Commencer à utiliser l'ETL sans serveur sur AWS Glue

AWS Conseils prescriptifs

AWS Conseils prescriptifs: Commencer à utiliser l'ETL sans serveur sur AWS Glue

Table of Contents

Commencer à utiliser l'ETL sans serveur sur AWS Glue

Dheer Toprani et Adnan Alvee, Amazon Web Services ()AWS

Mars 2024 (historique du document)

Le cloud Amazon Web Services (AWS) AWS Glueest un environnement sans serveur entièrement géré dans lequel vous pouvez extraire, transformer et charger des données (ETL) à grande échelle. Vous pouvez ainsi classer les données, les nettoyer, les enrichir et les déplacer de manière fiable entre différents magasins et flux de données de manière rentable. AWS Glue

AWS Glue est sans serveur, vous n'avez donc pas à vous soucier du provisionnement ou de la gestion des serveurs. Avec AWS Glue, vous ne payez que pour les ressources que vous utilisez, et vous pouvez les augmenter ou les réduire selon vos besoins.

AWS Glue se compose des éléments suivants :

- AWS Glue ETL — L' AWS Glue ETL fournit des options de traitement par lots et de streaming pour extraire, transformer et charger des données d'une source à l'autre.
- AWS Glue Data Catalog— Le catalogue de données est un référentiel central permettant d'organiser les métadonnées de tous vos actifs de données. Data Catalog fournit une interface unifiée dans laquelle vous pouvez rechercher, découvrir et partager des actifs de données entre les services d'analyse de données.
- AWS Glue DataBrew— DataBrew est un outil de préparation de données sans code que vous pouvez utiliser pour explorer, nettoyer et transformer visuellement les données. Vous pouvez choisir parmi plus de 250 transformations prédéfinies pour automatiser les tâches de préparation des données sans écrire de code.

Ce guide fournit une introduction de haut niveau AWS Glue, notamment son fonctionnement et la manière dont vous pouvez commencer à l'utiliser. Il couvre les concepts clés que vous devez connaître avant de créer des AWS Glue tâches, tels que l'automatisation, la surveillance et l'intégration à d'autres AWS services. La section Prochaines étapes vous permettra de vous familiariser avec l'écriture du code AWS Glue. Si vous avez déjà une certaine expérience en la matière AWS Glue, la section Bonnes pratiques vous aidera à combler les lacunes dans vos connaissances. À la fin de ce guide, vous disposerez des connaissances et des ressources dont vous avez besoin pour commencer à l'utiliser AWS Glue efficacement.

AWS Glue ETL

AWS Glue L'ETL permet d'extraire des données de diverses sources, de les transformer pour répondre aux besoins de votre entreprise et de les charger dans la destination de votre choix. Ce service utilise le moteur Apache Spark pour répartir les charges de travail Big Data entre les nœuds de travail, permettant ainsi des transformations plus rapides grâce au traitement en mémoire.

AWS Glue prend en charge diverses sources de données, notamment Amazon Simple Storage Service (Amazon S3), Amazon DynamoDB et Amazon Relational Database Service (Amazon RDS). Pour en savoir plus sur les sources de données prises en charge, consultez la section Types de connexion et options pour ETL dans AWS Glue.

Rédaction dans AWS Glue

AWS Glue propose plusieurs méthodes pour créer des tâches ETL, en fonction de votre expérience et de votre cas d'utilisation :

- Les jobs shell Python sont conçus pour exécuter des scripts ETL de base écrits en Python. Ces tâches s'exécutent sur une seule machine et conviennent mieux aux ensembles de données de petite ou moyenne taille.
- Les tâches Apache Spark peuvent être écrites en Python ou en Scala. Ces tâches utilisent Spark pour dimensionner horizontalement les charges de travail sur de nombreux nœuds de travail, afin de pouvoir gérer des ensembles de données volumineux et des transformations complexes.
- AWS Glue L'ETL de streaming utilise le moteur de streaming structuré Apache Spark pour transformer les données de streaming en tâches de microbatch en utilisant une sémantique unique. Vous pouvez créer des jobs de AWS Glue streaming en Python ou en Scala.
- AWS Glue Studioest une interface de boxes-and-arrows style visuel qui rend l'ETL basé sur Spark accessible aux développeurs novices en programmation Apache Spark.

