aws

Guide du développeur

AWS HealthLake

AWS HealthLake: Guide du développeur

Table of Contents

Qu'est-ce que c'est AWS HealthLake ?

AWS HealthLake est un service HIPAA éligible pour l'ingestion, le stockage et l'analyse de données cliniques utilisant la spécification Healthcare Interoperability FHIR (R4).

> **Note**
>
> Après le 20 février 2023, les magasins de HealthLake données n'utilisent pas le traitement du langage naturel intégré (NLP) par défaut. Si vous souhaitez activer cette fonctionnalité sur votre banque de données, consultez le Comment activer la fonction intégrée HealthLake de traitement du langage naturel ? chapitre Dépannage.

Les données de santé sont souvent incomplètes et incohérentes. Il est également souvent non structuré, avec des informations contenues dans des notes cliniques, des rapports de laboratoire, des réclamations d'assurance, des images médicales, des conversations enregistrées et des données chronologiques (par exemple, des EEG traces cardiaques ECG ou cérébrales).

Les prestataires de soins de santé peuvent les utiliser HealthLake pour stocker, transformer, interroger et analyser des données dans le AWS cloud. À l'aide des fonctionnalités HealthLake intégrées de traitement du langage naturel médical (NLP), vous pouvez analyser du texte clinique non structuré provenant de diverses sources. HealthLake transforme les données non structurées à l'aide de modèles de traitement du langage naturel et fournit de puissantes fonctionnalités de requête et de recherche. Vous pouvez l'utiliser HealthLake pour organiser, indexer et structurer les informations sur les patients de manière sécurisée, conforme et pouvant être auditée.

HealthLake est également intégré à Amazon Athena et à AWS Lake Formation. Vous pouvez utiliser cette intégration pour interroger votre banque de données à l'aide deSQL.

Avantages de AWS HealthLake

Avec AWS HealthLake, vous pouvez :

- Ingérez rapidement et facilement des données de santé : vous pouvez importer en masse des fichiers Fast Healthcare Interoperability Resources (FHIR) sur site, notamment des notes cliniques, des rapports de laboratoire, des réclamations d'assurance, etc., dans un bucket Amazon Simple Storage Service (Amazon S3). Vous pouvez ensuite utiliser les données dans des applications ou des flux de travail en aval.

- Utiliser les FHIR REST API opérations : HealthLake permet d'utiliser les FHIR REST API opérations pour effectuer CRUD (Create/Read/Update/Delete) des opérations sur votre magasin de données. FHIRla recherche est également prise en charge.
- Stockez vos données dans le AWS cloud d'une manière sécurisée et HIPAA éligible qui peut être auditée — Vous pouvez stocker les données dans le FHIR format qui permet de les interroger facilement. HealthLake crée une vue chronologique complète des antécédents médicaux de chaque patient et la structure au format FHIR standard R4.
- Intégration d'Athena : HealthLake l'intégration avec Athena vous permet de créer de puissantes SQL requêtes que vous pouvez utiliser pour créer et enregistrer des critères de filtre complexes. Vous pouvez ensuite utiliser ces données dans des applications en aval, telles que l' SageMaker IA pour entraîner un modèle d'apprentissage automatique ou Amazon QuickSight pour créer des tableaux de bord et des visualisations de données.
- Transformez des données non structurées à l'aide de modèles d'apprentissage automatique (ML) spécialisés : HealthLake fournit un traitement médical intégré du langage naturel (NLP) à l'aide d'Amazon Comprehend Medical. Les données textuelles médicales brutes sont transformées à l'aide de modèles de machine learning spécialisés. Ces modèles ont été formés pour comprendre et extraire des informations pertinentes à partir de données de santé non structurées. Grâce à la médecine intégréeNLP, vous pouvez extraire automatiquement des entités (par exemple, des procédures médicales et des médicaments), des relations entre entités (par exemple, un médicament et sa posologie) et des données relatives aux caractéristiques des entités (par exemple, résultat de test positif ou négatif ou heure de l'intervention) de votre texte médical. HealthLake crée ensuite de nouvelles ressources en fonction du signe, du symptôme et de l'état des traits. Ils sont ajoutés en tant que nouveaux types de condition, d'observation et de MedicationStatement ressource.

HealthLake cas d'utilisation

Vous pouvez l'utiliser HealthLake pour les applications médicales suivantes :

- Gestion de la santé de la population : HealthLake aide les établissements de santé à analyser les tendances, les résultats et les coûts en matière de santé de la population. Cela aide les organisations à identifier l'intervention la plus appropriée pour une population de patients et à choisir de meilleures options de gestion des soins.
- Améliorer la qualité des soins : HealthLake aide les hôpitaux, les compagnies d'assurance maladie et les organisations du secteur des sciences de la vie à combler les lacunes en matière de

soins, à améliorer la qualité des soins et à réduire les coûts en compilant une vue complète des antécédents médicaux d'un patient.

- Optimisation de l'efficacité HealthLake des hôpitaux : offre aux hôpitaux des outils d'analyse et d'apprentissage automatique essentiels pour améliorer l'efficacité et réduire le gaspillage hospitalier.

Accès HealthLake

Vous pouvez y accéder HealthLake via le AWS Management Console, AWS Command Line Interface (AWS CLI) ou le AWS SDKs.

1. AWS Management Console — Fournit une interface Web à laquelle vous pouvez accéder HealthLake.
2. AWS Command Line Interface (AWS CLI) — Fournit des commandes pour un large éventail de AWS services, notamment HealthLake, et est compatible avec Windows, macOS et Linux. Pour plus d'informations sur l'installation du AWS CLI, consultez AWS Command Line Interface.
3. AWS SDKs— AWS fournit SDKs (kits de développement logiciel) composés de bibliothèques et d'exemples de code pour différents langages de programmation et plateformes (Java, Python, Ruby,. NET, iOS, Android, etc.). Ils SDKs fournissent un moyen pratique de créer un accès programmatique à HealthLake et AWS. Pour plus d'informations, consultez le AWSSDKpour Python.

HIPAAéligibilité et sécurité des données

Il s'agit d'un service HIPAA éligible. Pour plus d'informations sur AWS le Health Insurance Portability and Accountability Act des États-Unis de 1996 (HIPAA) et sur l'utilisation de AWS services pour traiter, stocker et transmettre des informations de santé protégées (PHI), voir HIPAAPrésentation.

Les connexions HealthLake contenant des informations personnellement identifiables (PII) doivent être cryptées. Par défaut, toutes les connexions à HealthLake utiliser sont HTTPS terminéesTLS. HealthLake stocke le contenu crypté des clients et fonctionne selon le principe de responsabilité AWS partagée.

Tarification

Pour plus d'informations sur HealthLake les tarifs, consultez la page AWS HealthLake des tarifs. Pour mieux estimer les coûts potentiels associés HealthLake, vous pouvez utiliser le calculateur de HealthLake prix.

Comment AWS HealthLake fonctionne

AWS HealthLake crée un magasin de données qui stocke les dossiers médicaux en utilisant la spécification Healthcare Interoperability FHIR (R4). Avec HealthLake, vous pouvez effectuer les tâches suivantes.

> **Note**
>
> Après le 20 février 2023, les magasins de HealthLake données n'utilisent pas le traitement automatique du langage naturel intégré (NLP) par défaut. Si vous souhaitez activer cette fonctionnalité sur votre banque de données, consultez le Comment activer la fonction intégrée HealthLake de traitement du langage naturel ? chapitre Dépannage.

- Créez, surveillez et supprimez un magasin de données.
- `StartFHIRImportJob`À utiliser pour importer des données de santé en masse depuis un bucket Amazon Simple Storage Service (Amazon S3) vers un magasin de données.
- Utilisez les opérations Create, Read, Update et Delete (CRUD) pour gérer les données stockées dans votre magasin de données.
- SQLUtilisez-le dans Amazon Athena pour interroger votre banque de données.
- Utilisez un HTTP client dans les FHIR REST API opérations pour effectuer des recherches dans votre banque de données.
- Permettez aux opérations Amazon Comprehend API Medical de rechercher des informations médicales dans vos données à l'aide du traitement du langage naturel NLP ().

Création et surveillance de magasins de données

Avec HealthLake, vous pouvez créer et surveiller des magasins de données capables de stocker les données Fast Healthcare Interoperability Resources (FHIR).

Pour créer un nouveau magasin de données, vous pouvez utiliser le C reateFHIRDatastore ou la HealthLake console. Pour connaître l'état d'un magasin de données, utilisez escribeFHIRDatastoreD. Pour voir l'état de plusieurs magasins de données actifs, utilisez istFHIRDatastoresL. Pour supprimer un magasin de données, utilisez eleteFHIRDatastoreD.

FHIRRESTAPIopérations

Vous pouvez utiliser ces FHIR REST API opérations pour effectuer des opérations de création, de lecture, de mise à jour, de suppression (CRUD) sur votre magasin de HealthLake données. Pour en savoir plus sur la manière dont HealthLake les FHIR REST API opérations sont prises en charge, voirUtilisation FHIR REST API des interactions avec un magasin HealthLake de données.

Génération automatique de ressources à partir d'extensions de FHIR DocumentReference ressources

> **Note**
>
> Lorsque vous créez un magasin de HealthLake données et que vous ajoutez des données qui le contiennent`DocumentReference`, des frais sont facturés sur votre AWS compte. Pour plus de détails, consultez AWS HealthLake les tarifs.

HealthLake fournit des NLP informations sur les documents trouvés dans le type de `DocumentReference` ressource. Pour analyser le texte, HealthLake utilise les opérations Amazon Comprehend API Medical suivantes.

- `DetectEntitiesV2`: Inspecte le texte clinique de diverses entités médicales et renvoie des informations spécifiques les concernant, telles que la catégorie de l'entité, l'emplacement et le score de confiance.
- `InferICD10CM`: inspecte le texte clinique pour détecter les affections médicales sous la forme d'entités répertoriées dans le dossier d'un patient et associe ces entités à des identificateurs conceptuels normalisés dans la base de connaissances ICD -10 CM des Centers for Disease Control.
- `InferRxNorm`: Inspecte le texte clinique pour détecter les médicaments en tant qu'entités répertoriées dans le dossier d'un patient et établit des liens vers les identificateurs conceptuels normalisés de la RxNorm base de données de la National Library of Medicine.

HealthLake analyse automatiquement les données présentes dans le type de `DocumentReference` ressource lorsqu'il est ajouté à votre banque de données. Les fichiers de `DocumentReference` ressources d'origine restent inchangés. Les informations médicales extraites sont automatiquement

ajoutées sous forme d'extensions FHIR conformes. Pour en savoir plus sur le NLP fonctionnement dans HealthLake, voirUtilisation de la génération automatique de ressources basée sur le traitement du langage naturel (NLP) du type de FHIR DocumentReference ressource dans AWS HealthLake.

Recherche à l'aide SQL de requêtes basées

> **Note**
>
> Pour les banques de données créées avant le 14 novembre 2022, votre recherche est limitée aux FHIR REST API opérations. Pour utiliser des requêtes SQL basées sur les données de votre HealthLake banque de données, voirInterrogez les magasins de AWS HealthLake données SQL à l'aide d'Amazon Athena.

Amazon Athena est un service de requête sans serveurSQL. HealthLake les magasins de données sont ingérés dans Athena sous forme de tables Apache Iceberg. Ces tables sont conçues pour prendre en charge de grands ensembles de données analytiques. Dans Athena, chaque type de FHIR ressource est représenté sous forme de tableau. Avec Athena, vous ne pouvez effectuer des `READ` demandes que sur votre banque de données. Pour en savoir plus sur la recherche SQL basée, voirInterrogez votre banque HealthLake de données à l'aide de SQL.

Recherche à l'aide d'FHIRRESTAPIopérations

Vous pouvez rechercher les dossiers médicaux stockés dans votre banque de données soit en spécifiant un type de ressource avec des paramètres de recherche pris en charge, soit en utilisant un identifiant de ressource trouvé sur le serveur, sans spécifier le type de ressource. Pour en savoir plus sur la recherche à l'aide FHIR REST API des opérations, voirUtilisation FHIR REST API des interactions avec un magasin HealthLake de données.

Actions relatives aux importations de données

AWS HealthLake À utiliser pour importer vos fichiers en masse à partir d'un compartiment Amazon S3. Utilisez la console ou S tartFHIRImport Job pour commencer une tâche d'importation. Après avoir importé vos fichiers, vous pouvez utiliser D escribeFHIRImport Job pour surveiller l'état de la tâche. Une fois le travail d'importation terminé, les données peuvent être ajoutées à Athena, transformées ou analysées et utilisées dans des applications en aval.

Actions pour les exportations de données

HealthLake Utilisez-le pour exporter vos fichiers en masse vers un compartiment Amazon S3. Utilisez la console ou S tartFHIRExport Job pour commencer une tâche d'exportation. Après avoir exporté vos fichiers, vous pouvez utiliser D escribeFHIRExport Job pour surveiller l'état de la tâche et consulter ses propriétés. Une fois le travail d'exportation terminé, vous pouvez visualiser les données à l'aide d'Amazon QuickSight ou vous pouvez y accéder à l'aide d'autres AWS services.

AWS HealthLake validations FHIR de profil prises en charge

HealthLake prend en charge la spécification FHIR R4 de base. Les FHIR profils sont inclus dans la spécification R4. Les profils sont utilisés sur un type de FHIR ressource pour définir une définition de type de ressource plus spécifique à l'aide de contraintes et/ou d'extensions sur le type de ressource de base. Par exemple, un FHIR profil peut identifier les champs obligatoires tels que les extensions et les ensembles de valeurs. Une ressource peut prendre en charge plusieurs profils. Tous les magasins HealthLake de données prennent en charge l'utilisation FHIR de profils.

L'ajout d'un FHIR profil n'est pas obligatoire lors de l'ajout de données à un magasin de HealthLake données. Si aucun FHIR profil n'est spécifié lors de l'ajout ou de la mise à jour d'une ressource, la ressource est uniquement validée par rapport au schéma FHIR R4 de base.

FHIRLes profils auxquels une ressource est conforme sont inclus dans la ressource avant son ingestion. HealthLake HealthLake valide les FHIR profils spécifiés lorsqu'ils sont ajoutés à votre banque de HealthLake données.

FHIRLes profils sont spécifiés dans un guide de mise en œuvre. HealthLake valide les FHIR profils définis dans les guides d'implémentation suivants.

FHIRProfils pris en charge par HealthLake

Nom	Versi	Guide d'implémentation	Capacité
Noyau américain	3.1.1	http://hl7.org/fhir/us/core/STU3.1.1/	Par défaut
Noyau américain	4.0.0	https://hl7.org/fhir/us/core/STU4/index.html	Pris en charge
CARINBouton bleu	1.1.0	http://hl7.org/fhir/us/carin-bb/STU1.1/	Par défaut
CARINBouton bleu	1.0.0	https://hl7.org/fhir/us/carin-bb/STU1/	Pris en charge
Da Vinci Payer Data Exchange	1.0.0	https://hl7.org/fhir/us/davinci-pdex/	Par défaut
Échange de dossiers médicaux Da Vinci (HRex)	0.2.0	https://hl7.org/fhir/us/davinci-hrex/2020Sep/	Par défaut

Nom	Versi	Guide d'implémentation	Capacité
DaVinci PDEXPlan Net	1.1.0	https://hl7.org/fhir/us/davinci-pdex-plan-net/STU1.1/	Par défaut
DaVinci PDEXPlan Net	1.0.0	https://hl7.org/fhir/us/davinci-pdex-plan-net/STU1/	Pris en charge
DaVinci Payer Data Exchange (PDex) Formulaire de médicaments américain	1.1.0	https://hl7.org/fhir/us/davinci-drug-formulary/STU1.1/	Par défaut
DaVinci Payer Data Exchange (PDex) Formulaire de médicaments américain	1.0.1	https://hl7.org/fhir/us/davinci-drug-formulary/STU1.0.1/	Pris en charge
Mission numérique Ayushman Bharat de la National Health Authority () ABDM	2.0	https://www.nrces.in/ndhm/fhir/r4/index.html	Par défaut

Validation FHIR des profils spécifiés dans une ressource

Pour qu'un FHIR profil soit validé, ajoutez-le à l'`profile`élément des ressources individuelles en utilisant le profil URL indiqué dans le guide de mise en œuvre.

FHIRLes profils sont validés lorsque vous ajoutez une nouvelle ressource à votre banque de données. Pour ajouter une nouvelle ressource, vous pouvez utiliser l'APIopération S tartFHIRImport Job, faire une POST demande pour ajouter une nouvelle ressource ou faire PUT pour mettre à jour une ressource existante.

