



Migration des tâches SSIS ETL vers AWS

AWS Conseils prescriptifs



AWS Conseils prescriptifs: Migration des tâches SSIS ETL vers AWS

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

Les marques et la présentation commerciale d'Amazon ne peuvent être utilisées en relation avec un produit ou un service qui n'est pas d'Amazon, d'une manière susceptible de créer une confusion parmi les clients, ou d'une manière qui dénigre ou discrédite Amazon. Toutes les autres marques commerciales qui ne sont pas la propriété d'Amazon appartiennent à leurs propriétaires respectifs, qui peuvent ou non être affiliés ou connectés à Amazon, ou sponsorisés par Amazon.

Table of Contents

Introduction	1
Résultats commerciaux ciblés	1
Phases de migration	3
Phase de découverte	3
Phase d'analyse	5
Déterminer l'approche de migration	6
Phase de mise en œuvre	8
AWS SCT	9
AWS Glue	9
Amazon EMR	11
Test	11
Bonnes pratiques	13
FAQ	15
Dois-je améliorer les tâches SSIS pendant la migration ?	15
Comment puis-je suivre les offres d'emploi sur AWS ?	15
Quand puis-je mettre hors service mes tâches SSIS sur site ?	15
Étapes suivantes	16
Ressources connexes	17
Historique du document	18
.....	xix

Migration des tâches SSIS ETL vers AWS

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

Décembre 2021 ([historique du document](#))

Amazon Web Services (AWS) fournit des services pour le développement et l'exécution de tâches d'extraction, de transformation et de chargement (ETL) ou d'extraction, de chargement et de transformation (ELT) et de charges de travail Big Data. Microsoft SQL Server Integration Services (SSIS) est un outil ETL sur site doté d'une interface graphique permettant de créer des tâches ETL. En fonction de l'architecture de l'état cible, vous pouvez utiliser l'interface AWS Glue graphique ou le framework Apache Spark avec des bibliothèques Python pour créer des tâches ETL dans le AWS cloud.

Ce guide décrit les étapes de migration des tâches SSIS ETL/ELT d'une installation sur site vers AWS. Il décrit les phases de migration, les meilleures pratiques et les recommandations visant à réduire les efforts de migration et à améliorer l'expérience. Les informations sont applicables à n'importe quelle AWS architecture cible.

Ce guide s'adresse aux chefs de programme ou de projet, aux propriétaires de produits, aux architectes de solutions et aux développeurs qui sont :

- Migration de SSIS vers SSIS AWS pour diverses raisons, telles que la modernisation ou la réduction des coûts.
- Nous utilisons principalement AWS des services pour les charges de travail ETL et Big Data.
- Vous recherchez une solution basée sur le cloud pour tirer parti d'un modèle sans serveur et d'une tarification flexible.

Résultats commerciaux ciblés

La migration de vos tâches SSIS vers le AWS cloud vous permet d'atteindre les principaux résultats suivants :

- Modernisez les processus ETL sur site existants de manière rentable
- Répondez plus rapidement aux besoins d'information de l'organisation
- Fusionnez les processus existants et supprimez les processus inutilisés afin de réduire les frais de maintenance et d'exploitation

- Établissez les bases nécessaires pour répondre aux futures exigences de votre organisation en matière de données

Phases de migration

La migration des processus SSIS ETL vers le AWS cloud comprend quatre phases principales : découverte, analyse, détermination de l'approche et mise en œuvre.



Ces phases sont abordées dans les sections suivantes :

- [Phase de découverte](#)
- [Phase d'analyse](#)
- [Déterminer l'approche de migration](#)
- [Phase de mise en œuvre](#)

Phase de découverte

Au cours de la phase de découverte, vous créez une liste des packages SSIS vers lesquels vous souhaitez migrer AWS. Les différentes équipes de développement suivent différents styles, normes et modèles pour développer des tâches ETL. Nous vous recommandons de consulter les documents existants de votre organisation pour comprendre ces modèles. Cependant, la documentation est souvent incomplète. Vous pouvez automatiser l'extraction d'informations importantes à partir des scripts ETL. Cela permet d'économiser du temps et des efforts manuels, de réduire les erreurs humaines et de standardiser l'approche de migration. Voici certains des détails importants que vous souhaitez extraire :

- Nombre total de tâches de flux de contrôle
- Détails des tâches du flux de contrôle
- Nombre total de tâches de flux de données
- Transformations de flux de données utilisées
- Gestionnaires d'événements
- Gestionnaires de connexions

Utilisez ces informations pour comprendre les modèles ETL utilisés dans votre organisation, pour évaluer leur complexité et pour identifier le AWS service approprié vers lequel migrer ces informations.