Unités de traitement de données

AWS Glue utilise des unités de traitement de données (DPUs) pour mesurer les ressources de calcul allouées à une tâche ETL et en calculer le coût. Chaque DPU équivaut à 4 V CPUs et 16 Go de mémoire. DPUsdoit être affecté à votre AWS Glue tâche en fonction de sa complexité et du volume de données. L'allocation du montant approprié vous DPUs permettra de trouver un équilibre entre les besoins de performance et les contraintes de coûts.

AWS Glue fournit plusieurs types de travailleurs optimisés pour différentes charges de travail :

- G.1X ou G.2X (pour la plupart des transformations de données, des jointures et des requêtes)
- G.4X ou G.8X (pour les transformations de données, les agrégations, les jointures et les requêtes plus exigeantes)
- G.025X (pour les flux de données sporadiques et à faible volume)
- Standard (pour AWS Glue les versions 1.0 ou antérieures ; déconseillé pour les versions ultérieures de AWS Glue)

Utilisation du shell Python

Pour une tâche shell Python, vous pouvez utiliser 1 DPU pour utiliser 16 Go de mémoire ou 0,0625 DPU pour utiliser 1 Go de mémoire. Le shell Python est destiné aux tâches ETL de base avec des ensembles de données de petite ou moyenne taille (jusqu'à environ 10 Go).

Comparaison des types de travailleurs

Le tableau suivant présente les différents types de AWS Glue travail pour les charges de travail par lots, en streaming et AWS Glue Studio ETL utilisant l'environnement Apache Spark.

	G.1 X	G.2X	G.4X	G.8 X	G.025X	Standard
vCPU	4	8	16	32	2	4
Mémoire	16 Go	32 GO	64 Go	128 Go	4 Go	16 Go
Espace disque	64 Go	128 Go	256 Go	512 Go	64 Go	50 Go
Exécuteur par travailleur	1	1	1	1	1	2
DPU	1	2	4	8	0.25	1

Le type de travailleur standard n'est pas recommandé pour les AWS Glue versions 2.0 et ultérieures. Le type de travailleur G.025X n'est disponible que pour les tâches de streaming utilisant la AWS Glue version 3.0 ou ultérieure.

AWS Glue Data Catalog

AWS Glue Data CatalogIl s'agit d'un référentiel de métadonnées centralisé pour tous vos actifs de données provenant de différentes sources de données. Il fournit une interface unifiée pour stocker et interroger des informations sur les formats de données, les schémas et les sources. Lorsqu'une tâche AWS Glue ETL est exécutée, elle utilise ce catalogue pour comprendre les informations relatives aux données et s'assurer qu'elles sont correctement transformées.

AWS Glue Data CatalogIl est composé des composants suivants :

- Bases de données et tables
- crawlers et classifieurs
- Connexions
- Registre de schémas

AWS Glue bases de données et tables

AWS Glue Data CatalogIl est organisé en bases de données et en tables afin de fournir une structure logique pour le stockage et la gestion des métadonnées. Cette structure permet un contrôle précis de l'accès aux données au niveau d'une table ou d'une base de données en utilisant des politiques Gestion des identités et des accès AWS (IAM).

Une AWS Glue base de données peut contenir de nombreuses tables, et chaque table doit être associée à une seule base de données. Ces tables contiennent des références aux données réelles, qui peuvent être stockées dans l'une des différentes sources de données prises AWS Glue en charge. AWS Glue les tables stockent également des métadonnées essentielles telles que les noms de colonnes, les types de données et les clés de partition.

Il existe différentes méthodes pour créer une table dans AWS Glue :

- AWS Glue chenille
- AWS Glue Tâche ETL
- AWS Glue console
- `CreateTable`opération dans l'AWS Glue API
- AWS CloudFormation modèle

- Cloud Development Kit AWS (AWS CDK)
- Un métastore Apache Hive migré

AWS Glue chenilles et classificateurs

Un AWS Glue robot d'exploration découvre et extrait automatiquement les métadonnées d'un magasin de données, puis les met à jour AWS Glue Data Catalog en conséquence. Le robot d'exploration se connecte au magasin de données pour déduire le schéma des données. Il crée ou met ensuite à jour des tables dans le catalogue de données avec les informations de schéma qu'il a découvertes. Un analyseur peut analyser des magasins de données basés sur les fichiers et des magasins de données basées sur les tables. Pour en savoir plus sur les magasins de données pris en charge, voir Quels magasins de données puis-je explorer ?