Example — Pour voir quel FHIR profil est référencé dans une ressource

Le profil URL est ajouté à l'`profile`élément de la paire `"meta" : "profile"` clé-valeur. Cette ressource a été tronquée pour des raisons de clarté.

```
           {
            "resourceType": "Patient",
            "id": "abcd1234efgh5678hijk9012",
            "meta": {
             "lastUpdated": "2023-05-30T00:48:07.8443764-07:00",
             "profile": [
              "http://hl7.org/fhir/us/core/StructureDefinition/us-core-patient"
             ]
            }
           }
```

Example — Comment référencer un profil pris en charge FHIR autre que le profil par défaut

Pour valider par rapport à un profil pris en charge autre que le profil par défaut (par ex. CarinBB 1.0.0) - ajoutez le profil URL avec la version (séparés par « | ») et le profil de base URL dans l'élément. `meta.profile` Cet exemple de ressource a été tronqué pour des raisons de clarté.

```
           {
            "resourceType": "ExplanationOfBenefit",
            "id": "sample-EOB",
            "meta": {
             "lastUpdated": "2024-02-02T05:56:09.4+00:00",
             "profile": [
              "http://hl7.org/fhir/us/carin-bb/StructureDefinition/C4BB-
ExplanationOfBenefit-Pharmacy|1.0.0",
               "http://hl7.org/fhir/us/carin-bb/StructureDefinition/C4BB-
ExplanationOfBenefit-Pharmacy“
             ]
            }
           }
```

Types de données préchargés

HealthLake n'est pris en charge qu'SYNTHEAen tant que type de données préchargé. Synthea est un générateur de patients synthétique qui modélise les antécédents médicaux des patients générés par des modèles. Il s'agit d'un dépôt Git open source qui permet de générer des ensembles de ressources FHIR conformes HealthLake à la norme R4 afin que les utilisateurs puissent tester des modèles sans utiliser les données réelles des patients.

Les types de ressources suivants sont disponibles dans les magasins de données préchargés.

Types de ressources Synthea pris en charge

AllergyIntolerance	Emplacement
CarePlan	MedicationAdministration
CareTeam	MedicationRequest
Demander	Observation
Condition	Organisation
Appareil	Patient
DiagnosticReport	Praticien
Rencontre	PractitionerRole
ExplanationofBenefit	Procédure
ImagingStudy	Provenance
Immunisation	

Configuration des autorisations pour commencer à utiliser AWS HealthLake

Dans ce chapitre, vous allez utiliser le AWS Management Console pour configurer les autorisations requises pour commencer à utiliser AWS HealthLake et à créer un magasin de données. Pour configurer les autorisations nécessaires à la création d'un magasin de données, vous devez créer un IAM utilisateur ou un rôle qui est un administrateur et HealthLake un administrateur de lac de données. Vous faites de cet utilisateur un administrateur de lac de données dans AWS Lake Formation. L'administrateur du lac de données accorde à Lake Formation l'accès aux ressources nécessaires pour utiliser Amazon Athena afin d'interroger un magasin de données.

Après avoir créé un magasin de données dans HealthLake, vous pouvez configurer des autorisations pour importer des fichiers dans le magasin de données ou pour les exporter. Pour plus d'informations sur la configuration des autorisations d'importation de fichiers, consultezConfiguration des autorisations pour les tâches d'importation. Pour plus d'informations sur la configuration des autorisations d'exportation de fichiers, consultezConfiguration des autorisations pour les tâches d'exportation.

Rubriques

- Inscrivez-vous pour un Compte AWS
- Création d'un utilisateur doté d'un accès administratif
- Configuration d'un IAM utilisateur ou d'un rôle à utiliser HealthLake (IAMadministrateur)
- Ajouter un utilisateur ou un rôle en tant qu'administrateur du lac de données dans Lake Formation (IAMadministrateur)

Inscrivez-vous pour un Compte AWS

Si vous n'en avez pas Compte AWS, procédez comme suit pour en créer un.

Pour vous inscrire à un Compte AWS

1. Ouvrez l'https://portal.aws.amazon.com/billing/inscription.
2. Suivez les instructions en ligne.

   Dans le cadre de la procédure d‘inscription, vous recevrez un appel téléphonique et vous saisirez un code de vérification en utilisant le clavier numérique du téléphone.

Lorsque vous vous inscrivez à un Compte AWS, un Utilisateur racine d'un compte AWSest créé. Par défaut, seul l'utilisateur racine a accès à l'ensemble des Services AWS et des ressources de ce compte. La meilleure pratique de sécurité consiste à attribuer un accès administratif à un utilisateur, et à utiliser uniquement l'utilisateur racine pour effectuer les tâches nécessitant un accès utilisateur racine.

AWS vous envoie un e-mail de confirmation une fois le processus d'inscription terminé. À tout moment, vous pouvez consulter l'activité actuelle de votre compte et gérer votre compte en accédant à https://aws.amazon.com/et en choisissant Mon compte.

Création d'un utilisateur doté d'un accès administratif

Après vous être inscrit à un Compte AWS, sécurisez Utilisateur racine d'un compte AWS AWS IAM Identity Center, activez et créez un utilisateur administratif afin de ne pas utiliser l'utilisateur root pour les tâches quotidiennes.

Sécurisez votre Utilisateur racine d'un compte AWS

1. Connectez-vous en AWS Management Consoletant que propriétaire du compte en choisissant Utilisateur root et en saisissant votre adresse Compte AWS e-mail. Sur la page suivante, saisissez votre mot de passe.

   Pour obtenir de l'aide pour vous connecter en utilisant l'utilisateur racine, consultez Connexion en tant qu'utilisateur racine dans le Guide de l'utilisateur Connexion à AWS .
2. Activez l'authentification multifactorielle (MFA) pour votre utilisateur root.

   Pour obtenir des instructions, voir Activer un MFA périphérique virtuel pour votre utilisateur Compte AWS root (console) dans le guide de IAM l'utilisateur.

Création d'un utilisateur doté d'un accès administratif

1. Activez IAM Identity Center.

   Pour obtenir des instructions, consultez Activation d' AWS IAM Identity Center dans le Guide de l'utilisateur AWS IAM Identity Center .
2. Dans IAM Identity Center, accordez un accès administratif à un utilisateur.

Pour un didacticiel sur l'utilisation du Répertoire IAM Identity Center comme source d'identité, voir Configurer l'accès utilisateur par défaut Répertoire IAM Identity Center dans le Guide de AWS IAM Identity Center l'utilisateur.

Connexion en tant qu‘utilisateur doté d'un accès administratif

- Pour vous connecter avec votre utilisateur IAM Identity Center, utilisez l'URLidentifiant envoyé à votre adresse e-mail lorsque vous avez créé l'utilisateur IAM Identity Center.

  Pour obtenir de l'aide pour vous connecter en utilisant un utilisateur d'IAMIdentity Center, consultez la section Connexion au portail AWS d'accès dans le guide de Connexion à AWS l'utilisateur.

Attribution d'un accès à d'autres utilisateurs

1. Dans IAM Identity Center, créez un ensemble d'autorisations conforme à la meilleure pratique consistant à appliquer les autorisations du moindre privilège.

   Pour obtenir des instructions, consultez Création d’un ensemble d’autorisations dans le Guide de l’utilisateur AWS IAM Identity Center .
2. Attribuez des utilisateurs à un groupe, puis attribuez un accès par authentification unique au groupe.

   Pour obtenir des instructions, consultez Ajout de groupes dans le Guide de l’utilisateur AWS IAM Identity Center .

Configuration d'un IAM utilisateur ou d'un rôle à utiliser HealthLake (IAMadministrateur)

> Persona : Administrator IAM
>
> Un utilisateur qui peut créer des IAM utilisateurs et des rôles, et qui peut ajouter des administrateurs de data lake.

Les étapes décrites dans cette rubrique doivent être effectuées par un IAM administrateur.

Pour connecter votre banque de HealthLake données à Athena, vous devez créer un IAM utilisateur ou un rôle à la fois administrateur de lac de données et administrateur. HealthLake Ce nouvel utilisateur ou rôle accorde l'accès aux ressources trouvées dans un magasin de données via AWS Lake Formation, et la politique `AmazonHealthLakeFullAccess` AWS gérée est ajoutée à son utilisateur ou à son rôle.

> **Important**
>
> Un IAM utilisateur ou un rôle administrateur de lac de données ne peut pas créer de nouveaux administrateurs de lac de données. Pour ajouter un administrateur de lac de données supplémentaire, vous devez utiliser un IAM utilisateur ou un rôle auquel l'`AdministratorAccess`accès a été accordé.

Pour créer un administrateur

1. Ajoutez la politique **`AmazonHealthlakeFullAccess`** IAM AWS gérée à un utilisateur ou à un rôle au sein de votre organisation.

   Si vous n'êtes pas habitué à créer un IAM utilisateur, consultez les sections Création IAM d'un utilisateur et Présentation des AWS IAM politiques dans le guide de IAM l'utilisateur.
2. Accordez à l'IAMutilisateur ou au rôle l'accès à AWS Lake Formation.

   - Ajoutez la politique IAM AWS gérée suivante à un utilisateur ou à un rôle au sein de votre organisation : **`AWSLakeFormationDataAdmin`**

     > **Note**
     >
     > La `AWSLakeFormationDataAdmin` politique donne accès à toutes les ressources AWS du Lake Formation. Nous vous recommandons de toujours utiliser les autorisations minimales requises pour accomplir votre tâche. Pour plus d'informations, consultez la section IAMBonnes pratiques du guide de IAM l'utilisateur.
3. Ajoutez la politique intégrée suivante à l'utilisateur ou au rôle. Pour plus d'informations, consultez la section Politiques intégrées dans le guide de l'IAMutilisateur.

```
{
    "Version": "2012-10-17",
    "Statement": [
```

```
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-source-bucket/*",
                "arn:aws:s3:::amzn-s3-demo-logging-bucket/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ram:GetResourceShareInvitations",
                "ram:AcceptResourceShareInvitation",
                "glue:CreateDatabase",
                "glue:DeleteDatabase"
            ],
            "Resource": "*"
        }
    ]
}
```

Pour plus d'informations sur cette `AWSLakeFormationDataAdmin` politique, consultez la section Lake Formation Personas and IAM Permissions Reference dans le AWS Lake Formation Developer Guide.

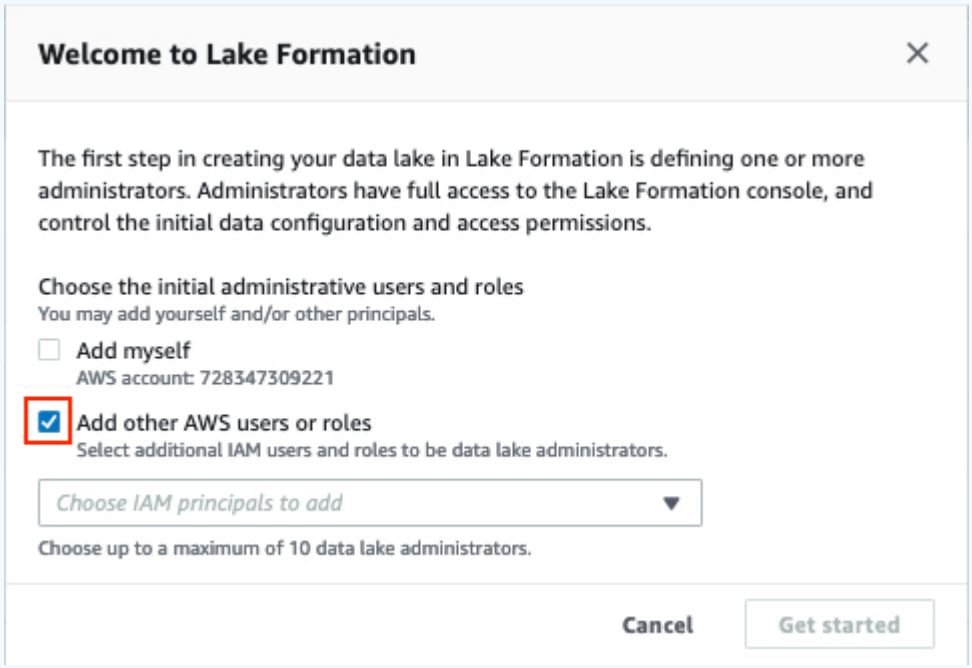
Ajouter un utilisateur ou un rôle en tant qu'administrateur du lac de données dans Lake Formation (IAMadministrateur)

Ensuite, l'IAMadministrateur doit ajouter l'utilisateur ou le rôle créé à l'étape 1 en tant qu'administrateur de lac de données dans Lake Formation.

Pour ajouter un IAM utilisateur ou un rôle en tant qu'administrateur du lac de données

1. Ouvrez la console AWS Lake Formation : https://console.aws.amazon.com/lakeformation/

> **Note**
>
> Si c'est la première fois que vous visitez Lake Formation, une boîte de dialogue Welcome to Lake Formation apparaît, vous demandant de définir un administrateur de Lake Formation.



2. Attribuez au nouvel utilisateur ou au nouveau rôle un administrateur AWS du lac de données de Lake Formation.
   - Option 1 : Si vous avez reçu la boîte de dialogue Welcome to Lake Formation.
     1. Choisissez Ajouter d'autres AWS utilisateurs ou rôles.
     2. Cliquez sur la flèche vers le bas (▼).
     3. Choisissez l' HealthLake administrateur que vous souhaitez également voir administrateur de Lake Formation.
     4. Choisissez Démarrer.
   - Option 2 : utilisez le volet de navigation (☰).
     1. Choisissez le volet de navigation (☰).
     2. Sous Autorisations, sélectionnez Rôles et tâches d'administration.
     3. Dans la section Administrateurs du lac de données, sélectionnez Choisir les administrateurs.

4. Dans la boîte de dialogue Gérer les administrateurs des lacs de données, cliquez sur la flèche vers le bas (▼).
5. Ensuite, sélectionnez ou recherchez les HealthLake administrateurs, utilisateurs ou rôles que vous souhaitez également voir devenir administrateurs de Lake Formation.
6. Choisissez Save (Enregistrer).

3. Modifiez les paramètres de sécurité par défaut à gérer par Lake Formation. Les ressources du magasin de HealthLake données ne doivent pas être gérées par Lake FormationIAM. Pour effectuer une mise à jour, voir Modifier le modèle d'autorisation par défaut dans le guide du développeur de AWS Lake Formation.

Création d'un magasin de données dans AWS HealthLake

Une fois que vous avez terminéConfiguration des autorisations pour commencer à utiliser AWS HealthLake, vous êtes prêt à créer un magasin de données. Dans AWS HealthLake, vous utilisez un magasin de données pour stocker les données au format HL7 FHIR (R4). Les rubriques de ce chapitre décrivent comment créer un magasin de données.

Pour créer des magasins de données compatibles avec l'analyse et pour autoriser l'accès à ceux-ci dans Athena, ajoutez la politique `AWSLakeFormationDataAdmin` gérée à votre IAM utilisateur, groupe ou rôle. La `AWSLakeFormationDataAdmin` politique vous permet de créer des administrateurs de lacs de données et d'accorder l'accès aux magasins de données d'Athena. Pour plus d'informations sur la définition des autorisations, consultezConfiguration des autorisations pour commencer à utiliser AWS HealthLake.

HealthLake est également intégré à AWS CloudTrail. Vous pouvez l'utiliser CloudTrail pour fournir un enregistrement des actions entreprises par un utilisateur, un rôle ou un AWS service dans HealthLake. CloudTrail capture tous les API appels et actions de console HealthLake sous forme d'événements. Pour en savoir plus, consultez Journalisation des appels AWS HealthLake API avec AWS CloudTrail.

Pour en savoir plus sur les types de ressources Fast Healthcare Interoperability Resources (FHIR) pris en charge par HealthLake, consultezTypes de FHIR ressources pris en charge dans AWS HealthLake.

> ⚠ Compatibilité avec Amazon Athena
>
> HealthLake les magasins de dates créés avant le 14 novembre 2022 ne peuvent pas effectuer de SQL requêtes à l'aide d'Athena. Pour utiliser les fonctionnalités de recherche d'Athena sur votre banque de données préexistante, migrez d'abord les données vers une nouvelle banque de données. Pour en savoir plus sur la migration de banques de données préexistantes, consultez. Migration d'un magasin de données existant pour utiliser Amazon Athena

Après avoir créé un magasin de données, vous pouvez obtenir ses propriétés, y compris son statut, à l'aide des API opérations API_DescribeFHIRDatastoreou API_ListFHIRDatastores.html. Vous pouvez également trouver les statuts des banques de données et d'autres informations sur la page des banques de données de la HealthLake console.

Un magasin de HealthLake données peut avoir les statuts suivants :

- Création : votre banque de données est en cours de création.
- Actif : votre banque de données est active. Vous pouvez importer et exporter des données à partir de celui-ci. Vous pouvez également gérer et rechercher les FHIR ressources que vous avez stockées dans le magasin de données.
- Suppression — Votre banque de données est en cours de suppression.
- Supprimé — Votre banque de données a été supprimée.

Rubriques

Création d'un magasin de données (AWS Management Console)

> **⚠ HealthLake différences entre les consoles**
>
> La HealthLake console ne prend pas en charge la création SMART d'un magasin de données FHIR activé. Pour créer un magasin de données non SMART FHIR activé, vous devez utiliser le AWS CLI ou l'un des magasins AWS pris en chargeSDKS. Pour en savoir plus, consultez Intégration SMART FHIR avec AWS HealthLake. En outre, la console ne fait pas la différence entre les deux types de magasins de données pris en charge HealthLake lorsque vous consultez la page de détails d'un magasin de données individuel.