La migration de ces informations ETL depuis SSIS constitue l'essentiel de l'effort de migration. Cependant, des propriétés supplémentaires peuvent fournir des informations sur les décisions de conception et d'architecture. Certaines de ces propriétés SSIS sont les suivantes :

- [Points de contrôle](#), utilisés dans SSIS pour redémarrer les tâches à partir de points de défaillance
- [Propagation de variables](#), qui aident un package SSIS à réussir dans des cas d'utilisation spécifiques, même en cas d'erreur
- [Niveaux d'isolation des transactions](#), qui contrôlent la qualité des données lues à partir des bases de données
- [Journalisation](#), pour comprendre les types de journaux capturés par la conception actuelle et leurs emplacements de stockage

Le résultat de la phase de découverte peut être un inventaire, comme le montre le tableau suivant.

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

Cet inventaire peut inclure les informations suivantes :

- Package : nom du package SSIS à migrer
- Flux : flux [de contrôle ou flux de données](#)
- Tâche : nom de la tâche du flux de contrôle ou du composant du flux de données
- Nombre : nombre de fois qu'une tâche a été utilisée dans le package SSIS

Phase d'analyse

L'analyse des informations issues de la phase de découverte vous aide à comprendre les modèles et à mieux concevoir l'architecture cible. Suivez ces conseils pour améliorer les résultats de l'analyse :

- L'option de journalisation au [niveau du package](#) enregistre les événements et capture les informations d'exécution pour chaque composant. Utilisez la journalisation personnalisée pour configurer les fichiers journaux afin d'inclure des informations supplémentaires, telles que les identifiants d'exécution et autres valeurs d'exécution.
- Redirigez les mauvais enregistrements vers un autre emplacement de stockage à des fins d'analyse, de correction et de retraitement.
- Découvrez les stratégies d'archivage et de purge des données mises en œuvre dans votre environnement SSIS sur site.
- Envoyez des alertes et des notifications à l'aide de tâches incluses dans SSIS (comme la [tâche Envoyer un e-mail](#)) ou de scripts personnalisés (tels que la [tâche Script](#)).
- Comprenez le comportement des [transformations](#) dans Scope afin d'éviter de corrompre les données. Par exemple, la [transformation Lookup](#) est une équijointure entre deux ensembles de données, et sa comparaison fait la distinction majuscules/majuscules.

Vous pouvez associer chaque entrée de l'inventaire à une complexité estimée, en termes de durée (minutes ou heures) ou de niveau (simple, moyen ou complexe). Cet inventaire élargi, présenté dans le tableau suivant, est le résultat de la phase d'analyse.

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

Déterminer l'approche de migration

Pour choisir une approche de migration, vous devez utiliser l'analyse que vous avez effectuée sur les modèles existants lors de la phase précédente. Les besoins futurs de votre entreprise en matière de données et d'analyses sont tout aussi importants. Les outils ETL traditionnels sur site traitent des modèles de données relationnels et des données structurées. Si vous avez des données semi-structurées et non structurées à traiter, vous pouvez utiliser AWS des services tels qu' AWS Glue Amazon EMR pour la migration. Parmi les autres facteurs susceptibles d'influencer l'approche migratoire, citons :

- Que vous souhaitiez utiliser une interface graphique (comme [AWS Glue Studio](#)) ou un framework personnalisé (comme les bibliothèques Spark/Python)
- Si vous disposez d'un accès sécurisé aux sources et AWS cibles sur site
- Compétences et formation requises pour l'équipe
- Exigences en matière d'audit et de conformité

Vous pouvez choisir entre trois approches de migration : big bang, phasée et lift and shift. Le tableau suivant compare ces trois approches.