Le robot utilise des classificateurs pour reconnaître avec précision le format des données et déterminer la manière dont elles doivent être traitées. Par défaut, le robot utilise un ensemble de classificateurs intégrés courants fournis par AWS Glue, mais vous pouvez également écrire des classificateurs personnalisés pour gérer des cas d'utilisation spécifiques.

AWS Glue connexions

Vous pouvez utiliser AWS Glue les connexions pour définir les paramètres de connexion qui permettent AWS Glue de se connecter à différentes sources de données. L'ajout de connexions centralise et simplifie la configuration requise pour se connecter à ces sources.

Lorsque vous définissez une connexion, vous spécifiez le type de connexion, le point de terminaison de la connexion et les informations d'identification requises. Une fois qu'une connexion est définie, elle peut être réutilisée par plusieurs AWS Glue jobs et robots d'exploration. L'utilisation de connexions avec AWS Glue réduit le besoin de saisir à plusieurs reprises les mêmes informations de connexion, telles que les informations de connexion ou le cloud privé virtuel (VPC). IDs

AWS Glue Registre des schémas

Le registre des AWS Glue schémas fournit un emplacement centralisé pour la gestion et l'application des schémas de flux de données. Il permet à des systèmes disparates, tels que les producteurs et les consommateurs de données, de partager un schéma de sérialisation et de désérialisation. Le partage d'un schéma permet à ces systèmes de communiquer efficacement et d'éviter les erreurs lors de la transformation.

Le registre des schémas garantit que les consommateurs de données en aval peuvent gérer les modifications apportées en amont, car ils connaissent le schéma attendu. Il prend en charge l'évolution du schéma, de sorte qu'un schéma peut changer au fil du temps tout en conservant la compatibilité avec les versions précédentes du schéma.

Le Schema Registry s'intègre à de nombreux AWS services, notamment Amazon Kinesis Data Streams, Firehose et Amazon Managed Streaming for Apache Kafka. Pour des exemples de cas d'utilisation et d'intégrations, consultez la section Intégration à AWS Glue Schema Registry.

Caractéristiques et concepts importants

Journalisation et surveillance

AWS Glue dispose de plusieurs options de journalisation et de surveillance. Par défaut, AWS Glue envoie les journaux au groupe de `aws-glue` journaux d'Amazon CloudWatch. Ces journaux contiennent des informations telles que les heures de début et de fin, les paramètres de configuration, ainsi que les erreurs ou les avertissements susceptibles de se produire.

En outre, les tâches AWS Glue Spark ETL proposent les options suivantes, qui doivent être activées pour une surveillance avancée :

- Les métriques relatives aux tâches transmettent des mesures spécifiques à la tâche à l'espace de AWS Glue noms CloudWatch toutes les 30 secondes. Ces indicateurs spécifiques à la tâche, tels que les enregistrements traités, la taille totale input/output des données et le temps d'exécution, fournissent des informations sur les performances d'une tâche. Ils peuvent aider à identifier les goulets d'étranglement ou les opportunités d'optimisation des configurations.
- La journalisation continue diffuse les journaux des tâches Apache Spark en temps réel vers le groupe de `/aws-glue/jobs/logs-v2` journalisation CloudWatch. En utilisant des journaux en temps réel, vous pouvez surveiller les AWS Glue tâches de manière dynamique pendant leur exécution.
- L'interface utilisateur de Spark fournit une interface Web de serveur d'historique Spark permettant de consulter les informations relatives à la tâche Spark, telles que la chronologie des événements de chaque étape, un graphe acyclique dirigé et les variables d'environnement de la tâche. Les journaux d'événements persistants de l'interface utilisateur Spark sont stockés dans Amazon S3 et vous pouvez les utiliser en temps réel ou une fois le travail terminé.
- Job Run Insights simplifie le débogage et l'optimisation des tâches en détectant les exceptions courantes de Spark, en analysant les causes premières et en proposant des actions recommandées pour résoudre les problèmes. Les informations sont stockées dans CloudWatch.

Automatisation

AWS Glue propose deux méthodes principales pour automatiser les tâches ETL : les déclencheurs et les flux de travail.