Pour créer un magasin HealthLake de données

1. Ouvrez la HealthLake console à la https://console.aws.amazon.com//healthlake/maison.
2. Ouvrez le volet de navigation (≡).
3. Choisissez ensuite Data Stores.
4. Ensuite, choisissez Create Data Store.
5. Dans la section Paramètres du magasin de données, pour le nom du magasin de données, spécifiez un nom.
6. (Facultatif) Dans la section des paramètres du magasin de données, pour Précharger les échantillons de données, cochez la case pour précharger les données Synthea.

- Les données Synthea sont un exemple de jeu de données préchargé. Pour de plus amples informations, veuillez consulter Types de données préchargés.

7. Dans la section Chiffrement du magasin de données, choisissez Utiliser la clé AWS détenue (par défaut) ou Choisir une autre AWS KMS clé (avancée).
8. Dans la section Balises - optionnelle, vous pouvez ajouter des balises à votre banque de données.

   - Pour en savoir plus sur le balisage de votre banque de données, consultezAjouter une étiquette à un magasin de données.

9. Ensuite, choisissez Create Data Store. L'état de vos magasins de données est disponible sur la page des magasins de données.

Création d'un magasin de données (AWS CLI et AWS SDKs)

Vous pouvez utiliser les exemples de code suivants pour créer un magasin de HealthLake données.

AWS CLI

L'exemple suivant montre comment utiliser l'`CreateFHIRDatastore`opération avec le AWS CLI. Pour exécuter cet exemple, vous devez installer le AWS CLI. Lorsque vous créez votre banque de données, le chiffrement au repos utilise par défaut une KMS clé AWS appartenant à l'utilisateur, sauf indication contraire. Pour en savoir plus sur le chiffrement, rendez-vous sur REST for HealthLake see,Chiffrement chez REST for AWS HealthLake.

L'exemple est mis en forme pour Unix, Linux et macOS. Pour Windows, remplacez le caractère de continuation Unix de la barre oblique inverse (\) à la fin de chaque ligne par un curseur (). ^

```
aws healthlake create-fhir-datastore \
  --datastore-type-version R4 \
  --preload-data-config PreloadDataType="SYNTHEA" \
  --datastore-name "your-data-store-name"
```

En cas de succès, vous obtenez la JSON réponse suivante. Lorsque votre banque de données est prête à ingérer des données, le statut passe à`ACTIVE`. Pour en savoir plus sur l'importation de données dans votre HealthLake banque de données, consultezImportation de fichiers dans un magasin HealthLake de données.

```
{
    "DatastoreId": "eeb8005725ae22b35b4edbdc68cf2dfd",
 "DatastoreArn": "arn:aws:healthlake:us-west-2:111122223333:datastore/fhir/
eeb8005725ae22b35b4edbdc68cf2dfd",
    "DatastoreStatus": "CREATING",
    "DatastoreEndpoint": "https://healthlake.us-west-2.amazonaws.com/datastore/
eeb8005725ae22b35b4edbdc68cf2dfd/r4/"
}
```

Pour afficher la liste de tous les magasins de données/magasins de données, vous pouvez utiliser l'`ListFHIRDataStore`opération. Vous pouvez également consulter la liste des magasins de données actifs dans la HealthLake console.

Python (boto3)

L'exemple suivant montre comment créer un magasin de HealthLake données à l'aide de cette `create_fhir_datastore` opération. Lorsque vous créez votre banque de données, le chiffrement au repos utilise par défaut une AWS KMS clé AWS appartenant à l'utilisateur, sauf indication contraire. Pour en savoir plus sur le chiffrement, rendez-vous sur REST for HealthLake see,Chiffrement chez REST for AWS HealthLake.

```
import boto3
import logging #built in logging library
from botocore.exceptions import ClientError, ValidationError #specific exception
 ClientError from the boto3 library

def create_healthlake_datastore(DatastoreName=None):
    '''
    :param DatastoreName: the name of the data store, string
    :param:
    :return: True if the data store is created, else False
    '''

    # Create an Amazon Healthlake data store
    # Should we say something about region setting?
    # Should this example have some handling KMS keys

    try:
        if DatastoreName is None:
            healthlake_client = boto3.client('healthlake')
            healthlake_client.create_fhir_datastore(DatastoreTypeVersion='R4')
```

```
        else:
            healthlake_client = boto3.client('healthlake')
            healthlake_client.create_fhir_datastore(DatastoreTypeVersion='R4',
                                                    DatastoreName=DatastoreName)
    except (ClientError, ValidationError) as e:
        logging.error(e)
        return False

    return True

# Run the function above
create_healthlake_datastore(DatastoreName='test-datastore-delete-me-2')
```

Un magasin de données peut avoir l'un des quatre statuts suivants. Utilisez-le `list_fhir_datastores` pour afficher la liste de vos magasins de HealthLake données, quel que soit leur statut. Cet exemple montre comment vous pouvez filtrer en fonction de l'état d'un magasin de données.

```
import boto3

healthlake_client = boto3.client('healthlake')
data_store_list = healthlake_client.list_fhir_datastores(Filter={'DatastoreStatus':
 'ACTIVE'})
print(data_store_list)
```

Pour en savoir plus, consultez `list_fhir_datastore`la documentation de Boto3.

Importation de fichiers dans un magasin HealthLake de données

Une fois que vous avez terminéCréation d'un magasin de données dans AWS HealthLake, vous pouvez importer des fichiers dans le magasin de données à partir d'un bucket Amazon Simple Storage Service (Amazon S3). Pour importer des fichiers, vous devez démarrer une tâche d'importation à l'aide de la HealthLake console ou de l'`StartFHIRImportJob`APIopération.

Lorsque vous créez une tâche d'importation, vous spécifiez l'emplacement de vos données d'entrée dans Amazon S3, un emplacement de compartiment Amazon S3 pour les fichiers journaux de sortie, un IAM rôle qui autorise l' HealthLake accès à vos compartiments et une AWS Key Management Service clé détenue ou AWS détenue par le client. HealthLake utilise cette clé pour chiffrer vos données à l'emplacement source et sera utilisée pour les déchiffrer afin de HealthLake permettre leur importation. Pour plus d'informations sur la configuration des autorisations pour les tâches d'importation, consultezConfiguration des autorisations pour les tâches d'importation. Pour en savoir plus sur la création et l'utilisation de AWS KMS clés, consultez la section Création de clés dans le Guide du développeur du service de gestion des AWS clés.

HealthLake accepte les fichiers d'entrée au format délimité par une nouvelle ligne JSON (`.ndjson`), où chaque ligne est constituée d'une ressource valideFHIR. Vous pouvez utiliser les API opérations `ListFHIRImportJobs` pour décrire `DescribeFHIRImportJob` et répertorier les tâches d'importation en cours.

HealthLake Génère un `manifest.json` fichier pour chaque tâche d'importation. Ce journal décrit à la fois les réussites et les échecs d'une tâche d'importation. HealthLake envoie le fichier dans le compartiment Amazon S3 que vous spécifiez lorsque vous créez une tâche d'importation. Pour de plus amples informations, veuillez consulter JSONFichier manifeste.

Vous pouvez mettre en file d'attente des tâches d'importation ou d'exportation. Ces tâches d'importation ou d'exportation asynchrones sont traitées selon le mode « premier entré, premier sorti ». FIFO Vous pouvez créer, lire, mettre à jour ou supprimer FHIR des ressources pendant qu'une tâche d'importation ou d'exportation est en cours.

Après avoir renseigné un magasin de données avec des données préchargées ou importé des données, vous pouvez commencer à interroger votre magasin de données à l'aide d'Amazon SQL Athena. Pour de plus amples informations, veuillez consulter Interrogez les magasins de AWS HealthLake données SQL à l'aide d'Amazon Athena.

Rubriques

Configuration des autorisations pour les tâches d'importation

Avant d'importer des fichiers dans un magasin de données, vous devez HealthLake autoriser l'accès à vos compartiments d'entrée et de sortie dans Amazon S3. Pour accorder HealthLake l'accès, vous créez un rôle de IAM service pour HealthLake, vous ajoutez une politique de confiance au rôle pour accorder les autorisations d' HealthLake assumer le rôle, et vous attachez une politique d'autorisation au rôle qui lui accorde l'accès à vos compartiments Amazon S3.

Lorsque vous créez une tâche d'importation, vous spécifiez le nom de ressource Amazon (ARN) de ce rôle pour le`DataAccessRoleArn`. Pour plus d'informations sur IAM les rôles et les politiques de confiance, consultez la section IAMRôles.

Après avoir configuré l'autorisation, vous êtes prêt à importer des fichiers dans votre banque de données à l'aide d'une tâche d'importation. Pour de plus amples informations, veuillez consulter Démarrage d'une tâche d'importation dans HealthLake.

Pour configurer les autorisations d'importation

1. Si ce n'est pas déjà fait, créez un compartiment Amazon S3 de destination pour les fichiers journaux de sortie. Le compartiment Amazon S3 doit se trouver dans la même AWS région que le service, et le blocage de l'accès public doit être activé pour toutes les options. Pour en savoir plus, consultez Utiliser Amazon S3 pour bloquer l'accès public. Une KMS clé appartenant à Amazon ou au client doit également être utilisée pour le chiffrement. Pour en savoir plus sur l'utilisation KMS des clés, consultez Amazon Key Management Service.
2. Créez un rôle de service d'accès aux données pour HealthLake et autorisez le HealthLake service à l'assumer conformément à la politique de confiance suivante. HealthLake l'utilise pour écrire le bucket Amazon S3 de sortie.

```
{
    "Version": "2012-10-17",
    "Statement": [{
```

```
        "Effect": "Allow",
        "Principal": {
            "Service": ["healthlake.amazonaws.com"]
        },
        "Action": "sts:AssumeRole",
        "Condition": {
            "StringEquals": {
                "aws:SourceAccount": "your-account-id"
            },
            "ArnEquals": {
                "aws:SourceArn": "arn:aws:healthlake:us-west-2:account:datastore/
fhir/data store ID"
            }
        }
    }]
}
```

3. Ajoutez une politique d'autorisation au rôle d'accès aux données qui lui permet d'accéder au compartiment Amazon S3. `amzn-s3-demo-bucket`Remplacez-le par le nom de votre compartiment.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Action": [
            "s3:ListBucket",
            "s3:GetBucketPublicAccessBlock",
            "s3:GetEncryptionConfiguration"
        ],
        "Resource": [
            "arn:aws:s3:::amzn-s3-demo-source-bucket"
        ],
        "Effect": "Allow"
    },
    {
        "Action": [
            "s3:PutObject"
        ],
        "Resource": [
            "arn:aws:s3:::amzn-s3-demo-logging-bucket/*"
        ],
        "Effect": "Allow"
    },
```

```
    {
        "Action": [
            "kms:DescribeKey",
            "kms:GenerateDataKey*"
        ],
        "Resource": [
            "arn:aws:kms:us-east-1:012345678910:key/d330e7fc-b56c-4216-a250-
f4c43ef46e83"
        ],
        "Effect": "Allow"
    }]
}
```

Démarrage d'une tâche d'importation dans HealthLake

Après avoir créé un magasin de données et défini les autorisations pour les tâches d'importation (Configuration des autorisations pour les tâches d'importation), vous pouvez commencer à importer des fichiers avec une tâche d'importation. Vous pouvez démarrer une tâche d'importation à l'aide de la AWS HealthLake console ou de l' AWS HealthLake importationAPI, start-fhir-import-jobAPI.

Rubriques

- Importation de fichiers avec API opérations
- Démarrage d'une tâche d'importation (console)

Importation de fichiers avec API opérations

Prérequis

Lorsque vous utilisez les AWS HealthLake API opérations, vous devez d'abord créer une politique AWS Identity and Access Management (IAM) et l'associer à un IAM rôle. Pour en savoir plus sur IAM les rôles et les politiques de confiance, consultez la section IAMPolitiques et autorisations. Les clients doivent également utiliser une KMS clé pour le chiffrement. Pour en savoir plus sur l'utilisation KMS des clés, consultez Amazon Key Management Service.

Pour importer des fichiers (API), procédez comme suit.

1. Chargez vos données dans un compartiment Amazon S3.

2. Utilisez l'`start-fhir-import-job API`APIopération. Lorsque vous démarrez la tâche, spécifiez le nom du compartiment Amazon S3 qui contient les fichiers d'entrée, la KMS clé que vous souhaitez utiliser pour le chiffrement et la configuration des données de sortie.
3. Pour en savoir plus sur une tâche FHIR d'importation, utilisez l'describe-fhir-import-jobopération pour obtenir l'ID, le nomARN, l'heure de début, l'heure de fin et le statut actuel de la tâche. Permet list-fhir-import-jobd'afficher toutes les tâches d'importation et leur statut.

Démarrage d'une tâche d'importation (console)

Pour importer des fichiers avec la console, vous chargez vos données dans un compartiment Amazon S3,

Pour importer des fichiers, procédez comme suit.

1. Chargez vos données dans un compartiment Amazon S3.
2. Ouvrez la HealthLake console à la https://console.aws.amazon.com//healthlake/maison.
3. Accédez à la page de détails du magasin de données de votre magasin de données et choisissez Importer.
4. Spécifiez votre compartiment Amazon S3 et créez ou identifiez le IAM rôle et la KMS clé que vous souhaitez utiliser.
5. Choisissez Import data (Importer les données).

JSONFichier manifeste

HealthLake Génère un `manifest.json` fichier pour chaque tâche d'importation. HealthLake envoie le fichier dans le compartiment Amazon S3 que vous spécifiez lorsque vous créez une tâche d'importation.

Le `manifest.json` fichier décrit à la fois les réussites et les échecs d'une tâche d'importation. Les fichiers journaux sont organisés en deux dossiers, nommés `SUCCESS` et`FAILURE`. Un fichier de sortie peut contenir des informations sensibles. Par conséquent, lorsque vous créez une tâche d'importation, vous devez fournir à la fois un compartiment de sortie Amazon S3 et une AWS KMS clé de chiffrement.

Voici un exemple de `manifest.json` fichier de sortie. Nous vous recommandons d'utiliser ce fichier comme première étape pour résoudre les problèmes liés à l'échec d'une tâche d'importation. Il fournit des détails sur chaque fichier et explique la cause de l'échec de la tâche d'importation.

```
  {
  "inputDataConfig": {
    "s3Uri": "s3://amzn-s3-demo-source-bucket/healthlake-input/invalidInput/"
  },
  "outputDataConfig": {
    "s3Uri": "s3://amzn-s3-demo-logging-bucket/32839038a2f47f17c2fe0f53f0c3a0ba-
FHIR_IMPORT-19dd7bb7bcc8ee12a09bf6d322744a3d/",
    "encryptionKeyID": "arn:aws:kms:us-west-2:123456789012:key/fbbbfee3-20b3-42a5-a99d-
c48c655ed545"
  },
  "successOutput": {
    "successOutputS3Uri": "s3://amzn-s3-demo-logging-
bucket/32839038a2f47f17c2fe0f53f0c3a0ba-FHIR_IMPORT-19dd7bb7bcc8ee12a09bf6d322744a3d/
SUCCESS/"
  },
  "failureOutput": {
    "failureOutputS3Uri": "s3://amzn-s3-demo-logging-
bucket/32839038a2f47f17c2fe0f53f0c3a0ba-FHIR_IMPORT-19dd7bb7bcc8ee12a09bf6d322744a3d/
FAILURE/"
  },
  "numberOfScannedFiles": 1,
  "numberOfFilesImported": 1,
  "sizeOfScannedFilesInMB": 0.023627,
  "sizeOfDataImportedSuccessfullyInMB": 0.011232,
  "numberOfResourcesScanned": 9,
  "numberOfResourcesImportedSuccessfully": 4,
  "numberOfResourcesWithCustomerError": 5,
  "numberOfResourcesWithServerError": 0
}
```

Exemple : Démarrage et surveillance des tâches d'importation à l'aide du AWS CLI

L'exemple suivant montre comment utiliser le AWS Command Line Interface pour démarrer et surveiller une tâche d'importation. Vous pouvez également utiliser start-fhir-import-job API.

```
aws healthlake start-fhir-import-job \
--input-data-config S3Uri=s3://amzn-s3-demo-source-bucket/inputFolder/ \
--datastore-id (Datastore ID) \
--data-access-role-arn "arn:aws:iam::012345678910:role/DataAccessRole" \
--job-output-data-config '{"S3Configuration": {"S3Uri":"s3://amzn-s3-demo-logging-bucket/healthlake-output","KmsKeyId":"arn:aws:kms:us-east-1:012345678910:key/d330e7fc-b56c-4216-a250-f4c43ef46e83"}}' \
--region us-east-1
```

Lorsque le travail d'importation commence, vous recevrez la confirmation suivante.

```
{
 "JobId": "8a4077553e9a485ad889c1a89c7541f0",
 "JobStatus": "SUBMITTED",
 "DatastoreId": "32839038a2f47f17c2fe0f53f0c3a0ba"
}
```

Pour surveiller l'état d'une tâche d'importation ou pour connaître ses propriétés de configuration, utilisez la AWS CLI commande describe-fhir-import-jobAPIou, comme indiqué dans l'exemple suivant.

