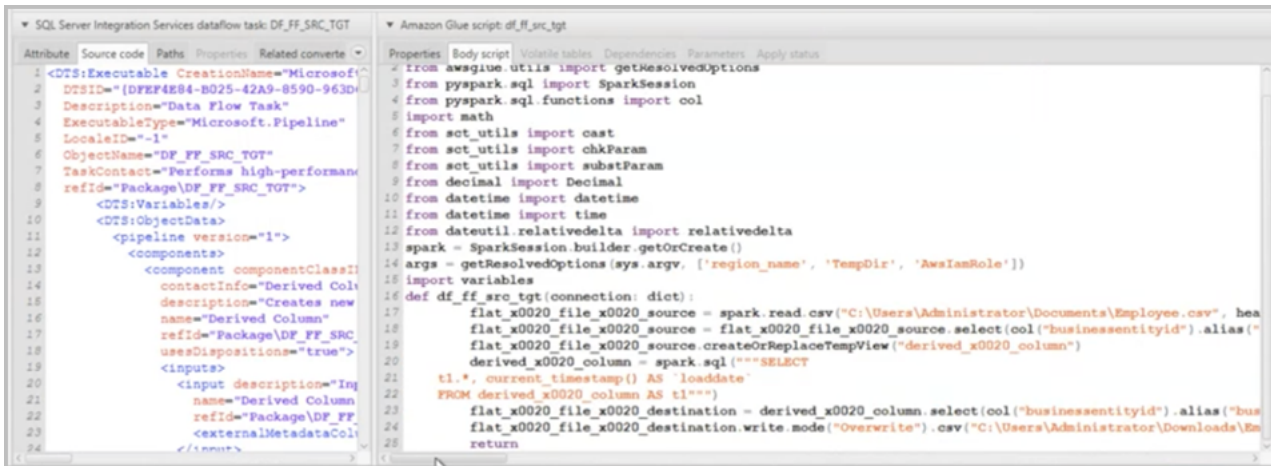
## Fase de implementação

A migração que segue o big bang ou a abordagem em fases requer novos desenvolvimentos e testes. O AWS Schema Conversion Tool (AWS SCT) pode gerar automaticamente AWS Glue trabalhos a partir de pacotes SSIS. Isso reduz significativamente o tempo e o esforço de migração. Ou você pode usar o AWS Glue Studio para desenvolvimento baseado em interface gráfica ou criar bibliotecas Spark que podem ser executadas em qualquer um deles ou no Amazon AWS Glue EMR.

As seções a seguir fornecem dicas úteis para usar AWS SCT AWS Glue, e o Amazon EMR.

## AWS SCT

A ilustração de tela a seguir mostra um script de AWS Glue trabalho que foi convertido por AWS SCT.