AWS Glue déclencheurs

Lorsqu'ils sont déclenchés, les AWS Glue déclencheurs démarrent les tâches et les robots d'exploration spécifiés. Un déclencheur peut être déclenché à la demande, selon un calendrier prédéfini ou en fonction d'événements spécifiques. Vous pouvez utiliser des déclencheurs pour concevoir une chaîne de tâches et de robots dépendants. Pour plus d'informations, consultez la section AWS Glue Déclencheurs.

AWS Glue flux de travail

Pour les charges de travail plus complexes, vous pouvez utiliser des AWS Glue flux de travail pour créer des graphes acycliques orientés et pour créer des dépendances entre AWS Glue des entités distinctes (déclencheurs, robots d'exploration et tâches). Les flux de travail fournissent également une interface unifiée dans laquelle vous pouvez partager des paramètres, suivre les progrès et résoudre les problèmes entre les entités associées.

La configuration de nombreuses entités associées dans les AWS Glue flux de travail peut devenir de plus en plus complexe. Les développeurs peuvent créer des AWS Glue plans pour partager des pipelines de données complexes avec des data scientists et des analystes commerciaux. Ces modèles permettent la création cohérente et reproductible de AWS Glue flux de travail, en faisant abstraction des détails techniques.

Pour en savoir plus sur les AWS Glue plans et les flux de travail, voir Exécution d'activités ETL complexes à l'aide de plans et de flux de travail dans. AWS Glue

Orchestrer les AWS Glue tâches avec d'autres services AWS

Pour davantage d'options d'automatisation, il AWS Glue s'intègre à d'autres AWS services, tels que AWS Lambda AWS Step Functions, et Amazon Managed Workflows for Apache Airflow (Amazon MWAA).

Pour plus d'informations sur les méthodes d'orchestration pour les tâches AWS Glue ETL, consultez Orchestration dans. AWS Glue

Signets de tâche

Les signets de tâches AWS Glue sont utilisés pour suivre la progression des tâches ETL, ce qui évite de devoir retraiter les données lors des exécutions de tâches suivantes. Lorsque les signets de tâches sont activés, AWS Glue conserve un enregistrement des données déjà traitées. Ensuite, à chaque exécution, il traite uniquement les nouvelles données de la source de données. Pour plus d'informations, consultez la section Suivi des données traitées à l'aide des signets de tâches.

AWS Glue DataBrew

AWS Glue DataBrew est un service de préparation visuelle des données entièrement géré pour le nettoyage, la normalisation et la transformation des données. Il diffère de l' AWS Glue ETL en ce sens que vous n'avez pas besoin d'écrire de code pour l'utiliser. DataBrew fournit plus de 250 transformations intégrées, avec une point-and-click interface visuelle permettant de créer et de gérer des tâches de transformation de données.

DataBrew est disponible dans une vue de console séparée de AWS Glue. Il est intégré nativement à plusieurs AWS services et prend en charge de nombreux formats de fichiers différents. Pour plus d'informations, consultez la section Intégrations de produits et de services.

DataBrew repose sur les six concepts fondamentaux suivants :

- Projet — L'ensemble de l'espace de travail de préparation des données dans DataBrew
- Ensemble de données : ensemble de données structurées ou semi-structurées
- Recette — Un ensemble d'étapes de transformation des données ; chaque étape peut contenir de nombreuses actions
- Job : ensemble d'instructions pour exécuter une recette ou une tâche de profilage de données
- Lignage des données — Suivi des données dans une interface visuelle pour identifier leur origine
- Profil des données : vue récapitulative de la forme de vos données

AWS Glue DataBrew est intégré à AWS Glue Studio, afin que vous puissiez orchestrer des DataBrew recettes dans vos tâches et flux de travail AWS Glue ETL. DataBrew les recettes peuvent également tirer parti de AWS Glue fonctionnalités telles que les signets de tâches, les nouvelles tentatives automatiques et la mise à l'échelle automatique. Pour commencer DataBrew, utilisez l'AWS Glue DataBrew exemple de didacticiel de projet.

Bonnes pratiques

Lorsque vous développez avec AWS Glue, tenez compte des meilleures pratiques suivantes.