Approche	Description	Cas d'utilisation	Avantages et inconvénients
Big Bang	Migrez tous les packages SSIS au cours d'une période donnée.	- La complexité, la portée et l'architecture cible sont claires. - L'équipe possède les compétences requises ou la courbe d'apprentissage est superficielle.	- Risque élevé. - Cela prend moins de temps que l'approche progressive. - Vous pouvez utiliser AWS Glue Amazon EMR ou des frameworks personnalisés.
Progressif	Identifiez un package SSIS pour chaque modèle et chaque complexité distincts. Migrez le package vers l'architecture existante AWS, testez-le et comparez les résultats avec celle-ci.	- Le temps n'est pas une contrainte. - Vous souhaitez différents modèles pour différents modèles ETL.	- Moins risquée que l'approche du big bang, mais elle demande plus de temps et d'efforts. - Vous pouvez utiliser AWS Glue Amazon EMR ou des frameworks personnalisés.
Soulever et changer de vitesse	Migrez l'architecture actuelle telle quelle vers AWS.	- Votre matériel sur site n'est plus pris en charge. - Vous ne disposez pas des ressources nécessaires pour planifier une migration immédiatement.	- Minimiser les efforts et le temps nécessaires à la migration. - Les problèmes liés à la solution existante persistent AWS. - Les packages SSIS sont exécutés tels quels. Aucun autre

Approche	Description	Cas d'utilisation	Avantages et inconvénients
			outil ou framework ETL n'est nécessaire.

La comparaison des données des systèmes source et cible est fondamentale pour une migration réussie. Le système de production existant étant régulièrement mis à jour par les systèmes sources, cette comparaison peut prêter à confusion. C'est pourquoi, lorsque vous déterminez votre approche de migration, nous vous recommandons de définir également votre stratégie de validation des données.

- Effectuez des sauvegardes de toutes les bases de données et de tous les fichiers applicables depuis l'environnement de production sur le système source à une date et à une heure spécifiques.
- Effectuez des sauvegardes de toutes les bases de données de l'environnement de production sur le système cible une fois que toutes les tâches ont correctement chargé les données à partir des données sources sauvegardées.
- Restaurez les données sources dans un environnement de test et exécutez les nouvelles tâches.
- Convenez d'un pourcentage de différences valides entre les bases de données source et cible (ancienne et nouvelle). Par exemple, vous pouvez décider qu'une différence inférieure à 1 % est acceptable.
- Répertoriez toutes les règles de validation à couvrir.
- Automatisez la comparaison autant que possible et couvrez toutes les règles.