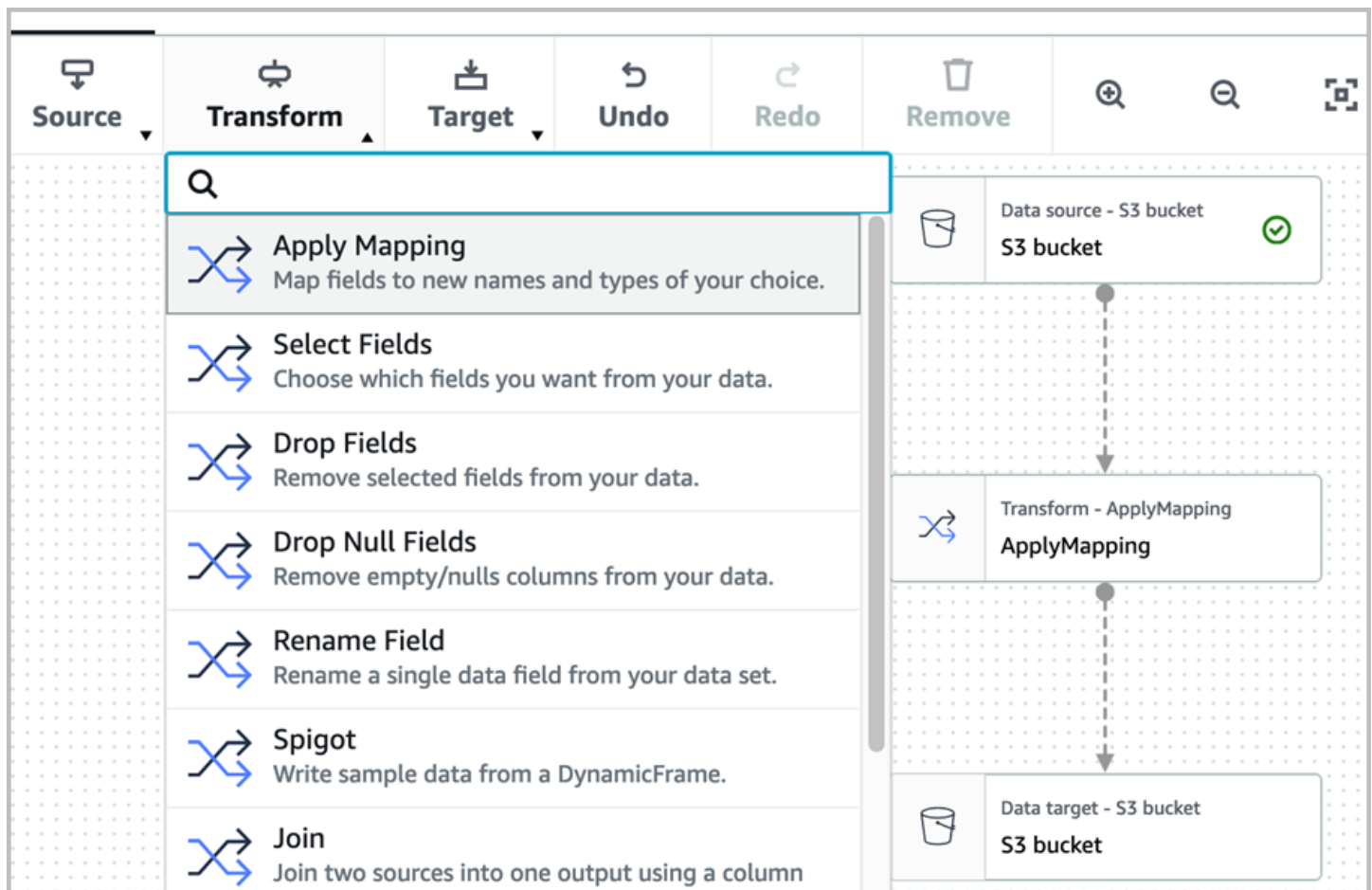


AWS SCT pode converter pacotes SSIS em AWS Glue trabalhos em massa. Você pode editar o script para atualizar a lógica existente ou adicionar uma nova lógica, com base no seu novo design. Recomendamos que você siga as convenções de nomenclatura nos scripts AWS SCT convertidos para personalizá-los.

Para obter mais informações, consulte [Conversão do SSIS AWS SCT em AWS Glue uso](#) na AWS SCT documentação.

## AWS Glue

AWS Glue O Studio fornece uma interface gráfica e uma experiência de desenvolvimento semelhantes às do SSIS, conforme ilustrado na tela a seguir.



Se você preferir não usar uma interface gráfica, você também pode executar seus scripts personalizados com as bibliotecas Python necessárias no AWS Glue console. Para obter mais informações, consulte Como [fornecer seus próprios scripts personalizados](#) na AWS Glue documentação.

AWS Glue fornece um conjunto de transformações integradas para processar seus dados. Elas são semelhantes às transformações de fluxo de dados do SSIS. Siga estas melhores práticas ao migrar suas tarefas de ETL do SSIS usando: AWS Glue

- Prepare um mapeamento das AWS Glue transformações para as transformações SSIS equivalentes.
- Se suas transformações não puderem ser mapeadas para AWS Glue transformações, crie-as usando um script personalizado em Python ou Scala.
- Para registros personalizados (como linhas lidas, linhas gravadas ou registros incorretos), use scripts personalizados além [da Amazon CloudWatch](#).

- Adicione um [endpoint de desenvolvimento](#) para desenvolver e depurar scripts personalizados localmente.

## Amazon EMR

Você pode executar scripts personalizados (escritos em Python ou Scala) ou bibliotecas Python compiladas em clusters do EMR, como acontece com. AWS Glue Siga estas práticas recomendadas:

- Comece com tipos de instância otimizados para memória enquanto cria clusters EMR com a estrutura Spark. (O SSIS usa buffers de memória.)
- Crie métodos Python genéricos que sejam equivalentes a cada tarefa ou transformação do SSIS. Por exemplo, na ilustração a seguir, um método que usa dois quadros de dados como entrada produz um terceiro quadro de dados que tem registros correspondentes dos dois quadros de dados como saída. Isso funciona como uma transformação de [junção de mesclagem](#).

```
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
```

## Teste

É necessária uma estrutura de teste para validar a integridade e a exatidão dos dados. Essa estrutura deve abranger todos os cenários existentes e todas as melhorias que você fez ao migrar seus trabalhos para o AWS

- Validação de integridade:
  - Todos os trabalhos são migrados para o estado de destino.
  - Todas as funcionalidades são migradas em cada trabalho.
  - Todos os tipos de registros estão disponíveis, incluindo detalhes da execução do trabalho, mensagens de erro, registros incorretos e contagens de linhas.
- Validação de exatidão:
  - A qualidade dos dados é consistente nos ambientes existentes e novos.
  - Todas as colunas de todas as tabelas coincidem ou as tabelas foram aprimoradas AWS.
  - Todas as informações de auditoria e registro coincidem.