```
    aws healthlake describe-fhir-import-job \
    --datastore-id (Datastore ID) \
    --job-id c145fbb27b192af392f8ce6e7838e34f \
    --region us-east-1
```

Vous recevez les informations suivantes en réponse.

```
{
    "ImportJobProperties": {
    "InputDataConfig": {
    "S3Uri": "s3://amzn-s3-demo-source-bucket/(Prefix Name)/"
    },
    "DataAccessRoleArn": "arn:aws:iam::(AWS Account ID):role/(Role Name)",
    "JobStatus": "COMPLETED",
    "JobId": "c145fbb27b192af392f8ce6e7838e34f",
```

```
    "SubmitTime": 1606272542.161,
    "EndTime": 1606272609.497,
    "DatastoreId": "(Datastore ID)"
    }
}
```

Pour afficher la liste de toutes les tâches d'importation, utilisez la AWS CLI commande list-fhir-import-jobsAPIou, comme indiqué dans l'exemple suivant. Vous pouvez ajouter un ou plusieurs filtres pour limiter les résultats.

```
    aws healthlake list-fhir-import-jobs\
    --datastore-id (Datastore ID) \
    --submitted-before (DATE like 2024-10-13T19:00:00Z)\
    --submitted-after (DATE like 2020-10-13T19:00:00Z )\
    --job-name "FHIR-IMPORT" \
    --job-status SUBMITTED  \
    --max-results (Integer between 1 and 500)
```

Vous recevez les informations suivantes en réponse.

```
{
    "ImportJobProperties": {
    "OutputDataConfig": {
    "S3Uri": "s3://(Bucket Name)/(Prefix Name)/",
    "S3Configuration": {
     "S3Uri": "s3://(Bucket Name)/(Prefix Name)/",
     "KmsKeyId" : "(KmsKey Id)"
     },
    },
    "DataAccessRoleArn": "arn:aws:iam::(AWS Account ID):role/(Role Name)",
    "JobStatus": "COMPLETED",
    "JobId": "c145fbb27b192af392f8ce6e7838e34f",
    "JobName" "FHIR-IMPORT",
    "SubmitTime": 1606272542.161,
    "EndTime": 1606272609.497,
    "DatastoreId": "(Datastore ID)"
    }
}
"NextToken": String
```

Exportation de fichiers depuis un magasin HealthLake de données

Après avoir créé un magasin de données et importé des données (ou si vous utilisez des exemples de données préchargés), vous pouvez exporter les données vers un compartiment Amazon S3. Pour exporter des données depuis votre HealthLake banque de données, effectuez les opérations suivantes.

- Faites une demande d'exportation à l'aide de l'`StartFHIRExportJob`APIopération à l'aide des AWS SDKs touches et HealthLake.
  - Cette opération permet uniquement de faire une demande d'exportation à l'échelle du système.
- Faites une demande d'exportation en utilisant la `export` syntaxe utilisant le HealthLake FHIR RESTAPI.
  - Cette opération permet de faire des demandes d'exportation à l'échelle du système, du patient et du groupe. Vous pouvez également appliquer des paramètres pour filtrer davantage les données de la demande d'exportation.

> **Important**
>
> HealthLake SDKles demandes d'exportation utilisant l'`StartFHIRExportJob`APIopération et les demandes FHIR REST API d'exportation utilisant l'`StartFHIRExportJobWithPost`APIopération ont IAM des actions distinctes. Les autorisations d'autorisation/de refus peuvent être traitées `StartFHIRExportJob` séparément pour chaque IAM action`StartFHIRExportJobWithPost`, qu'il s'agisse d'une FHIR REST API exportation ou d'une exportation avec. SDK Si vous souhaitez restreindre à la fois les FHIR REST API exportations SDK et les exportations, assurez-vous de refuser les autorisations pour chaque IAM action.

Ces deux opérations ne prennent en charge que l'exportation de vos fichiers vers un compartiment Amazon S3 (S3). Tous les fichiers de votre HealthLake banque de données sont exportés sous forme de fichiers délimités par des nouvelles lignes JSON (`.ndjson`), chaque ligne contenant une ressource valideFHIR.

Ces deux opérations nécessitent un rôle de service. Il HealthLake doit être défini comme le principal du service, et vous devez définir un compartiment Amazon Simple Storage Service (S3) dans lequel vous souhaitez exporter vos fichiers. Pour en savoir plus, consultez Configuration des autorisations pour les tâches d'exportation.

Vous pouvez mettre en file d'attente des tâches d'importation ou d'exportation. Ces tâches d'importation ou d'exportation asynchrones sont traitées selon le mode « premier entré, premier sorti ». FIFO Vous pouvez créer, lire, mettre à jour ou supprimer FHIR des ressources pendant qu'une tâche d'importation ou d'exportation est en cours.

Pour exporter des fichiers depuis votre HealthLake banque de données, consultez les sections suivantes.

- Configuration des autorisations pour les tâches d'exportation
- Exportation de fichiers depuis votre banque de données à l'aide de la HealthLake console ou AWS SDKs
- Exportation de données depuis votre HealthLake banque de données avec FHIR REST API opérations

Configuration des autorisations pour les tâches d'exportation

Avant d'exporter des fichiers depuis un magasin de données, vous devez HealthLake autoriser l'accès à votre compartiment de sortie dans Amazon S3. Pour accorder HealthLake l'accès, vous créez un rôle de IAM service pour HealthLake, vous ajoutez une politique de confiance au rôle pour accorder les autorisations d' HealthLake assumer le rôle, et vous attachez une politique d'autorisation au rôle qui lui accorde l'accès à votre compartiment Amazon S3.

Si vous avez déjà créé un rôle pour HealthLake inConfiguration des autorisations pour les tâches d'importation, vous pouvez le réutiliser et lui accorder les autorisations supplémentaires pour votre compartiment Amazon S3 d'exportation répertoriées dans cette rubrique. Pour en savoir plus sur IAM les rôles et les politiques de confiance, consultez la section IAMPolitiques et autorisations.

> **Important**
>
> HealthLake SDKles demandes d'exportation utilisant l'`StartFHIRExportJob`APIopération et les demandes FHIR REST API d'exportation utilisant l'`StartFHIRExportJobWithPost`APIopération ont IAM des actions distinctes. Les autorisations d'autorisation/de refus peuvent être traitées `StartFHIRExportJob`

séparément pour chaque IAM action`StartFHIRExportJobWithPost`, qu'il s'agisse d'une FHIR REST API exportation ou d'une exportation avec. SDK Si vous souhaitez restreindre à la fois les FHIR REST API exportations SDK et les exportations, assurez-vous de refuser les autorisations pour chaque IAM action. Si vous accordez aux utilisateurs un accès complet à HealthLake, aucune modification des autorisations IAM utilisateur n'est requise.

L'utilisateur ou le rôle qui définit les autorisations doit être autorisé à créer des rôles, à créer des politiques et à associer des politiques aux rôles. La IAM politique suivante accorde ces autorisations.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": ["iam:CreateRole", "iam:CreatePolicy", "iam:AttachRolePolicy"],
    "Effect": "Allow",
    "Resource": "*"
    }, {
    "Action": "iam:PassRole"
    "Effect": "Allow",
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "iam:PassedToService": "healthlake.amazonaws.com"
      }
    }
  }]
}
```

Pour configurer les autorisations d'exportation

1. Si ce n'est pas déjà fait, créez un compartiment Amazon S3 de destination pour les données que vous allez exporter depuis votre magasin de données. Le compartiment Amazon S3 doit se trouver dans la même AWS région que le service, et le blocage de l'accès public doit être activé pour toutes les options. Pour en savoir plus, consultez Utiliser Amazon S3 pour bloquer l'accès public. Une KMS clé appartenant à Amazon ou au client doit également être utilisée pour le chiffrement. Pour en savoir plus sur l'utilisation KMS des clés, consultez Amazon Key Management Service.
2. Si ce n'est pas déjà fait, créez un rôle de service d'accès aux données HealthLake et autorisez le HealthLake service à l'assumer conformément à la politique de confiance suivante. HealthLake

l'utilise pour écrire le bucket Amazon S3 de sortie. Si vous en avez déjà créé unConfiguration des autorisations pour les tâches d'importation, vous pouvez le réutiliser et lui accorder des autorisations pour votre compartiment Amazon S3 à l'étape suivante.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Principal": {
            "Service": ["healthlake.amazonaws.com"]
        },
        "Action": "sts:AssumeRole",
        "Condition": {
            "StringEquals": {
                "aws:SourceAccount": "your-account-id"
            },
            "ArnEquals": {
                "aws:SourceArn": "arn:aws:healthlake:us-west-2:account:datastore/
fhir/data store ID"
            }
        }
    }]
}
```

3. Ajoutez une politique d'autorisation au rôle d'accès aux données qui lui permet d'accéder à votre compartiment Amazon S3 de sortie. `amzn-s3-demo-bucket`Remplacez-le par le nom de votre compartiment.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Action": [
            "s3:ListBucket",
            "s3:GetBucketPublicAccessBlock",
            "s3:GetEncryptionConfiguration"
        ],
        "Resource": [
            "arn:aws:s3:::amzn-s3-demo-source-bucket"
        ],
        "Effect": "Allow"
    },
    {
```

```
        "Action": [
            "s3:PutObject"
        ],
        "Resource": [
            "arn:aws:s3:::amzn-s3-demo-logging-bucket/*"
        ],
        "Effect": "Allow"
    },
    {
        "Action": [
            "kms:DescribeKey",
            "kms:GenerateDataKey*"
        ],
        "Resource": [
            "arn:aws:kms:us-east-1:012345678910:key/d330e7fc-b56c-4216-a250-
f4c43ef46e83"
        ],
        "Effect": "Allow"
    }]
}
```

Exportation de fichiers depuis votre banque de données à l'aide de la HealthLake console ou AWS SDKs

Une fois que vous avez terminéConfiguration des autorisations pour les tâches d'exportation, vous pouvez exporter des fichiers depuis votre magasin de données vers un bucket Amazon Simple Storage Service (Amazon S3). Pour exporter des fichiers depuis un magasin de données, vous devez démarrer une tâche d'exportation dans HealthLake. Une tâche d'exportation exporte des fichiers depuis votre banque de données au format newline délimité JSON (`.ndjson`), où chaque ligne correspond à une ressource valideFHIR. Lorsque vous démarrez une tâche d'exportation, vous devez spécifier une AWS KMS clé pour le chiffrement. Pour en savoir plus sur la création d'une KMS clé, consultez la section Création de clés dans le Guide du développeur du service de gestion des AWS clés.

Les rubriques suivantes expliquent comment démarrer une tâche d'exportation avec la AWS HealthLake console et AWS SDKs avec l'start-fhir-export-jobAPIopération.

Rubriques

- Exportation de fichiers depuis votre banque de données (console)

Exportation de fichiers depuis votre banque de données (console)

Pour exporter des fichiers (console), procédez comme suit.

1. Créez un compartiment S3 de sortie dans la même région que HealthLake.
2. Pour démarrer une nouvelle tâche d'exportation, identifiez le compartiment Amazon S3 de sortie et créez ou identifiez le IAM rôle que vous souhaitez utiliser. Pour en savoir plus sur IAM les rôles et les politiques de confiance, consultez la section IAMrôles. Utilisez également un chiffrement par KMS clé. Pour en savoir plus sur l'utilisation KMS des clés, consultez Amazon Key Management Service.
3. Pour voir le statut de votre tâche d'exportation, utilisez `ListFHIRExportJobs` API operation.

Exportation de fichiers depuis votre banque de données (AWS SDKs)

Pour exporter des fichiers depuis votre banque de données avec le AWS SDKs, utilisez l'start-fhir-export-jobopération. Le code suivant montre comment démarrer une tâche d'exportation avec le SDK for Python (Boto3).

```
import boto3

client = boto3.client('healthlake')

response = client.start_fhir_export_job(
    JobName='job name',
    OutputDataConfig={
        'S3Configuration': {
            'S3Uri': 's3://amzn-s3-demo-bucket/output-folder',
            'KmsKeyId': 'arn:aws:kms:us-west-2:account-number:key/AWS KMS key ID'
        }
    },
    DatastoreId='data store ID',
    DataAccessRoleArn='role ARN',
)
print(response['JobStatus'])
```

Pour obtenir l'ID, le nomARN, l'heure de début, l'heure de fin et le statut actuel d'une tâche d'FHIRexportation, utilisez describe-fhir-export-job. Permet list-fhir-export-jobsde répertorier toutes les tâches d'exportation et leur statut.

Le code suivant montre comment obtenir les propriétés d'une tâche d'exportation spécifique avec SDK for Python (Boto3).

```
import boto3

client = boto3.client('healthlake')

describe_response = client.describe_fhir_export_job(
    DatastoreId=datastoreId,
    JobId=jobId
)
print(describe_response['ExportJobProperties'])
```

Exportation de données depuis votre HealthLake banque de données avec FHIR REST API opérations

Une fois que vous avez terminéConfiguration des autorisations pour les tâches d'exportation, vous pouvez exporter les données de votre HealthLake banque de données avec FHIR REST API les opérations. Pour effectuer une demande d'exportation à l'aide de FHIR RESTAPI, vous devez disposer IAM d'un utilisateur, d'un groupe ou d'un rôle doté des autorisations requises, le spécifier dans le `$export` cadre de la `POST` demande et inclure les paramètres de la demande dans le corps de votre demande. Selon les FHIR spécifications, le FHIR serveur doit prendre en charge les `GET` demandes et peut prendre en charge les `POST` demandes. Afin de prendre en charge des paramètres supplémentaires, un corps est nécessaire pour démarrer l'exportation. Il HealthLake prend donc en charge les POST demandes.

> **Important**
>
> HealthLake les banques de données créées avant le 1er juin 2023 ne prennent en charge que les demandes de travail d'exportation FHIR REST API basées sur les exportations à l'échelle du système.
>
> HealthLake les magasins de données créés avant le 1er juin 2023 ne permettent pas d'obtenir le statut d'une exportation à l'aide d'une `GET` demande sur le point de terminaison d'un magasin de données.

Toutes les demandes d'exportation que vous effectuez à l'aide du FHIR REST API sont renvoyées au `ndjson` format et exportées vers un compartiment Amazon S3. Chaque objet S3 ne contiendra qu'un seul type de FHIR ressource.

Vous pouvez mettre en file d'attente les demandes d'exportation conformément aux quotas du AWS compte. Pour en savoir plus sur les Quotas de Service associés à HealthLake, consultezAWS HealthLake points de terminaison et quotas.

HealthLake prend en charge les trois types suivants de demandes de terminaux d'exportation en masse.

Type	Descriptions	Syntaxe
Exportation du système	Exportez toutes les données depuis le HealthLake FHIR serveur.	`POST https://healthlake. your-region .amazonaws.com/datastore/ your-datastore-id /r4/$export`
Tous les patients	Exportez toutes les données relatives à tous les patients, y compris les types de ressources associés au type de ressource Patient.	`POST https://healthlake. your-region .amazonaws.com/datastore/ your-datastore-id /r4/Patient/$export`
Groupe de patients	Exportez toutes les données relatives à un groupe de patients spécifié par un identifiant de groupe.	`POST https://healthlake. your-region .amazonaws.com/datastore/ your-datastore-id /r4/Group/ ID/$export`

Avant de commencer

Répondez aux exigences suivantes pour effectuer une demande d'exportation à l'aide du FHIR REST API for HealthLake.

- Vous devez avoir configuré un utilisateur, un groupe ou un rôle disposant des autorisations nécessaires pour effectuer la demande d'exportation. Pour en savoir plus, consultez Autoriser une demande `export`.
- Vous devez avoir créé un rôle de service qui accorde l' HealthLake accès au compartiment Amazon S3 vers lequel vous souhaitez que vos données soient exportées. Le rôle de service

doit également être spécifié HealthLake comme principal de service. Pour plus d'informations sur la configuration des autorisations, consultezConfiguration des autorisations pour les tâches d'exportation.

Autoriser une demande **`export`**

Pour effectuer une demande d'exportation réussie à l'aide du FHIR RESTAPI, autorisez votre utilisateur, votre groupe ou votre rôle en utilisant soit IAM OAuth2 .0. Vous devez également avoir un rôle de service.

Autoriser une demande en utilisant IAM

Lorsque vous faites une `$export` demande, l'utilisateur, le groupe ou le rôle doit avoir `StartFHIRExportJobWithPost` des `CancelFHIRExportJobWithDelete` IAM actions incluses dans la politique. `DescribeFHIRExportJobWithGet`

> **Important**
>
> HealthLake SDKles demandes d'exportation utilisant l'`StartFHIRExportJob`APIopération et les demandes FHIR REST API d'exportation utilisant l'`StartFHIRExportJobWithPost`APIopération ont IAM des actions distinctes. Les autorisations d'autorisation/de refus peuvent être traitées `StartFHIRExportJob` séparément pour chaque IAM action`StartFHIRExportJobWithPost`, qu'il s'agisse d'une FHIR REST API exportation ou d'une exportation avec. SDK Si vous souhaitez restreindre à la fois les FHIR REST API exportations SDK et les exportations, assurez-vous de refuser les autorisations pour chaque IAM action.