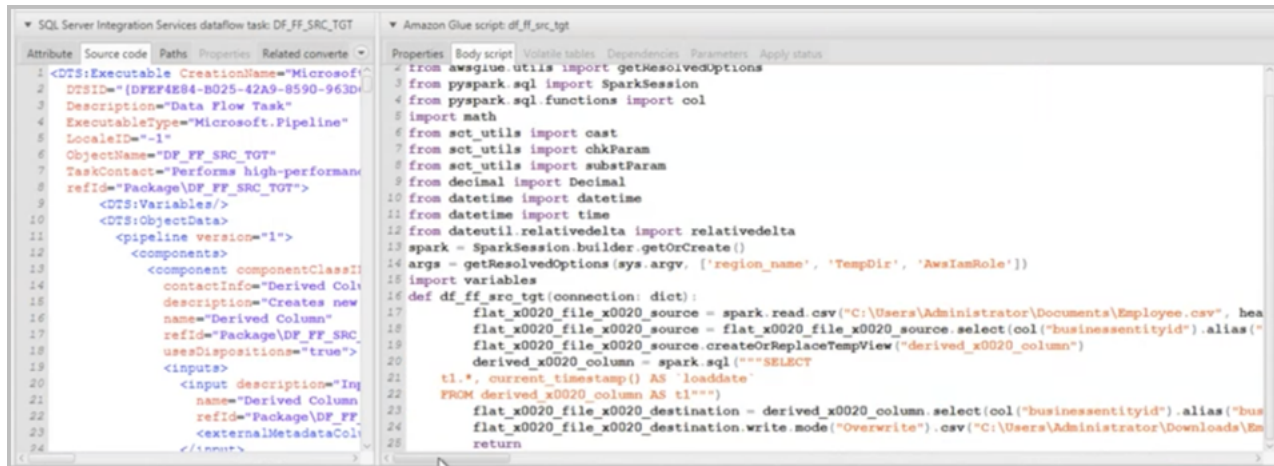
Phase de mise en œuvre

La migration qui suit le Big Bang ou une approche progressive nécessite de nouveaux développements et tests. Le AWS Schema Conversion Tool (AWS SCT) peut générer automatiquement des AWS Glue tâches à partir de packages SSIS. Cela réduit considérablement le temps et les efforts de migration. Vous pouvez également utiliser AWS Glue Studio pour le développement basé sur une interface graphique, ou créer des bibliothèques Spark que vous pouvez exécuter sur Amazon EMR ou sur AWS Glue Amazon EMR.

Les sections suivantes fournissent des conseils utiles pour utiliser AWS SCT Amazon EMR et Amazon EMR. AWS Glue

AWS SCT

L'illustration d'écran suivante montre un script de AWS Glue tâche converti par AWS SCT.