Développez d'abord localement

Pour économiser du temps et des coûts lors de la création de vos tâches ETL, testez d'abord votre code et votre logique métier localement. Pour obtenir des instructions sur la configuration d'un conteneur Docker qui peut vous aider à tester des tâches AWS Glue ETL à la fois dans un shell et dans un environnement de développement intégré (IDE), consultez le billet de blog Développez et testez des AWS Glue tâches localement à l'aide d'un conteneur Docker.

Utilisez des sessions AWS Glue interactives

AWS Glue les sessions interactives fournissent un backend Spark sans serveur, associé à un noyau Jupyter open source qui s'intègre aux ordinateurs portables IDEs tels que PyCharm IntelliJ et VS Code. En utilisant des sessions interactives, vous pouvez tester votre code sur des ensembles de données réels avec le backend AWS Glue Spark et l'IDE de votre choix. Pour commencer, suivez les étapes décrites dans Getting started with AWS Glue interactive sessions.

Utilisez le partitionnement pour rechercher exactement ce dont vous avez besoin

Le partitionnement consiste à diviser un grand ensemble de données en partitions plus petites en fonction de colonnes ou de clés spécifiques. Lorsque les données sont partitionnées, AWS Glue vous pouvez effectuer des analyses sélectives sur un sous-ensemble de données répondant à des critères de partitionnement spécifiques, plutôt que de scanner l'ensemble de données dans son intégralité. Cela se traduit par un traitement des requêtes plus rapide et plus efficace, en particulier lorsque vous travaillez avec de grands ensembles de données.

Partitionnez les données en fonction des requêtes qui seront exécutées sur celles-ci. Par exemple, si la plupart des requêtes filtrent sur une colonne en particulier, le partitionnement sur cette colonne peut réduire considérablement le temps de requête. Pour en savoir plus sur le partitionnement des données, consultez la section Utilisation de données partitionnées dans. AWS Glue

Optimisez la gestion de la mémoire

La gestion de la mémoire est cruciale lors de l'écriture de tâches AWS Glue ETL, car celles-ci s'exécutent sur le moteur Apache Spark, qui est optimisé pour le traitement en mémoire. Le billet de blog Optimize memory management in AWS Glue fournit des détails sur les techniques de gestion de mémoire suivantes :

- Implémentation de la liste Amazon S3 de AWS Glue
- Regroupement
- Exclusion des chemins et des classes de stockage Amazon S3 non pertinents
- Partitionnement Spark et AWS Glue Read
- Inserts en vrac
- Optimisations des jointures
- PySpark fonctions définies par l'utilisateur () UDFs
- Traitement incrémentiel

Utilisez des formats de stockage de données efficaces

Lorsque vous créez des tâches ETL, nous vous recommandons de générer les données transformées dans un format de données basé sur des colonnes. Les formats de données en colonnes, tels qu'Apache Parquet et ORC, sont couramment utilisés dans le stockage et l'analyse de mégadonnées. Ils sont conçus pour minimiser le mouvement des données et optimiser la compression. Ces formats permettent également de diviser les données entre plusieurs lecteurs pour augmenter le nombre de lectures parallèles lors du traitement des requêtes.

La compression des données permet également de réduire la quantité de données stockées et d'améliorer les performances des opérations de lecture/écriture. AWS Glue prend en charge plusieurs formats de compression de manière native. Pour plus d'informations sur l'efficacité du stockage et de la compression des données, consultez la section Création d'un pipeline de données performant.

Utilisez le type de mise à l'échelle approprié

Il est important de comprendre quand procéder à une mise à l'échelle horizontale (modification du nombre de travailleurs) ou à une échelle verticale (modification du type de travailleur), AWS Glue car cela peut avoir un impact sur le coût et les performances des tâches ETL.

En général, les tâches ETL comportant des transformations complexes consomment plus de mémoire et nécessitent une mise à l'échelle verticale (par exemple, le passage du type de travail G-1X au type de travail G.2X). Pour les tâches ETL gourmandes en ressources informatiques impliquant de gros volumes de données, nous recommandons la mise à l'échelle horizontale, car elle AWS Glue est conçue pour traiter ces données en parallèle sur plusieurs nœuds.

En surveillant de près les indicateurs relatifs aux AWS Glue tâches sur CloudWatch Amazon, vous pouvez déterminer si un goulot d'étranglement en matière de performances est dû à un manque de mémoire ou de calcul. Pour plus d'informations sur les types de AWS Glue travailleurs et le dimensionnement, consultez Bonnes pratiques pour dimensionner les tâches Apache Spark et partitionner les données avec AWS Glue.