Autoriser une demande à l'aide de SMART on FHIR (OAuth2.0)

Lorsque vous faites une `$export` demande SMART sur un magasin de HealthLake données FHIR activé, les étendues appropriées doivent vous être attribuées. Pour en savoir plus sur les étendues prises en charge, consultezHealthLake portées spécifiques aux FHIR ressources du magasin de données.

Faire une **`export`** demande

Cette section décrit les étapes obligatoires que vous devez suivre lorsque vous effectuez une demande d'exportation à l'aide du FHIR RESTAPI.

Pour éviter des frais accidentels sur votre AWS compte, nous vous recommandons de tester vos demandes en effectuant une POST demande sans fournir de `export` syntaxe.

Pour faire la demande, vous devez effectuer les opérations suivantes :

1. Spécifiez `export` dans la POST demande un point URL de terminaison pris en charge.
2. Spécifiez les paramètres d'en-tête requis.
3. Spécifiez un corps de demande qui définit les paramètres requis.

Étape 1 : Spécifiez **`export`** dans la **`POST`** demande un point URL de terminaison pris en charge

HealthLake prend en charge trois types de demandes d'exportation groupées pour les terminaux. Pour effectuer une demande d'exportation groupée, vous devez effectuer une demande POST basée sur l'un des trois points de terminaison pris en charge. Les exemples suivants montrent comment spécifier `export` dans la demandeURL.

- `POST https://healthlake.`**`your-region`**`.amazonaws.com/datastore/`**`your-datastore-id`**`/r4/$export`
- `POST https://healthlake.`**`your-region`**`.amazonaws.com/datastore/`**`your-datastore-id`**`/r4/Patient/$export`
- `POST https://healthlake.`**`your-region`**`.amazonaws.com/datastore/`**`your-datastore-id`**`/r4/Group/`**`ID`**`/$export`

Dans cette chaîne de POST requête, vous pouvez utiliser les paramètres de recherche pris en charge suivants.

Paramètres de recherche pris en charge

HealthLake prend en charge les modificateurs de recherche suivants dans les demandes d'exportation groupées.

Ces exemples incluent des caractères spéciaux qui doivent être codés avant de soumettre votre demande.

Nom	Obligatoire ?	Description	Exemple
`_outputFormat`	Non	Format des fichiers de données en masse demandés à générer. Les valeurs acceptées sont `application/fhir+ndjson` `application/ndjson` ,`ndjson`.	
`_type`	Non	Chaîne de types de FHIR ressources séparés par des virgules que vous souhaitez inclure dans votre tâche d'exportation. Nous vous recommandons de l'inclure, `_type` car cela peut avoir une incidence financière lorsque toutes les ressources sont exportées.	`&_type=MedicationStatement,Observation`
`_since`	Non	Types de ressources modifiés le jour ou après l'horodatage. Si un type de ressource n'a pas d'heure de dernière mise à jour, il sera inclus dans votre réponse.	`&_since=2024-05-09T00%3A00%3A00Z`

Étape 2 : Spécifier les paramètres d'en-tête requis

Pour effectuer une demande d'exportation à l'aide du FHIR RESTAPI, vous devez spécifier les deux paramètres d'en-tête suivants.

- Type de contenu : `application/fhir+json`
- Préférez : `respond-async`

Ensuite, vous devez spécifier les éléments requis dans le corps de la demande.

Étape 3 : Spécifiez un corps de demande qui définit les paramètres requis.

La demande d'exportation nécessite également un corps au `JSON` format. Le corps peut inclure les paramètres suivants.

Clé	Obligatoire ?	Description	Valeur
DataAccessRoleArn	Oui	Et ARN d'un rôle HealthLake de service. Le rôle de service utilisé doit être spécifié HealthLake comme principal de service.	`arn:aws:iam:: 444455556666 :role/your-healthlake-service-role`
JobName	Non	Nom de la demande d'exportation.	**your-export-job-name**
S3Uri	Oui	Partie d'une OutputDataConfig clé. Le S3 URI du compartiment de destination dans lequel vos données exportées seront téléchargées.	`s3://DOC-EXAMPLE-DESTINATION-BUCKET/ EXPORT-JOB /`

Clé	Obligatoire ?	Description	Valeur
KmsKeyId	Oui	Partie d'une OutputDataConfig clé. La ARN AWS KMS clé utilisée pour sécuriser le compartiment Amazon S3.	arn:aws:kms: **region-of-bucket:123456789012** :key/**1234abcd-12ab-34cd-56ef-1234567890ab**

Example — Le corps d'une demande d'exportation effectuée à l'aide du FHIR REST API

Pour effectuer une demande d'exportation à l'aide du FHIR RESTAPI, vous devez spécifier un corps, comme indiqué ci-dessous.

```
{
  "DataAccessRoleArn": "arn:aws:iam::444455556666:role/your-healthlake-service-role",
  "JobName": "your-export-job",
  "OutputDataConfig": {
    "S3Configuration": {
      "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/EXPORT-JOB",
      "KmsKeyId": "arn:aws:kms:region-of-
bucket:444455556666:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    }
  }
}
```

Lorsque votre demande sera acceptée, vous recevrez la réponse suivante.

En-tête de réponse

```
content-location: https://healthlake.your-region.amazonaws.com/datastore/your-
datastore-id/r4/export/your-export-request-job-id
```

Organisme de réponse

```
{
  "datastoreId": "your-data-store-id",
  "jobStatus": "SUBMITTED",
  "jobId": "your-export-request-job-id"
```

```
}
```

Gestion de votre demande d'exportation

Après avoir effectué une demande d'exportation réussie, vous pouvez gérer cette demande en utilisant `export` pour décrire le statut d'une demande d'exportation en cours et `export` pour annuler une demande d'exportation en cours.

Lorsque vous annulez une demande d'exportation à l'aide du RESTAPI, vous ne serez facturée que pour la partie des données exportées jusqu'au moment où vous avez soumis la demande d'annulation.

Les rubriques suivantes décrivent comment vous pouvez obtenir le statut d'une demande d'exportation en cours ou l'annuler.

Annulation d'une demande d'exportation

Pour annuler une demande d'exportation, faites une `DELETE` demande et indiquez l'ID de tâche dans la demandeURL.

```
DELETE https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/
export/your-export-request-job-id
```

Lorsque votre demande est acceptée, vous recevez ce qui suit.

```
{
  "exportJobProperties": {
    "jobId": "your-original-export-request-job-id",
    "jobStatus": "CANCEL_SUBMITTED",
    "datastoreId": "your-data-store-id"
  }
}
```

Lorsque votre demande n'aboutit pas, vous recevez ce qui suit.

```
{
  "resourceType": "OperationOutcome",
  "issue": [
    {
      "severity": "error",
      "code": "not-supported",
```

```
      "diagnostics": "Interaction not supported."
    }
  ]
}
```

Décrire une demande d'exportation

Pour connaître le statut d'une demande d'exportation, faites une `GET` demande en utilisant `export` et votre**`export-request-job-id`**.

```
GET https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/
export/your-export-request-id
```

La JSON réponse contiendra un `ExportJobProperties` objet. Il peut contenir les paires clé:valeur suivantes.

Nom	Obligatoire ?	Description	Valeur
DataAccessRoleArn	Non	Et ARN d'un rôle HealthLake de service. Le rôle de service utilisé doit être spécifié HealthLake comme principal de service.	`arn:aws:iam:: 444455556666 :role/your-healthlake-service-role`
SubmitTime	Non	Date à laquelle une tâche d'exportation a été soumise.	`Apr 21, 2023 5:58:02`
EndTime	Non	Heure à laquelle une tâche d'exportation a été terminée.	`Apr 21, 2023 6:00:08 PM`
JobName	Non	Nom de la demande d'exportation.	**`your-export-job-name`**
JobStatus	Non		Les valeurs valides sont :

Nom	Obligatoire ?	Description	Valeur
			`SUBMITTED | IN_PROGRESS | COMPLETED_WITH_ERRORS | COMPLETED | FAILED`
S3Uri	Oui	Partie d'un OutputDataConfigobjet. L'Amazon S3 URI du compartiment de destination dans lequel vos données exportées seront téléchargées.	`s3://DOC-EXAMPLE-DESTINATION-BUCKET/` **`EXPORT-JOB /`**
KmsKeyId	Oui	Partie d'un OutputDataConfigobjet. La ARN AWS KMS clé utilisée pour sécuriser le compartiment Amazon S3.	`arn:aws:kms:` **`region-of-bucket:123456789012`** `:key/`**`1234abcd-12ab-34cd-56ef-1234567890ab`**

Example : corps d'une description de la demande d'exportation effectuée à l'aide du FHIR REST API

En cas de succès, vous obtiendrez la JSON réponse suivante.

```
{
  "exportJobProperties": {
    "jobId": "your-export-request-id",
    "JobName": "your-export-job",
    "jobStatus": "SUBMITTED",
    "submitTime": "Apr 21, 2023 5:58:02 PM",
    "endTime": "Apr 21, 2023 6:00:08 PM",
    "datastoreId": "your-data-store-id",
    "outputDataConfig": {
```

```
      "s3Configuration": {
        "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/EXPORT-JOB",
        "KmsKeyId": "arn:aws:kms:region-of-
bucket:444455556666:key/1234abcd-12ab-34cd-56ef-1234567890ab""
      }
    },
    "DataAccessRoleArn": "arn:aws:iam::444455556666:role/your-healthlake-service-role",
  }
}
```

Suppression d'un magasin de données dans HealthLake

La suppression d'un magasin de données est une opération asynchrone. Une fois lancé, le statut passe à Supprimer. Un magasin de données conserve le statut Suppression jusqu'à ce que toutes les FHIR données du magasin de données et de l'infrastructure sous-jacente nécessaires soient également supprimées.

Une fois les données et l'infrastructure supprimées, le statut de votre magasin de HealthLake données passe à Supprimé. Après la suppression, les informations concernant vos magasins de données ne sont disponibles qu'en utilisant les `ListFHIRDataStores` opérations `DescribeFHIRDataStore` et pendant sept jours. Après sept jours, le magasin de données supprimé n'apparaîtra pas dans les résultats.

Pour supprimer correctement un magasin de données, l'IAMaction de l'utilisateur, du groupe ou du rôle à l'origine de la demande doit être `glue:DeleteDatabase` ajoutée à sa IAM politique. Cette IAM action n'est pas incluse dans la politique AWS gérée,`AmazonHealthLakeFullAccess`.

Vous pouvez supprimer un magasin de données avec le AWS Management Console AWS SDKs, ou le AWS CLI.

Rubriques

- Supprimer un magasin de données (console)
- Supprimer un magasin de données (AWS SDKset AWS CLI)

Supprimer un magasin de données (console)

Pour supprimer un magasin de données à l'aide de la console, choisissez votre magasin de données sur la page Stockages de données, puis choisissez Supprimer.

Pour supprimer un magasin HealthLake de données

1. Ouvrez la HealthLake console à la https://console.aws.amazon.com//healthlake/maison.
2. Ouvrez le volet de navigation (≡).
3. Choisissez ensuite Data Stores.
4. Sur la page Stockages de données, choisissez l'option à côté du magasin de données que vous souhaitez supprimer.

5. Ensuite, choisissez Supprimer
6. Dans la boîte de dialogue, tapez **`delete`** pour confirmer que vous souhaitez supprimer le magasin de données sélectionné.
7. Ensuite, choisissez Supprimer. Le statut de votre banque de données passera alors de Actif à Suppression.

Supprimer un magasin de données (AWS SDKset AWS CLI)

Vous pouvez utiliser les exemples de code ci-dessous pour supprimer un magasin de HealthLake données.

AWS CLI

Les exemples suivants illustrent l'utilisation de l'`DeleteFHIRDatastore`opération avec le AWS CLI. Pour exécuter cet exemple, vous devez installer le AWS CLI.

```
aws healthlake delete-fhir-datastore --datastore-id
 'eeb8005725ae22b35b4edbdc68cf2dfd'
```

En cas de succès, vous obtenez la JSON réponse suivante.

```
{
    "DatastoreProperties": {
  "DatastoreId": "eeb8005725ae22b35b4edbdc68cf2dfd",
        "DatastoreArn": "arn:aws:healthlake:us-west-2:728347309221:datastore/fhir/",
        "DatastoreName": "delete-me",
        "DatastoreStatus": "ACTIVE",
        "CreatedAt": "2022-10-03T10:53:45.020000-07:00",
        "DatastoreTypeVersion": "R4",
        "DatastoreEndpoint": "https://healthlake.us-west-2.amazonaws.com/
datastore/5b6e4cd798289a4ab8dad6c1002dd731/r4/",
        "SseConfiguration": {
            "KmsEncryptionConfig": {
                "CmkType": "AWS_OWNED_KMS_KEY"
            }
        },
        "PreloadDataConfig": {
            "PreloadDataType": "SYNTHEA"
        }
    }
```

```
}
```

Python (boto3)

AWS SDKfor Python supporte la `describe_fhir_datastore` méthode qui prend en compte un seul paramètre`DatastoreId`.

```
import boto3

#Create a Healthlake client
healthlake_client = boto3.client('healthlake')

#Call the describe_fhir_datastore method
data_store_details =
 healthlake_client.describe_fhir_datastore(DatastoreId='cdf8f1557e57c543bdc627fb8f12b7fd')

print(data_store_details)
```

En cas de succès, il renvoie un dictionnaire python.

```
{'DatastoreProperties': {'DatastoreId': 'cdf8f1557e57c543bdc627fb8f12b7fd',
 'DatastoreArn': 'arn:aws:healthlake:us-west-2:728347309221:datastore/fhir/
cdf8f1557e57c543bdc627fb8f12b7fd', 'DatastoreName': '08-24-2022-test-data-
store', 'DatastoreStatus': 'ACTIVE', 'CreatedAt': datetime.datetime(2022,
 8, 23, 22, 12, 14, 359000, tzinfo=tzlocal()), 'DatastoreTypeVersion': 'R4',
 'DatastoreEndpoint': 'https://healthlake.us-west-2.amazonaws.com/datastore/
cdf8f1557e57c543bdc627fb8f12b7fd/r4/', 'SseConfiguration': {'KmsEncryptionConfig':
 {'CmkType': 'AWS_OWNED_KMS_KEY'}}, 'PreloadDataConfig': {'PreloadDataType':
 'SYNTHEA'}}, 'ResponseMetadata': {'RequestId': 'aef4b268-ad4b-4b57-
bc97-2da956356835', 'HTTPStatusCode': 200, 'HTTPHeaders': {'date': 'Wed, 05 Oct
 2022 01:21:44 GMT', 'content-type': 'application/x-amz-json-1.0', 'content-
length': '547', 'connection': 'keep-alive', 'x-amzn-requestid': 'aef4b268-ad4b-4b57-
bc97-2da956356835'}, 'RetryAttempts': 0}}
```

Pour renvoyer des informations sur plusieurs banques de données à la fois, utilisez `ListFHIRDatastore`

utilisez la `DeleteFHIRDataStore` commande en utilisant le AWS CLI comme indiqué dans l'exemple suivant. Vous pouvez également supprimer un magasin de données à l'aide de la console delete-fhir-datastore APIou de la console. La suppression d'un magasin de données entraîne la suppression de toutes les versions de FHIR ressources contenues dans le magasin de données et

dans l'infrastructure sous-jacente. Les journaux relatifs à un magasin de données supprimé sont conservés dans le compte de service conformément aux HIPAA directives.

```
aws healthlake delete-fhir-datastore
    --datastore-id (Data Store ID)
```

Comme indiqué dans l'exemple de JSON réponse suivant, le statut passe à `DELETING` « » pour confirmer que le magasin de données et son contenu sont en cours de suppression.

```
{
  "DatastoreEndpoint": "https://healthlake.us-east-1.amazonaws.com/
datastore/eeb8005725ae22b35b4edbdc68cf2dfd/r4/",
    "DatastoreArn": "arn:aws:healthlake:us-east-1:(AWS Account ID):datastore/(Datastore
 ID)",
    "DatastoreStatus": "DELETING",
    "DatastoreId": "(Datastore ID)"
}
```

Utilisation FHIR REST API des interactions avec un magasin HealthLake de données

Dans AWS HealthLake, vous utilisez les REST API interactions Fast Healthcare Interoperability Resources (FHIR) pour gérer et rechercher FHIR des ressources dans votre banque de données. FHIRRESTAPIles interactions sont utilisées pour effectuer des interactions de création, de lecture, de mise à jour et de suppression (CRUD) sur les ressources d'un magasin de données. Vous pouvez également former des chaînes de recherche complexes à l'aide d'une `POST` HTTP requête `GET` ou, car elle HealthLake prend en charge un sous-ensemble d'opérations de recherche FHIR prises en charge.