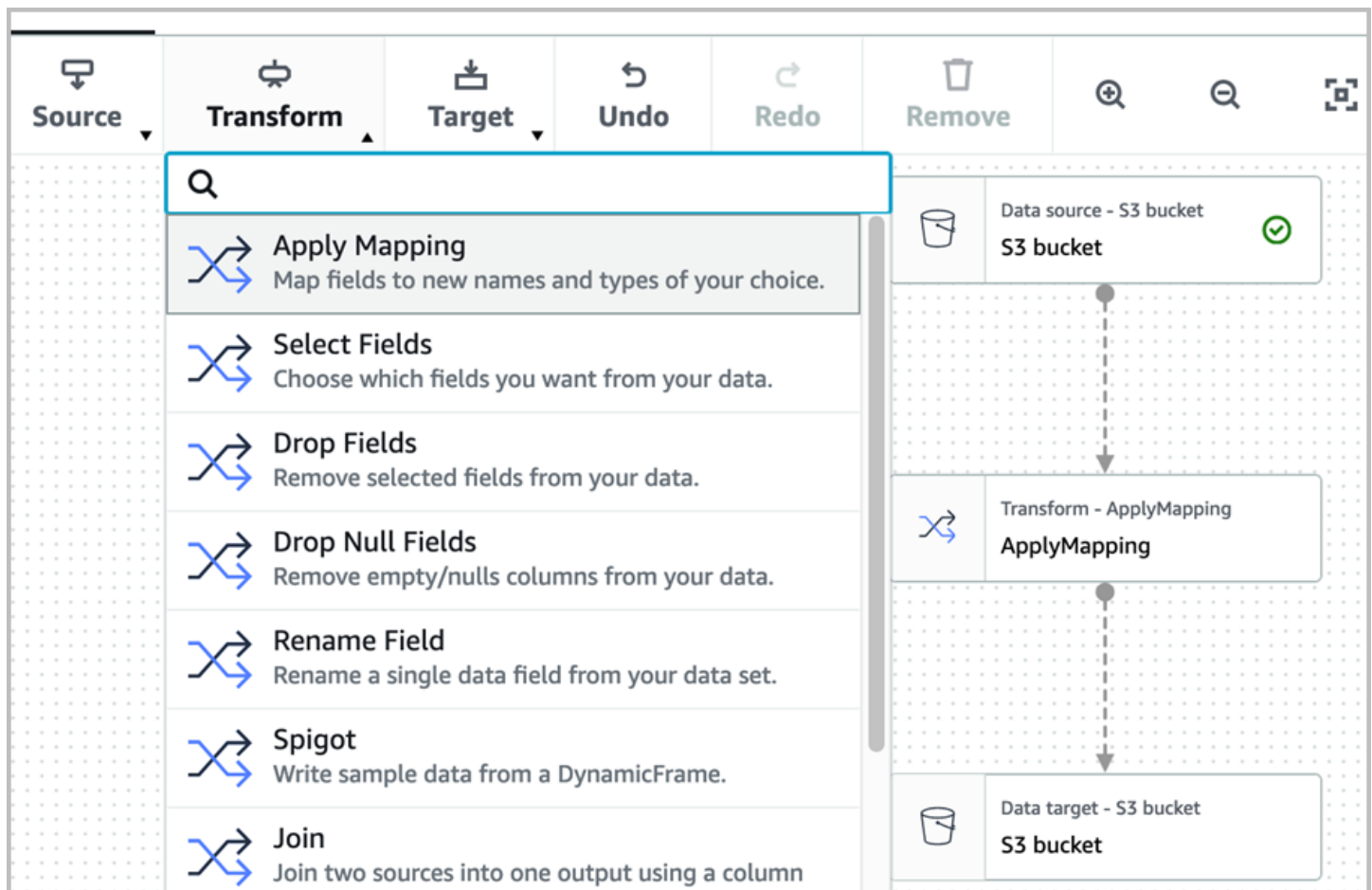


AWS SCT peut convertir des packages SSIS en AWS Glue tâches en masse. Vous pouvez modifier le script pour mettre à jour la logique existante ou pour ajouter une nouvelle logique, en fonction de votre nouveau design. Nous vous recommandons de suivre les conventions de dénomination des scripts AWS SCT convertis pour personnaliser les scripts.

Pour plus d'informations, consultez la section [Conversion de SSIS AWS SCT en mode AWS Glue utilisation](#) dans la AWS SCT documentation.

AWS Glue

AWS Glue Studio fournit une interface graphique et une expérience de développement similaires à SSIS, comme illustré dans l'écran suivant.



Si vous préférez ne pas utiliser d'interface graphique, vous pouvez également exécuter vos scripts personnalisés avec les bibliothèques Python requises depuis la AWS Glue console. Pour plus d'informations, consultez la section [Fournir vos propres scripts personnalisés](#) dans la AWS Glue documentation.

AWS Glue fournit un ensemble de transformations intégrées pour le traitement de vos données. Elles sont similaires aux transformations de flux de données SSIS. Suivez ces bonnes pratiques lorsque vous migrez vos tâches SSIS ETL en utilisant AWS Glue :

- Préparez un mappage entre les AWS Glue transformations et les transformations SSIS équivalentes.
- Si vos transformations ne peuvent pas être mappées à des AWS Glue transformations, créez-les à l'aide d'un script personnalisé Python ou Scala.
- Pour la journalisation personnalisée (lignes lues, lignes écrites ou enregistrements incorrects, par exemple), utilisez des scripts personnalisés en plus d'[Amazon CloudWatch](#).

- Ajoutez un [point de terminaison de développement](#) pour développer et déboguer des scripts personnalisés localement.

Amazon EMR

Vous pouvez exécuter des scripts personnalisés (écrits en Python ou Scala) ou des bibliothèques Python compilées dans des clusters EMR, comme avec. AWS Glue Suivez ces bonnes pratiques :

- Commencez par des types d'instances optimisés pour la mémoire lors de la création de clusters EMR avec le framework Spark. (SSIS utilise des tampons de mémoire.)
- Créez des méthodes Python génériques équivalentes à chaque tâche ou transformation SSIS. Par exemple, dans l'illustration suivante, une méthode qui prend deux dataframes en entrée produit une troisième dont les enregistrements correspondants des deux dataframes sont en sortie. Cela fonctionne comme une transformation par [fusion et jointure](#).

```

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

```

Test

Un cadre de test est nécessaire pour valider l'exhaustivité et l'exactitude des données. Ce cadre doit couvrir tous les scénarios existants et toutes les améliorations que vous avez apportées lors de la migration de vos tâches. AWS

- Validation d'exhaustivité :
 - Toutes les tâches sont migrées vers leur état cible.
 - Toutes les fonctionnalités sont migrées dans chaque tâche.
 - Tous les types de journaux sont disponibles, y compris les détails de l'exécution des tâches, les messages d'erreur, les mauvais enregistrements et le nombre de lignes.
- Validation de l'exactitude :
 - La qualité des données est constante dans les environnements existants et nouveaux.
 - Toutes les colonnes de tous les tableaux correspondent, ou les tableaux sont améliorés AWS.
 - Toutes les informations d'audit et de journalisation correspondent.

Vous devez également vérifier que les performances de vos tâches migrées correspondent à celles de vos tâches existantes.