Você também deve verificar se o desempenho dos trabalhos migrados corresponde ao desempenho dos trabalhos existentes.

# Práticas recomendadas

Siga estas melhores práticas gerais ao migrar suas tarefas do SSIS para: AWS

- Se você estiver migrando seus bancos de dados para um serviço AWS gerenciado, como o Amazon Relational Database Service (Amazon RDS), resolva tarefas [de manutenção](#) (como backup, restauração, atualização de estatísticas ou verificação da integridade do banco de dados) que não são aplicáveis ou gerenciadas pela AWS.
- Crie [scripts](#) e métodos reutilizáveis para padronizar o desenvolvimento e reduzir o esforço.
- Sempre teste os trabalhos migrados em volumes de infraestrutura e dados que correspondam aos recursos de produção.
- Os pacotes SSIS estão no formato XML. Crie utilitários de análise de XML para automatizar as atividades de migração sempre que possível. O exemplo a seguir mostra um pacote SSIS no 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
```

A ilustração a seguir mostra um exemplo de analisador 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:

```

- Use somas de verificação e valores de hash para testar a exatidão e a integridade.
- Se você estiver criando bibliotecas personalizadas, implemente o threading para executar tarefas em paralelo. Isso melhora o desempenho no trabalho.
- Desative o ambiente existente somente depois que todos os trabalhos estiverem estáveis AWS por um período de tempo.
- Use a Amazon CloudWatch para registrar, para reduzir os esforços de personalização do registro.
- Use o Amazon Simple Notification Service (Amazon SNS) para substituir tarefas personalizadas de envio de notificações. Use o Amazon Simple Email Service (Amazon SES) para substituir tarefas personalizadas de e-mail.

## Perguntas frequentes

Esta seção fornece respostas às perguntas mais comuns sobre a migração de suas tarefas de ETL do SSIS para o AWS.

### Devo melhorar os trabalhos do SSIS durante a migração?

Sim. Implemente todos os aprimoramentos ou correções de erros importantes em trabalhos existentes e novos ao mesmo tempo. Mudanças de baixa prioridade podem ser implementadas após a migração.

### Como posso monitorar os trabalhos em andamento AWS?

Você pode usar CloudWatch a [Amazon](#) para monitorar trabalhos e enviar alertas baseados em regras.

### Quando posso descomissionar minhas tarefas do SSIS no local?

Recomendamos que você mantenha trabalhos antigos e novos em paralelo por um período de tempo até que o desempenho, a precisão e a integridade dos dados sejam os mesmos nos dois ambientes. Você também pode replicar sistemas de relatórios para conectar e comparar relatórios de ambos os ambientes. Você pode descomissionar seus trabalhos do SSIS quando os relatórios forem idênticos.

## Próximas etapas

Na próxima etapa, recomendamos que você crie uma prova de conceito para determinar sua abordagem de implantação para a arquitetura de destino. Para sua prova de conceito:

- Crie AWS Glue trabalhos usando a interface gráfica do AWS Glue Studio.
- Use o AWS Schema Conversion Tool (AWS SCT) para gerar AWS Glue scripts a partir de pacotes SSIS.
- Crie uma estrutura de migração personalizada usando as bibliotecas do Spark.
- Crie uma estrutura de testes.
- Identifique ferramentas e processos para integração e implantação contínuas (CI/CD).

## Recursos relacionados

- [Convertendo SSIS em uso AWS GlueAWS SCT](#)
- [Backup e restauração do Amazon RDS](#)
- [AWS Glue documentação](#)
- [AWS Glue FAQs](#)
- [Registro contínuo para AWS Glue trabalhos](#)
- [Documentação do Amazon EMR](#)
- [Amazon EMR FAQs](#)
- [Monitore métricas com CloudWatch](#)
- [Acelere a validação da migração de dados em grande escala usando PyDeequ](#)
- [AWS Parceiros de competência em dados e análises](#)

## Histórico do documento

A tabela a seguir descreve alterações significativas feitas neste guia. Se desejar receber notificações sobre futuras atualizações, inscreva-se em um [feed RSS](#).

| Alteração                          | Descrição | Data                  |
|------------------------------------|-----------|-----------------------|
| <a href="#">Publicação inicial</a> | —         | 6 de dezembro de 2021 |

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.