À des fins de conformité, les types de FHIR ressources sont validés en fonction de la ressource HL7 FHIR R4 StructureDefinition. Pour connaître les FHIR fonctionnalités associées d'un magasin de HealthLake données actif, faites une `GET` demande à l'endroit `metadata` indiqué dans leURL, comme suit.

```
GET https://healthlake.region.amazonaws.com/datastore/datastore-id/r4/metadata
```

En cas de succès, vous recevrez un code de `200` HTTP réponse et la déclaration de capacité pour votre HealthLake banque de données. Pour plus d'informations, consultez `CapabilityStatement`la documentation HL7 FHIR R4.

Le tableau suivant répertorie les FHIR interactions prises en charge par AWS HealthLake.

FHIRinteractions prises en charge par AWS HealthLake

FHIRinteraction	Description
Interactions avec l'ensemble du système	
`capabilities`	Obtenez une déclaration de capacité pour le système
`batch/transaction`	Mettre à jour, créer ou supprimer un ensemble de ressources en une seule interaction
Interactions au niveau du type	
`create`	Création d'une nouvelle ressource avec un ID attribué par le serveur

FHIRinteraction	Description
search	Rechercher un type de ressource en fonction de certains critères de filtre
history	Récupérer l'historique des modifications pour un type de ressource en particulier
Interactions au niveau de l'instance	
read	Lire l'état actuel d'une ressource
history	Lire l'historique des modifications pour une ressource en particulier
vread	Lire l'état d'une version spécifique de la ressource
update	Mettre à jour une ressource par son identifiant (ou la créer s'il s'agit d'une nouvelle ressource)
delete	Suppression d'une ressource

Rubriques

- Types de FHIR ressources pris en charge dans AWS HealthLake
- Exécution des opérations de création, de lecture, de mise à jour et de suppression (CRUD) sur les magasins de HealthLake données
- Recherche dans votre HealthLake banque de données à l'aide des FHIR REST API opérations
- Lire l'historique FHIR des ressources
- Obtenir les données des patients grâce à l'opération Patient $everything FHIR REST API
- Exportation de données depuis votre magasin de HealthLake données à l'aide de $export

Types de FHIR ressources pris en charge dans AWS HealthLake

Le tableau suivant répertorie les types de ressources FHIR R4 pris en charge par AWS HealthLake. Pour plus d'informations, consultez l'index des ressources dans la documentation HL7 FHIR R4.

FHIRtypes de ressources R4 pris en charge par HealthLake

Compte	DetectedIssue	Facture	Praticien

ActivityDefinition	Appareil	d'outils	PractitionerRole
AdverseEvent	DeviceDefinition	Liaison	Procédure
AllergyIntolerance	DeviceMetric	Liste	Provenance
Rendez-vous	DeviceUseStatement	Emplacement	Questionnaire
AppointmentResponse	DeviceRequest	Mesure	QuestionnaireResponse
AuditEvent- Voir note	DiagnosticReport	MeasureReport	RelatedPerson
Binaire	DocumentManifest	Multimédia	RequestGroup
BodyStructure	DocumentReference	Médicaments	ResearchStudy
Bundle - Voir note	EffectEvidenceSynthesis	MedicationAdministration	ResearchSubject
CapabilityStatement	Rencontre	MedicationDispense	RiskAssessment
CarePlan	Point de terminaison	MedicationKnowledge	RiskEvidenceSynthesis
CareTeam	EpisodeOfCare	MedicationRequest	Planificateur
ChargeItem	EnrollmentRequest	MedicationStatement	ServiceRequest
ChargeItemDefinition	EnrollmentResponse	MessageHeader	Slot
Demander	ExplanationOfBenefit	MolecularSequence	Spécimen
ClaimResponse	FamilyMemberHistory	NutritionOrder	StructureDefinition
Communication	Indicateur	Observation	StructureMap
CommunicationRequest	Objectif	OperationOutcome	Substance
Montage	Groupe	Organisation	SupplyDelivery

ConceptMap	GuidanceResponse	OrganizationAffiliation	SupplyRequest
Condition	HealthcareService	Paramètres	Tâche
Consentement	ImagingStudy	Patient	ValueSet
Contrat	Immunisation	PaymentNotice	VisionPrescription
Couverture	ImmunizationEvaluation	PaymentReconciliation	VerificationResult
CoverageEligibilityRequest	ImmunizationRecommendation	Personne	
CoverageEligibilityResponse	InsurancePlan	PlanDefinition	

⚠ FHIRspécifications et HealthLake

- Vous ne pouvez pas effectuer `GET` de `POST` demandes avec les types de ressources suivants : binaire OperationOutcome, bundle et parameters.
- AuditEvent— Une AuditEvent ressource peut être créée ou lue, mais elle ne peut pas être mise à jour ou supprimée.
- Bundle — Il existe plusieurs façons de HealthLake gérer les demandes de bundle. Pour en savoir plus, consultez Gestion de plusieurs FHIR ressources à l'aide de Bundle.
- VerificationResult— Ce type de ressource n'est pris en charge que pour les magasins de données créés après le 9 décembre 2023.

Exécution des opérations de création, de lecture, de mise à jour et de suppression (CRUD) sur les magasins de HealthLake données

Bien que vous utilisiez AWS des actions natives pour gérer des magasins de données, importer des données et exporter des données, vous utilisez quatre FHIR HTTP opérations principales pour créer (`POST`), lire (`GET`), mettre à jour (`PUT`) et supprimer (`DELETE`) FHIR des ressources dans un magasin de HealthLake données. Les rubriques suivantes décrivent comment effectuer les opérations de

création, de lecture, de mise à jour et de suppression (CRUD) sur votre magasin de HealthLake données à l'aide FHIR REST API des services. Vous devez utiliser un processus de signature Signature version 4 pour authentifier les HealthLake API demandes envoyées par le biais d'un HTTP client. Pour en savoir plus, consultez le processus de signature de la version 4 de Signature dans le Références générales AWS.

Rubriques

- Création d'une ressource avec POST
- Lire une ressource avec GET
- Mettre à jour une ressource à l'aide de PUT
- Supprimer une ressource à l'aide de DELETE
- Gestion de plusieurs FHIR ressources à l'aide de Bundle

Création d'une ressource avec **POST**

Vous utilisez une `POST` demande pour créer une nouvelle ressource dans un magasin de HealthLake données. `POST`les demandes ne nécessitent pas que vous fournissiez un `id` élément. Les HealthLake serveurs renvoient un `201` code d'HTTPétat Created lorsqu'une ressource a été créée avec succès.

> **Note**
>
> Lorsque vous faites une `POST` demande sur le type de `DocumentReference` ressource, les extensions existantes ne sont pas modifiées. Ajoutez plutôt AWS HealthLake les nouvelles extensions aux extensions existantes dans votre magasin de données. Pour plus de détails sur la façon dont le traitement du langage naturel (NLP) est HealthLake utilisé sur le type de `DocumentReference` ressource pour extraire des données médicales précieuses, voirUtilisation de la génération automatique de ressources basée sur le traitement du langage naturel (NLP) du type de FHIR DocumentReference ressource dans AWS HealthLake.

Example Création d'une **`Patient`** ressource à l'aide d'une **`POST`** demande.

Pour créer une `POST` demande de banque de HealthLake données, utilisez le point de terminaison de votre banque de données et fournissez un corps de JSON demande. Pour trouver le point de terminaison d'un magasin de données, consultez la HealthLake console sous Data Stores ou utilisez l'escribeFHIRDatastoreopération D dans la AWS HealthLake APIréférence.

POST Request

```
POST https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/
Patient
```

JSON Request Body

```
{
  "resourceType": "Patient",
  "identifier": [ { "system": "urn:oid:1.2.36.146.595.217.0.1", "value":
 "12345" } ],
  "name": [ {
      "family": "Silva",
      "given": ["Ana", "Carolina"]
  } ],
  "gender": "female",
  "birthDate": "1992-02-10"
}
```

Réponse d'JSON

Pour confirmer la création de la ressource destinée aux patients, vous recevrez un code de HTTP statut 201 Créé et la JSON réponse suivante.

```
{
  "resourceType": "Patient",
  "identifier": [
    {
      "system": "urn:oid:1.2.36.146.595.217.0.1",
      "value": "12345"
    }
  ],
  "name": [
    {
      "family": "Silva",
      "given": [
        "Ana",
        "Carolina"
      ]
    }
  ],
```

```
  "gender": "female",
  "birthDate": "1992-02-10",
  "id": "274b408a-1201-4e9f-a621-1df937f1a26d",
  "meta": {
    "lastUpdated": "2022-06-13T23:31:24.427Z"
  }
}
```

Lire une ressource avec **GET**

Cet exemple montre comment lire une FHIR ressource destinée aux patients à l'aide d'une GET demande.

Example Lire une **Patient** ressource spécifique à l'aide d'une **GET** requête.

Pour créer une GET demande de banque de HealthLake données, utilisez le point de terminaison de votre banque de données. Pour trouver le point de terminaison d'un magasin de données, consultez la HealthLake console sous Data Stores ou utilisez l'escribeFHIRDatastoreopération D dans la AWS HealthLake APIréférence.

Vous devez également inclure le type de ressource **Patient** et un identifiant valide**2de04858-ba65-44c1-8af1-f2fe69a977d9**.

```
GET https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/
Patient/2de04858-ba65-44c1-8af1-f2fe69a977d9
```

Réponse d'JSON

En cas de succès, vous recevrez un code de 200 HTTP statut et la JSON réponse suivante.

```
{
    "resourceType": "Patient",
    "active": true,
    "name": [
        {
            "use": "official",
            "family": "Doe",
            "given": [
                "Jane"
            ]
        },
```

```
        {
            "use": "usual",
            "given": [
                "Jane"
            ]
        }
    ],
    "gender": "female",
    "birthDate": "1966-09-01",
    "meta": {
        "lastUpdated": "2020-11-23T06:24:13.202Z"
    },
    "id": "2de04858-ba65-44c1-8af1-f2fe69a977d9"
}
```

Mettre à jour une ressource à l'aide de **PUT**

L'exemple suivant montre comment mettre PUT à jour les informations relatives à un patient dans le type de FHIR ressource patient. De plus, lorsque vous faites une PUT demande sur une ressource qui n'a pas encore été créée, une version initiale est créée.

Votre demande renverra soit un code d'`200`HTTPétat si la ressource a été mise à jour, soit un code d'`201`HTTPétat si une nouvelle ressource a été créée.

> **Note**
>
> Lorsque vous faites une PUT demande sur le type de `DocumentReference` ressource, les extensions existantes ne sont pas modifiées. Ajoutez plutôt AWS HealthLake les nouvelles extensions aux extensions existantes dans votre magasin de données. Pour plus de détails sur la façon dont le traitement du langage naturel (NLP) est HealthLake utilisé sur le type de `DocumentReference` ressource pour extraire des données médicales précieuses, voirUtilisation de la génération automatique de ressources basée sur le traitement du langage naturel (NLP) du type de FHIR DocumentReference ressource dans AWS HealthLake.

Example Mise à jour d'un type de **`Patient`** ressource à l'aide d'une **`PUT`** demande

Lorsque vous faites une PUT demande, vous avez besoin du point de terminaison du magasin de données, du nom du type de ressource que vous souhaitez mettre à jour, d'un identifiant et d'un corps de JSON demande.

Si vous créez une nouvelle ressource, elle utilise l'identifiant fourni pour créer la nouvelle ressource.
PUT

PUT Request

Exemple de structure d'une PUT demande valide :

```
PUT https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/
Patient/2de04858-ba65-44c1-8af1-f2fe69a977d9
```

JSON Request Body

Exemple de JSON corps utilisé pour mettre à jour la ressource patient spécifiée.

```
{
  "id": "2de04858-ba65-44c1-8af1-f2fe69a977d9",
  "resourceType": "Patient",
  "active": true,
  "name": [
    {
      "use": "official",
      "family": "Doe",
      "given": [
        "Jane"
      ]
    },
    {
      "use": "usual",
      "given": [
        "Jane"
      ]
    }
  ],
  "gender": "female",
  "birthDate": "1985-12-31"
}
```

Réponse de JSON

Vous recevrez ce qui suit JSON en réponse pour confirmer le changement :

```
{
```

```
 "id": "2de04858-ba65-44c1-8af1-f2fe69a977d9",
 "resourceType": "Patient",
 "active": true,
 "name": [{
   "use": "official",
   "family": "Doe",
   "given": [
    "Jane"
   ]
  },
  {
   "use": "usual",
   "given": [
    "Jane"
   ]
  }
 ],
 "gender": "female",
 "birthDate": "1985-12-31",
 "meta": {
  "lastUpdated": "2020-11-23T06:43:45.133Z"
 }
}
```

Mise à jour conditionnelle

La mise à jour conditionnelle permet de mettre à jour une ressource existante en fonction de certains critères de recherche d'identification, plutôt que d'un identifiant logique. Lorsque le serveur traite cette mise à jour, il effectue une recherche à l'aide de ses fonctionnalités de recherche standard pour le type de ressource, dans le but de résoudre un identifiant logique unique pour cette demande.

L'action entreprise dépend du nombre de correspondances trouvées :

- Aucune correspondance, aucun identifiant fourni dans le corps de la requête : le serveur crée la ressource.
- Aucune correspondance, identifiant fourni et la ressource n'existe pas déjà avec l'identifiant : le serveur traite l'interaction comme une interaction Update as Create.
- Aucune correspondance, identifiant fourni et existant déjà : le serveur rejette la mise à jour avec une `409 Conflict` erreur.
- Une correspondance, aucun identifiant de ressource fourni OU (identifiant de ressource fourni et correspondant à la ressource trouvée) : le serveur effectue la mise à jour par rapport à la ressource

correspondante comme ci-dessus où, si la ressource a été mise à jour, le serveur SHALL renvoie un `200 OK` ;

- Une correspondance, l'identifiant de ressource fourni mais ne correspond pas à la ressource trouvée : le serveur renvoie une `409 Conflict` erreur indiquant que la spécification de l'identifiant du client posait un problème, de préférence avec un OperationOutcome
- Correspondances multiples : le serveur renvoie une `412 Precondition Failed` erreur indiquant que les critères du client n'étaient pas suffisamment sélectifs, de préférence avec un OperationOutcome

Example — Mettez à jour une ressource destinée aux patients dont le nom est Peter, la date de naissance est le 1er janvier 2000 et le numéro de téléphone 1234567890 :

```
PUT https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/
Patient?name=peter&birthdate=2000-01-01&phone=1234567890
```

Supprimer une ressource à l'aide de **DELETE**

Pour supprimer une ressource de votre banque de HealthLake données, vous devez en faire la `DELETE` HTTP demande.

Example Suppression d'un type de **`Patient`** ressource spécifique à l'aide d'une **`DELETE`** demande.

Pour créer une `DELETE` demande, utilisez le point de terminaison du magasin de données. Pour trouver le point de terminaison d'un magasin de données, consultez la HealthLake console sous Data Stores ou en utilisant l'escribeFHIRDatastoreopération D trouvée dans la AWS HealthLake APIréférence.

Vous devez également inclure le type de ressource et un identifiant valide.

```
DELETE https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/
Patient/2de04858-ba65-44c1-8af1-f2fe69a977d9
```

Réponse de HTTP

En cas de succès, vous recevrez un code d'`204`HTTPétat confirmant que la ressource ne se trouve plus dans le magasin de données. Lorsqu'une demande de suppression échoue, vous recevrez un code d'HTTPétat de la série 400 indiquant pourquoi la `DELETE` demande a échoué.

Gestion de plusieurs FHIR ressources à l'aide de Bundle

Dans la spécification HL7 FHIR R4, les bundles sont simplement un ensemble de ressources. HealthLake prend en charge la création d'un type de ressource Bundle dans une FHIR REST API demande et l'utilisation d'une transaction groupée pour effectuer plusieurs CRUD opérations dans une seule FHIR REST API demande. Dans une transaction groupée, vous devez spécifier le type de bundle comme `batch` dans la FHIR REST API demande.

Toutes les demandes de bundle sont enregistrées par AWS CloudTrail. Pour en savoir plus sur l'utilisation CloudTrail avec HealthLake, voirJournalisation des appels AWS HealthLake API avec AWS CloudTrail.

HL7FHIRRessources R4 (externes)

- Pour lire la spécification complète, voir Type de ressource : Bundle dans l'index de la FHIR documentation.
- Pour en savoir plus sur les interactions par lots utilisant le FHIR RESTAPI, consultez la section Interactions par lots utilisant le FHIR REST API dans l'index de FHIR documentation.

Les sections ci-dessous décrivent comment structurer une FHIR REST API demande afin de créer une nouvelle ressource Bundle ou de traiter les ressources individuellement à l'aide de transactions groupées.

> ⚠ Différences entre la HealthLake console, le AWS CLI, et le AWS SDKs
>
> La HealthLake console prend uniquement en charge les opérations de type Bundle lorsque le type de ressource Bundle est spécifié dans la FHIR REST API demandeURL.

Exécution de plusieurs CRUD opérations à l'aide de FHIR bundles

Lorsqu'aucun type de ressource n'est spécifié dans votre demandeURL, celle-ci est analysée en tant que transactions de banque de données individuelles. FHIR REST API Chaque CRUD opération fournie dans le JSON corps est évaluée et un code d'HTTPétat spécifique est renvoyé. HealthLake prend en charge le type de bundle`batch`.