FAQ sur AWS Glue l'ETL sans serveur

Cette section fournit des réponses aux questions fréquemment posées à propos de l'ETL sans serveur activé AWS Glue.

Quand dois-je utiliser le shell Python au lieu d'Apache Spark pour les AWS Glue tâches ?

Utilisez le shell Python lorsque vous avez des tâches ETL de base ou de petits ensembles de données qui ne nécessitent pas les capacités informatiques distribuées d'Apache Spark. Utilisez Apache Spark pour des tâches ETL plus complexes ou pour des ensembles de données volumineux qui nécessitent la puissance de traitement élevée pour laquelle Spark est optimisé.

Quelle est la AWS Glue version recommandée pour mon projet ?

Nous recommandons généralement d'utiliser la dernière version de AWS Glue. La page des AWS Glue versions répertorie les différences entre les versions, ainsi que leur compatibilité avec les différentes versions de Python et de Spark.

Étapes suivantes

Comprendre les AWS Glue transformations

Pour un traitement des données plus efficace, AWS Glue inclut des fonctions de transformation intégrées. Les fonctions passent d'une transformation à l'autre dans une structure de données appelée a DynamicFrame, qui est une extension d'un SQL Apache Spark DataFrame. A DynamicFrame est similaire à a DataFrame, sauf que chaque enregistrement est autodescriptif, de sorte qu'aucun schéma n'est requis au départ.

Pour vous familiariser avec plusieurs fonctions AWS Glue PySpark intégrées, consultez le billet de blog Construire un pipeline AWS Glue ETL localement sans Compte AWS.

Création de votre première tâche ETL

Si vous n'avez jamais écrit de tâche ETL auparavant, vous pouvez commencer par utiliser les trois types de tâches AWS Glue ETL pour convertir les données en modèle Apache Parquet.

Si vous avez de l'expérience dans la rédaction de tâches ETL, vous pouvez utiliser les AWS Glue GitHub exemples pour les explorer plus en profondeur.

Tarification

Pour en savoir plus sur la tarification, consultez Tarification AWS Glue. Vous pouvez également utiliser le Calculateur de tarification AWSpour estimer le coût mensuel de l'utilisation de différents AWS Glue composants.

Ressources supplémentaires

Références

- AWS Glue Manuel du développeur
- AWS Glue DataBrew Manuel du développeur
- API AWS Glue
- AWS Glue diffusion ETL
- AWS Glue Studio
- Tâches Shell en Python dans AWS Glue
- Emplois Apache Spark
- AWS Glue PySparktransforme la référence
- Catalogue de données et robots d'exploration dans AWS Glue
- AWS Glue Registre des schémas
- Connexion et surveillance AWS Glue
- Versions AWS Glue

Billets de blogs

- Développez et testez AWS Glue des jobs localement à l'aide d'un conteneur Docker
- Optimisez la gestion de la mémoire dans AWS Glue
- Meilleures pratiques pour dimensionner les tâches Apache Spark et partitionner les données avec AWS Glue
- Construire un pipeline AWS Glue ETL localement sans Compte AWS
- Travaillez avec des données partitionnées dans AWS Glue

Exemples

- AWS Glue DataBrew exemple de projet
- Débuter avec des sessions AWS Glue interactives
- AWS Glue Référentiel d'exemples de code ETL

Motifs

- Créez un pipeline de services ETL pour charger les données de manière incrémentielle d'Amazon S3 vers Amazon Redshift en utilisant AWS Glue
- Déployez et gérez un lac de données sans serveur sur le AWS Cloud en utilisant l'infrastructure sous forme de code
- Trois types de tâches AWS Glue ETL pour convertir des données vers Apache Parquet

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
Mis à jour	Nous avons ajouté des informations sur les types de travailleurs dans la section AWS Glue ETL.	13 mars 2024
Mis à jour	Contenu mis à jour tout au long du guide.	15 mars 2023
Ajout d'une meilleure pratique	Nous avons ajouté des informations sur l'utilisation du partitionnement pour rechercher des données spécifiques.	4 février 2021
Publication initiale	—	29 janvier 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.