Bonnes pratiques

Suivez ces bonnes pratiques générales lors de la migration de vos tâches SSIS vers AWS :

- Si vous migrez vos bases de données vers un service AWS géré tel qu'Amazon Relational Database Service (Amazon RDS), [délimitez les tâches de maintenance](#) (telles que la sauvegarde, la restauration, la mise à jour des statistiques ou la vérification de l'intégrité de la base de données) qui ne sont pas applicables ou gérées par AWS.
- Créez des [scripts et des](#) méthodes réutilisables pour standardiser le développement et réduire les efforts.
- Testez toujours les tâches migrées sur l'infrastructure et les volumes de données correspondant aux ressources de production.
- Les packages SSIS sont au format XML. Créez des utilitaires d'analyse XML pour automatiser les activités de migration dans la mesure du possible. L'exemple suivant montre un package SSIS au format 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
```

L'illustration suivante montre un exemple d'analyseur 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:

```

- Utilisez des checksums et des valeurs de hachage pour tester l'exactitude et l'exhaustivité.
- Si vous créez des bibliothèques personnalisées, implémentez le threading pour exécuter des tâches en parallèle. Cela améliore les performances au travail.
- Ne mettez hors service l'environnement existant qu'une fois que tous les emplois sont stables AWS pendant un certain temps.
- Utilisez Amazon CloudWatch pour la journalisation, afin de réduire les efforts de personnalisation de la journalisation.
- Utilisez Amazon Simple Notification Service (Amazon SNS) pour remplacer les tâches personnalisées d'envoi de notifications. Utilisez Amazon Simple Email Service (Amazon SES) pour remplacer les tâches de messagerie personnalisées.

FAQ

Cette section fournit des réponses aux questions fréquemment posées concernant la migration de vos tâches ETL SSIS vers AWS.

Dois-je améliorer les tâches SSIS pendant la migration ?

Oui. Mettez en œuvre toutes les améliorations ou corrections de bogues importantes dans les tâches existantes et nouvelles en même temps. Les modifications de faible priorité peuvent être mises en œuvre après la migration.

Comment puis-je suivre les offres d'emploi sur AWS ?

Vous pouvez utiliser [Amazon CloudWatch](#) pour surveiller les offres d'emploi et envoyer des alertes basées sur des règles.

Quand puis-je mettre hors service mes tâches SSIS sur site ?

Nous vous recommandons de maintenir les anciennes et les nouvelles tâches en parallèle pendant un certain temps jusqu'à ce que les performances, la précision et l'exhaustivité des données soient les mêmes dans les deux environnements. Vous pouvez également répliquer les systèmes de reporting pour connecter et comparer les rapports des deux environnements. Vous pouvez mettre hors service vos tâches SSIS lorsque les rapports sont identiques.

Étapes suivantes

À l'étape suivante, nous vous recommandons de créer une preuve de concept afin de déterminer votre approche de déploiement pour l'architecture cible. Pour votre preuve de concept :

- Créez des AWS Glue tâches à l'aide de l'interface graphique de AWS Glue Studio.
- Utilisez le AWS Schema Conversion Tool (AWS SCT) pour générer des AWS Glue scripts à partir de packages SSIS.
- Créez un framework de migration personnalisé à l'aide des bibliothèques Spark.
- Créez un cadre de test.
- Identifiez les outils et les processus d'intégration et de déploiement continu (CI/CD).

Ressources connexes

- [Conversion de SSIS en mode AWS Glue utilisation AWS SCT](#)
- [Sauvegarde et restauration Amazon RDS](#)
- [AWS Glue documentation](#)
- [AWS Glue FAQs](#)
- [Enregistrement continu des AWS Glue tâches](#)
- [Documentation Amazon EMR](#)
- [Amazon EMR FAQs](#)
- [Surveillez les métriques avec CloudWatch](#)
- [Accélérez la validation des migrations de données à grande échelle grâce PyDeequ](#)
- [AWS Partenaires compétents en matière de données et d'analyse](#)

Historique du document

Le tableau suivant décrit les modifications importantes apportées à ce guide. Pour être averti des mises à jour à venir, abonnez-vous à un [fil RSS](#).

Modification	Description	Date
Publication initiale	—	6 décembre 2021

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.