Pour effectuer plusieurs CRUD opérations dans une seule FHIR REST API demande, procédez comme suit :

La liste suivante montre les parties tronquées du corps d'une demande utilisée dans une FHIR REST API demande groupée. Pour un corps de demande complet, voir Création d'une demande groupée impliquant plusieurs CRUD opérations.

1. Ne spécifiez aucun type de ressource dans votre POST demande :

```
POST https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/
```

2. Dans le corps de la demande, spécifiez le type de bundle comme `"type": "batch"`
3. Dans le corps de la demande, spécifiez les données spécifiques à la ressource pour chaque CRUD interaction en commençant par la `resource` clé.
4. Chaque CRUD opération est spécifiée comme suit `request` dans le corps de la demande :

```
{ ...
  "request" : {
    "method" : "HTTP-VERB",
    "url" : "FHIR-RESOURCE-TYPE-URL"
  }
  ...
}
```

Dans la JSON réponse, vous obtenez un code d'HTTPétat pour chaque CRUD opération spécifiée dans la demande.

HealthLake limite les transactions groupées

- Pour en savoir plus sur les limites imposées HealthLake aux offres groupées, consultezAWS HealthLake points de terminaison et quotas.

Voici un exemple d'opération Bundle contenant plusieurs CRUD opérations.

Example — Création d'une demande de bundle impliquant plusieurs CRUD opérations.

Pour effectuer une FHIR REST API demande qui effectue plusieurs CRUD opérations, vous devez effectuer une POST demande en utilisant le point de terminaison de votre banque de données et fournir un corps de JSON demande.

Vous pouvez trouver le point de terminaison de votre magasin de données dans la HealthLake console sous Data Stores ou en utilisant l'escribeFHIRDatastoreopération D dans la AWS HealthLake APIréférence.

POST Request

Faites une POST demande en utilisant le point de terminaison de votre banque de données. Utilisez l'onglet suivant, Corps de la JSON demande, pour voir les éléments requis du corps de la demande.

```
POST https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/
```

JSON Request Body

Dans le corps de la demande, vous devez fournir les paires clé:valeur suivantes ainsi que toute autre donnée FHIR spécifique à la ressource concernant les demandes individuelles. CRUD Le premier exemple montre un corps de JSON demande tronqué mettant en évidence les éléments requis. Le deuxième exemple montre le corps complet de JSON la demande.

```
{
  "resourceType": "Bundle",
  "id": "bundle-batch-operation",
  "meta": {
    "lastUpdated": "2014-08-18T01:43:30Z"
  },
  "type": "batch", ## Required
  "entry": [
    {
      ## CRUD Transaction - 1
      "resource": {
        "resourceType": "Patient",
        ...
      },
      "request": { ## Required
        "method": "POST",
        "url": "Patient"
      }
    },
    {
      ## CRUD Transaction - 2
      "resource": {
        "resourceType": "Medication",
```

```
        ...
      },
      "request": { ## Required
        "method": "POST",
        "url": "Medication"
      }
    }
  ]
}
```

Voici un exemple complet illustrant la création d'un nouveau type `Patient` de `Medication` ressource.

```
{
  "resourceType": "Bundle",
  "id": "bundle-transaction",
  "meta": {
    "lastUpdated": "2014-08-18T01:43:30Z"
  },
  "type": "batch",
  "entry": [
    {
      "resource": {
        "resourceType": "Patient",
        "meta": {
          "lastUpdated": "2022-06-03T17:53:36.724Z"
        },
        "text": {
          "status": "generated",
          "div": "Some narrative"
        },
        "active": true,
        "name": [
          {
            "use": "official",
            "family": "Jackson",
            "given": [
              "Mateo",
              "James"
            ]
          }
        ],
        "gender": "male",
```

```
        "birthDate": "1974-12-25"
      },
      "request": {
        "method": "POST",
        "url": "Patient"
      }
    },
    {
      "resource": {
        "resourceType": "Medication",
        "id": "med0310",
        "contained": [
          {
            "resourceType": "Substance",
            "id": "sub03",
            "code": {
              "coding": [
                {
                  "system": "http://snomed.info/sct",
                  "code": "55452001",
                  "display": "Oxycodone (substance)"
                }
              ]
            }
          }
        ],
        "code": {
          "coding": [
            {
              "system": "http://snomed.info/sct",
              "code": "430127000",
              "display": "Oral Form Oxycodone (product)"
            }
          ]
        },
        "form": {
          "coding": [
            {
              "system": "http://snomed.info/sct",
              "code": "385055001",
              "display": "Tablet dose form (qualifier value)"
            }
          ]
        },
```

```
        "ingredient": [
          {
            "itemReference": {
              "reference": "#sub03"
            },
            "strength": {
              "numerator": {
                "value": 5,
                "system": "http://unitsofmeasure.org",
                "code": "mg"
              },
              "denominator": {
                "value": 1,
                "system": "http://terminology.hl7.org/CodeSystem/v3-
orderableDrugForm",
                "code": "TAB"
              }
            }
          }
        ]
      },
      "request": {
        "method": "POST",
        "url": "Medication"
      }
    }
  ]
}
```

Réponse d'JSON

Pour confirmer la création des ressources spécifiées dans l'exemple de transaction groupée, vous obtenez 201 le code d'HTTPétat Created pour chaque CRUD opération incluse. Lorsqu'une CRUD opération échoue, vous obtenez le HTTP statut de la série 400 indiquant pourquoi la demande individuelle a échoué.

```
{
  "resourceType": "Bundle",
  "type": "batch-response",
  "timestamp": "2022-06-15T01:31:34.300+00:00",
  "entry": [
    {
```

```
      "response": {
        "status": "201",
        "location": "Patient/fd68ce38-ba30-4459-9eeb-476ad9f4f4ca",
        "lastModified": "2022-06-15T01:31:34.180+00:00"
      }
    },
    {
      "response": {
        "status": "201",
        "location": "Medication/5bf3b8cc-4076-4219-aba1-e2c53d7916f4",
        "lastModified": "2022-06-15T01:31:34.180+00:00"
      }
    }
  ]
}
```


Regroupement des ressources en tant que type de ressource Bundle

Pour créer un nouveau type de ressource Bundle, vous devez le spécifier `Bundle` dans la FHIR REST API demande et fournir un JSON corps valide contenant les ressources que vous souhaitez regrouper.

Lorsque Bundle est spécifié dans la demandeURL, le contenu du corps de la JSON demande est enregistré tel quel dans votre magasin de HealthLake données. Par conséquent, aucune CRUD opération ne peut être effectuée sur les différents types de ressources. Les ensembles de ce type se voient attribuer un nouvel identifiant de ressource unique. Les ressources étant enregistrées telles quelles, vous ne pouvez pas créer `GET` ou `POST` demander des ressources individuelles enregistrées dans le type de ressource Bundle.

> **Note**
>
> La spécification HL7 FHIR R4 prend également en charge le regroupement de ressources à l'aide de groupes, de compositions et de listes. Lorsque vous créez ces types de ressources, les ressources individuelles ne sont pas contenues directement. Ils utilisent plutôt l'`Reference`élément pour pointer vers les ressources individuelles. L'utilisation de ces types de ressources vous permet donc de modifier les ressources individuelles qu'ils contiennent.

Pour créer un type de `Bundle` ressource, vous devez le spécifier dans votre `POST` demande et fournir une JSON énumération des ressources que vous souhaitez inclure.

Example — Création d'une ressource Bundle à l'aide d'une **POST** requête

Pour créer une `bundle` ressource, procédez comme suit

1. Formatez une FHIR REST API demande comme suit :

   `POST https://healthlake.`**`your-region`**`.amazonaws.com/datastore/`**`your-datastore-id`**`/r4/`**`Bundle`**

2. Indiquez JSON le corps qui spécifie les ressources que vous souhaitez regrouper. Cet exemple regroupe deux ressources destinées aux patients.

```
{
  "resourceType": "Bundle",
  "id": "bundle-transaction",
  "meta": {
    "lastUpdated": "2018-03-11T11:22:16Z"
  },
  "type": "document",
  "entry": [
    {
      "resource": {
        "resourceType": "Patient",
        "name": [
          {
            "family": "Smith",
            "given": [
              "Jane"
            ]
          }
        ],
        "gender": "female",
        "address": [
          {
            "line": [
              "123 Main St."
            ],
            "city": "Anycity",
            "state": "Any State",
            "postalCode": "12345"
          }
        ]
      }
    },
```

```
    {
      "resource": {
        "resourceType": "Patient",
        "name": [
          {
            "family": "Jackson",
            "given": [
              "Mateo"
            ]
          }
        ],
        "gender": "male",
        "address": [
          {
            "line": [
              "1234 Main St."
            ],
            "city": "Anycity",
            "state": "Any State",
            "postalCode": "12345"
          }
        ]
      }
    }
  ]
}
```

Recherche dans votre HealthLake banque de données à l'aide des FHIR REST API opérations

HealthLake prend en charge la recherche dans votre banque de données à l'aide REST API des opérations fournies dans le cadre de la FHIR norme. Dans cette section, vous trouverez des exemples illustrant comment effectuer `GET` des `POST` demandes sur plusieurs types de ressources différents.

> **Note**
>
> Pour les requêtes impliquant des informations personnellement identifiables (PII) ou des informations de santé protégées (PHI), il est recommandé d'utiliser les `POST` demandes.

Dans une `POST` demande, PII ou PHI est ajouté dans le corps de la demande et est crypté pendant le transit.

La FHIR spécification prend en charge plusieurs types de paramètres de recherche, mais ne HealthLake prend en charge qu'un sous-ensemble. Pour plus d'informations, consultez Types de paramètres de recherche pris en charge et Paramètres de recherche avancés pris en charge par HealthLake.

Recherche dans votre magasin de données à l'aide FHIR REST API des opérations.

- Types de paramètres de recherche pris en charge
- Paramètres de recherche avancés pris en charge par HealthLake
  - _include
  - _revinclude
  - _summary
  - _elements
  - _total
  - _sort
  - _count
  - Chaining and Reverse Chaining(_has)
- Modificateurs de recherche pris en charge
- Comparateurs de recherche pris en charge
- Paramètres de recherche non pris en charge par HealthLake
- Recherche à l'aide POST d'exemples
- Recherche à l'aide GET d'exemples

Types de paramètres de recherche pris en charge

Le tableau suivant indique les types de paramètres de recherche pris en charge dans HealthLake.

Types de paramètres de recherche pris en charge

Paramètre de recherche	Description
_identifiant	ID de ressource (pas completURL)
_lastUpdated	Date de dernière mise à jour. Le serveur peut décider de la précision des limites.
_tag	Effectuez une recherche à l'aide d'une balise de ressource.
_profil	Recherchez toutes les ressources associées à un profil.
_sécurité	Effectuez une recherche sur les étiquettes de sécurité appliquées à cette ressource.
_source	Recherchez d'où vient la ressource.
_texte	Effectuez une recherche sur le récit de la ressource.
createdAt	Recherche sur une extension personnalisée - createdAt.

> **Note**
>
> Les paramètres de recherche suivants ne sont pris en charge que pour les banques de données créées après le 9 décembre 2023 : _security, _source, _text,. createdAt

Le tableau suivant présente des exemples de modification des chaînes de requête en fonction des types de données spécifiés pour un type de ressource donné. Pour des raisons de clarté, les caractères spéciaux de la colonne des exemples n'ont pas été codés. Pour réussir une requête, assurez-vous que la chaîne de requête a été correctement encodée.

Types de paramètres de recherche	Détails	Exemples

Nombre	Recherche une valeur numérique dans une ressource spécifiée. Des chiffres significatifs sont observés.
	
	Le nombre de chiffres significatifs est défini par la valeur du paramètre de recherche, à l'exception des zéros en tête.
	
	Les préfixes de comparaison sont autorisés.
		`[parameter]=100`
		
		`[parameter]=1e2`
		
		`[parameter]=lt100`

Date/ DateTime	Recherche une date ou une heure spécifique. Le format attendu est `yyyy-mm-ddThh:mm:ss[Z|(+|-)hh:mm]` mais peut varier.
	
	Accepte les types de données suivants : `datedateTime`,`instant`,`PeriodTiming`. Pour plus de détails sur l'utilisation de ces types de données dans les recherches, voir la date dans l'index de la FHIR documentation.
	
	Les préfixes de comparaison sont autorisés.
		`[parameter]=eq2013-01-14`
		
		`[parameter]=gt2013-01-14T10:00`
		
		`[parameter]=ne2013-01-14`

Types de paramètres de recherche	Détails	Exemples

Chaîne	Recherche une séquence de caractères en distinguant majuscules et minuscules.
	
	Supporte `HumanName` les deux `Address` types. Pour plus de détails, consultez l'entrée de type de HumanName données et les entrées de type de `Address` données dans l'index de FHIR documentation.
	
	La recherche avancée est prise en charge à l'aide de `:text` modificateurs.
		`[base]/Patient?given=eve`
		
		`[base]/Patient?given:contains=eve`

Jeton	Recherche une close-to-exact correspondance par rapport à une chaîne de caractères, souvent comparée à une paire de valeurs de code médical. La distinction majuscule s/minuscules est liée au système de code utilisé lors de la création d'une requête. Les requêtes basées sur les subsumptions peuvent aider à réduire les problèmes liés à la distinction majuscules/minuscules. Pour plus de clarté, `|` il n'a pas été encodé.	`[parameter]=[system]|[code]` : Ici, il `[system]` fait référence à un système de codage et `[code]` à une valeur de code trouvée dans ce système spécifique. `[parameter]=[code]` : Ici, votre saisie correspondra à un code ou à un système. `[parameter]=|[code]` : Ici, votre entrée correspondra à un code, et la propriété du système n'a aucun identifiant.

Types de paramètres de recherche	Détails	Exemples

Composite	Recherche plusieurs paramètres au sein d'un même type de ressource à l'aide des modificateurs $ et de l'`,` opération.
	
	Les préfixes de comparaison sont autorisés.
		`/Patient?language=FR,NL&language=EN`
		
		`Observation?component-code-value-quantity=http://loinc.org|8480-6$lt60`
		
		`[base]/Group?characteristic-value=gender$mixed`

Quantité	Recherche un nombre, un système et un code sous forme de valeurs. Un numéro est requis, mais le système et le code sont facultatifs. Basé sur le type de données de quantité. Pour plus de détails, consultez la section Quantité dans l'index de la FHIR documentation.
	
	Utilise la syntaxe supposée suivante `[parameter]=[prefix][number]|[system]|[code]`
		`[base]/Observation?value-quantity=5.4|http://unitsofmeasure.org|mg`
		
		`[base]/Observation?value-quantity=5.4|http://unitsofmeasure.org|mg`
		
		`[base]/Observation?value-quantity=5.4|http://unitsofmeasure.org|mg`
		
		`[base]/Observation?value-quantity=le5.4|http://unitsofmeasure.org|mg`

Types de paramètres de recherche	Détails	Exemples

Référence	Recherche des références à d'autres ressources.	`[base]/Observation?subject=Patient/23`
		
		`test`

URI	Recherche une chaîne de caractères identifiant sans ambiguïté une ressource particulière.	`[base]/ValueSet?url=http://acme.org/fhir/ValueSet/123`
Spécial	Recherches basées sur des NLP extensions médicales intégrées.	

Paramètres de recherche avancés pris en charge par HealthLake

HealthLake prend en charge les paramètres de recherche avancée suivants.

Name (Nom)	Description	Exemple	Capacité
`_includ`	Utilisé pour demander que des ressources supplémentaires soient renvoyées dans une demande de recherche . Elle renvoie les ressources référencées par l'instance de ressource cible.	`Encounter?_include=Encounter:subject`	
`_revinc` de	Utilisé pour demander que des ressources supplémentaires soient renvoyées dans une demande de recherche. Elle	`Patient?_id= **patient-identifier** &_revincl`	

Name (Nom)	Description	Exemple	Capacité
	renvoie des ressources qui font référence à l'instance de ressource principale.	`ude=Encou nter:patient`	
`_summary`	Le résumé peut être utilisé pour demander un sous-ensemble de la ressource.	`Patient?_ summary=text`	Les paramètres récapitulatifs suivants sont pris en charge : `_summary= true _summary= false ,,_summary= text ,_summary=data .`
`_elemen`	Demandez qu'un ensemble spécifique d'éléments soit renvoyé dans le cadre d'une ressource dans les résultats de recherche.	`Patient?_ elements= identifie r,active, link`	
`_total`	Renvoie le nombre de ressources correspondant aux paramètres de recherche.	`Patient?_ total=acc urate`	Support`_total=ac curate ,_total=none .`
`_sort`	Indiquez l'ordre de tri des résultats de recherche renvoyés à l'aide d'une liste séparée par des virgules. Le `-` préfixe peut être utilisé pour n'importe quelle règle de tri de la liste séparée par des virgules afin d'indiquer l'ordre décroissant.	`Observati on?_sort= status,-date`	Support : tri par champs avec types`Number, String, Quantity, Token, URI, Reference` . Le tri par n'`Date`est pris en charge que pour les banques de données créées après le 9 décembre 2023. Support jusqu'à 5 règles de tri.
`_count`	Contrôlez le nombre de ressources renvoyées par page du bundle de recherche.	`Patient?_ count=100`	La taille de page maximale est de 100.

Name (Nom)	Description	Exemple	Capacité
`chainin`	Éléments de recherche des ressources référencées. `.`Dirige la recherche en chaîne vers l'élément de la ressource référencée.	`DiagnosticReport?subject:Patient.name=peter`	
`reverse chainin (_has)`	Recherchez une ressource en fonction des éléments des ressources qui y font référence.	`Patient?_has:Observation:patient:code=1234-5`	

`_include`

L'utilisation `_include` dans une requête de recherche permet de renvoyer également FHIR des ressources spécifiées supplémentaires. `_include`À utiliser pour inclure des ressources liées vers le bas.

Example — À utiliser **`_include`** pour retrouver les patients ou le groupe de patients chez lesquels une toux a été diagnostiquée

Vous devez effectuer une recherche sur le type de `Condition` ressource en spécifiant le code de diagnostic de la toux, puis en utilisant « `_include` Spécifier que vous souhaitez que le `subject` diagnostic soit également renvoyé ». Dans le type de `Condition` ressource, on `subject` entend soit le type de ressource du patient, soit le type de ressource du groupe.

Pour des raisons de clarté, les caractères spéciaux de l'exemple n'ont pas été codés. Pour réussir une requête, assurez-vous que la chaîne de requête a été correctement encodée.

```
GET https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/
Condition?code=49727002&_include=Condition:subject
```

_revinclude

L'utilisation _revinclude dans une requête de recherche permet de renvoyer également FHIR des ressources spécifiées supplémentaires. _revincludeÀ utiliser pour inclure des ressources liées à l'envers.

Example — À utiliser **_revinclude** pour inclure des types de ressources de rencontre et d'observation connexes liés à un patient spécifique

Pour effectuer cette recherche, vous devez d'abord définir l'individu Patient en spécifiant son identifiant dans le paramètre _id de recherche. Vous devez ensuite spécifier des FHIR ressources supplémentaires à l'aide de la structure Encounter:patient et0bservation:patient.

Pour des raisons de clarté, les caractères spéciaux de l'exemple n'ont pas été codés. Pour réussir une requête, assurez-vous que la chaîne de requête a été correctement encodée.

```
GET https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/
Patient?_id=patient-
identifier&_revinclude=Encounter:patient&_revinclude=0bservation:patient
```

_summary

L'utilisation _summary dans une requête de recherche permet à l'utilisateur de demander un sous-ensemble de la FHIR ressource. Il peut contenir l'une des valeurs suivantes :true, text, data, false. Toutes les autres valeurs seront considérées comme non valides. Les ressources renvoyées seront marquées 'SUBSETTED' dans le méta.tag, pour indiquer que les ressources sont incomplètes.

- true: renvoie tous les éléments pris en charge marqués comme « résumé » dans la définition de base de la ou des ressources.
- text: Renvoie uniquement les éléments « text », « id », « méta » et uniquement les éléments obligatoires de haut niveau.
- data: Renvoie toutes les parties sauf l'élément « texte ».
- false: renvoie toutes les parties de la ou des ressources

Dans une seule demande de recherche, il _summary=text ne peut pas être combiné avec _include des paramètres _revinclude de recherche.

Example — Récupère l'élément « texte » des ressources destinées aux patients dans une banque de données.

```
GET https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/
Patient?_summary=text
```

_elements

L'utilisation `_elements` dans une requête de recherche permet de demander FHIR des éléments de ressources spécifiques. Les ressources renvoyées seront marquées `'SUBSETTED'` dans le méta.tag, pour indiquer que les ressources sont incomplètes.

Le `_elements` paramètre consiste en une liste de noms d'éléments de base séparés par des virgules, tels que les éléments définis au niveau racine de la ressource. Seuls les éléments listés doivent être renvoyés. Si les valeurs des `_elements` paramètres contiennent des éléments non valides, le serveur les ignorera et renverra des éléments obligatoires et des éléments valides.

`_elements`ne sera pas applicable aux ressources incluses (ressources renvoyées dont le mode de recherche est`include`).

Dans une seule demande de recherche, il `_elements` ne peut pas être combiné avec les paramètres `_summary` de recherche.

Example — Obtenez des éléments « identifiant », « actif », « lien » des ressources destinées aux patients dans votre HealthLake banque de données.

```
GET https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/
Patient?_elements=identifier,active,link
```

_total

L'utilisation `_total` dans une requête de recherche renverra le nombre de ressources correspondant aux paramètres de recherche demandés. HealthLake renverra le nombre total de ressources correspondantes (ressources renvoyées dont le mode de recherche est`match`) dans la réponse `Bundle.total` de recherche.

`_total`prend en charge `accurate` les valeurs des `none` paramètres. `_total=estimate`n'est pas pris en charge. Toutes les autres valeurs seront considérées comme non valides. `_total`n'est pas applicable aux ressources incluses (ressources renvoyées dont le mode de recherche est`include`).

Example — Obtenez le nombre total de ressources pour les patients dans une banque de données :

```
GET https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/
Patient?_total=accurate
```

_sort

L'utilisation `_sort` dans la requête de recherche organise les résultats dans un ordre spécifique. Les résultats sont classés par ordre de priorité en fonction de la liste des règles de tri séparées par des virgules. Les règles de tri doivent être des paramètres de recherche valides. Toutes les autres valeurs seront considérées comme non valides.

Dans une seule demande de recherche, vous pouvez utiliser jusqu'à 5 paramètres de recherche de tri. Vous pouvez éventuellement utiliser un `-` préfixe pour indiquer l'ordre décroissant. Le serveur triera par ordre croissant par défaut.

Les types de paramètres de recherche de tri pris en charge sont :`Number, String, Date, Quantity, Token, URI, Reference`. Si un paramètre de recherche fait référence à un élément imbriqué, il n'est pas pris en charge pour le tri. Par exemple, une recherche sur le « nom » du type de ressource Patient fait référence à l'élément Patient.Name dont le type de HumanName données est considéré comme imbriqué. Par conséquent, le tri des ressources pour les patients par « nom » n'est pas pris en charge.

Example — Accédez aux ressources des patients dans une banque de données et triez-les par date de naissance dans l'ordre croissant :

```
GET https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/
Patient?_sort=birthdate
```

_count

Le paramètre `_count` est défini comme une instruction adressée au serveur concernant le nombre de ressources à renvoyer sur une seule page.

La taille de page maximale est de 100. Toute valeur supérieure à 100 n'est pas valide. `_count=0`n'est pas pris en charge.

Example — Recherchez la ressource Patient et définissez la taille de la page de recherche sur 25 :

```
GET https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/
Patient?_count=25
```

Chaining and Reverse Chaining(_has)

Le chaînage et le chaînage inversé FHIR constituent un moyen plus efficace et plus compact d'obtenir des données interconnectées, réduisant ainsi le besoin de plusieurs requêtes distinctes et rendant la récupération de données plus pratique pour les développeurs et les utilisateurs.

Si un niveau de récursivité renvoie plus de 100 résultats, il renverra 4xx pour éviter que la banque de données HealthLake ne soit surchargée et ne provoque pas de paginations multiples.

Example — Chainage - Obtient tout DiagnosticReport ce qui fait référence à un patient dont le nom est Peter.

```
GET https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/
DiagnosticReport?subject:Patient.name=peter
```

Example — Chainage inversé - Accédez aux ressources du patient, où la ressource du patient est référencée par au moins une observation dont le code de l'observation est 1234, et où l'observation fait référence à la ressource du patient dans le paramètre de recherche du patient.

```
GET https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/
Patient?_has:Observation:patient:code=1234
```

Modificateurs de recherche pris en charge

Les modificateurs de recherche sont utilisés avec les champs basés sur des chaînes. Tous les modificateurs de recherche HealthLake utilisés utilisent une logique booléenne. Par exemple, vous pouvez `:contains` spécifier qu'un champ de chaîne plus grand doit inclure une petite chaîne pour qu'il soit inclus dans les résultats de recherche.

Modificateurs de recherche pris en charge

Modificateur de recherche	Type
:manquant	Tous les paramètres sauf `Composite`

Modificateur de recherche	Type
:exact	Chaîne
:contient	Chaîne
:non	Jeton
:texte	Jeton
:identifiant	Référence

Comparateurs de recherche pris en charge

Vous pouvez utiliser des comparateurs de recherche pour contrôler la nature de la correspondance lors d'une recherche. Vous pouvez utiliser des comparateurs lorsque vous effectuez une recherche dans les champs de numéro, de date et de quantité. Le tableau suivant répertorie les comparateurs de recherche et leurs définitions pris en charge par HealthLake.

Comparateurs de recherche pris en charge

Comparateur de recherche	Description
eq	La valeur du paramètre dans la ressource est égale à la valeur fournie.
ne	La valeur du paramètre dans la ressource n'est pas égale à la valeur fournie.
gt	La valeur du paramètre dans la ressource est supérieure à la valeur fournie.
lt	La valeur du paramètre dans la ressource est inférieure à la valeur fournie.
gm	La valeur du paramètre dans la ressource est supérieure ou égale à la valeur fournie.

Comparateur de recherche	Description
le	La valeur du paramètre dans la ressource est inférieure ou égale à la valeur fournie.
sa	La valeur du paramètre dans la ressource commence après la valeur fournie.
eb	La valeur du paramètre dans la ressource se termine avant la valeur fournie.

Paramètres de recherche non pris en charge par HealthLake

Pour obtenir la liste complète des paramètres de recherche pris en charge, consultez le registre FHIR des paramètres de recherche. HealthLake prend en charge tous les paramètres de recherche à l'exception de ceux répertoriés dans le tableau.

Paramètres de recherche non pris en charge

Composition du bundle	Lieu : à proximité
Identifiant du bundle	C onsent-source-reference
Message groupé	Patient sous contrat
Type d'offre groupée	Contenu des ressources
Horodatage du bundle	Requête de ressources

Recherche à l'aide POST d'exemples

Vous pouvez effectuer une recherche dans un magasin de HealthLake données en effectuant des POST demandes. Vous pouvez fournir des paramètres de requête dans le corps de la demande URI ou dans le corps d'une demande, mais vous ne pouvez pas utiliser les deux dans une seule demande.

Les exemples présentés dans cette rubrique suivent cette bonne pratique.

> **Note**
>
> Pour les requêtes impliquant des informations personnellement identifiables (PII) ou des informations de santé protégées (PHI), il est recommandé d'utiliser les `POST` demandes. Dans une `POST` demande, PII ou PHI est ajouté dans le corps de la demande et est crypté pendant le transit.

Lorsque vous faites une POST demande avec un paramètre dans le corps de la demande, utilisez-le dans le `Content-Type: application/x-www-form-urlencoded` cadre de l'en-tête.

Cette rubrique fournit des exemples de recherche à l'aide POST des types de ressources suivants.

- Âge : L'âge n'est pas un type de ressource défini dansFHIR. L'âge est plutôt saisi dans le cadre du type de ressource Patient. Pour rechercher un groupe de patients en fonction d'un âge ou d'une tranche d'âge spécifique, utilisez unthe section called “Comparateurs de recherche pris en charge”. Pour plus de détails, voir Type de ressource : Patient dans l'index de FHIR documentation.
- État : ce type de ressource contient des informations relatives à des concepts cliniques tels qu'un diagnostic, des situations, un état clinique et des problèmes devenus préoccupants. Pour en savoir plus, voir Type de ressource : Condition dans l'index de FHIR documentation. HealthLake crée de nouvelles conditions sur la base des documents trouvés dans le DocumentReference. Ces ajouts sont exclus par défaut lors d'une `POST` demande. Pour les inclure, vous devez spécifier un identifiant valide pour une ressource conditionnelle dans votre recherche.
- DocumentReference:Ce type de ressource est pris en charge par HealthLake. Ce type de ressource prend en charge le référencement de documents de tout type. Pour en savoir plus, voir Type de ressource : DocumentReference dans l'index de la FHIR documentation. HealthLake fournit également un traitement intégré du langage naturel (NLP) des documents trouvés dans le DocumentReference. Pour en savoir plus, consultez Utilisation de la génération automatique de ressources basée sur le traitement du langage naturel (NLP) du type de FHIR DocumentReference ressource dans AWS HealthLake.
- Emplacement : Ce type de ressource inclut à la fois les sites secondaires (un lieu utilisé pour les soins de santé sans désignation ni autorisation préalable) et les sites dédiés officiellement désignés. Pour plus de détails, voir Type de ressource : Emplacement dans l'index de FHIR documentation.
- Observation : mesures et assertions simples faites à propos d'un patient, d'un appareil ou d'un autre sujet. HealthLake crée de nouvelles ressources d'observation à partir des documents trouvés dans la DocumentReference ressource. Pour en savoir plus sur la façon de HealthLake créer de

nouvelles ressources, consultezUtilisation de la génération automatique de ressources basée sur le traitement du langage naturel (NLP) du type de FHIR DocumentReference ressource dans AWS HealthLake. Ces ajouts sont exclus par défaut lors d'une POST demande. Pour les inclure, vous devez spécifier un identifiant valide pour une ressource d'observation dans votre recherche. Pour en savoir plus, voir Type de ressource : Observation dans l'index de la FHIR documentation.

Chaque onglet présente des exemples de recherche sur le type de ressource spécifié. Il inclut un exemple de la manière de spécifier la demande dans le corps de la demande.

Age

Utilisez ce qui suit pour effectuer une demande de recherche POST basée sur le type de `Patient` ressource. Cette recherche utilise le `eq` comparateur de recherche pour rechercher des personnes nées en 1997.

Vous devez spécifier une demande URL et un corps de demande. Voici un exemple de demandeURL.

```
POST https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/
Patient/_search
```

Pour spécifier l'année 1997 dans la recherche, vous devez ajouter l'élément suivant dans le corps de la demande.

```
birthdate=eq1997
```

Réponse d'JSON

En cas de succès, vous obtiendrez un code de `200` HTTP réponse et une JSON réponse similaire.

Condition

Utilisez ce qui suit pour effectuer une POST demande sur le type de `Condition` ressource. Cette recherche permet de trouver les emplacements de votre banque de HealthLake données qui contiennent le code médical`72892002`.

Vous devez spécifier une demande URL et un corps de demande. Voici un exemple de demandeURL.

```
POST https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/
Condition/_search
```

Pour spécifier le code médical que vous souhaitez rechercher, vous devez ajouter cet JSON élément dans le corps de la demande.

```
code=72892002
```

Réponse d'JSON

En cas de succès, vous recevrez un code de 200 HTTP réponse. La JSON réponse suivante a été tronquée pour des raisons de clarté.

```
{
 "resourceType": "Bundle",
 "type": "searchset",
 "entry": [{
   "resource": {
    "resourceType": "Condition",
    "id": "0063326c-6b42-4d13-af2f-1efe0a65f016",
    "meta": {
     "lastUpdated": "2022-08-23T00:22:49.681Z"
    },
    "clinicalStatus": {
     "coding": [{
      "system": "http://terminology.hl7.org/CodeSystem/condition-clinical",
      "code": "resolved"
     }]
    },
    "verificationStatus": {
     "coding": [{
      "system": "http://terminology.hl7.org/CodeSystem/condition-ver-status",
      "code": "confirmed"
     }]
    },
    "code": {
     "coding": [{
      "system": "http://snomed.info/sct",
      "code": "72892002",
      "display": "Normal pregnancy"
     }],
     "text": "Normal pregnancy"
```

```
    },
    "subject": {
     "reference": "Patient/5fc0070a-696a-4855-94a9-175f1c641a33"
    },
    "encounter": {
     "reference": "Encounter/44078ab9-7ac7-4731-9ac8-4b3ff21a7bdb"
    },
    "onsetDateTime": "2019-08-15T01:19:17-07:00",
    "abatementDateTime": "2020-03-26T01:19:17-07:00",
    "recordedDate": "2019-08-15T01:19:17-07:00"
   },
   "search": {
    "mode": "match"
   }
  },
  {
   "resource": {
    "resourceType": "Condition",
    "id": "d00afdb2-1d2c-44fe-9f3b-033c0fe751a3",
    "meta": {
     "lastUpdated": "2022-08-23T00:20:47.100Z"
    },
    "clinicalStatus": {
     "coding": [{
      "system": "http://terminology.hl7.org/CodeSystem/condition-clinical",
      "code": "resolved"
     }]
    },
    "verificationStatus": {
     "coding": [{
      "system": "http://terminology.hl7.org/CodeSystem/condition-ver-status",
      "code": "confirmed"
     }]
    },
    "code": {
     "coding": [{
      "system": "http://snomed.info/sct",
      "code": "72892002",
      "display": "Normal pregnancy"
     }],
     "text": "Normal pregnancy"
    },
    "subject": {
     "reference": "Patient/d0a5cd1e-8da7-41bd-9b2f-41eef45246e5"
```

```
    },
    "encounter": {
     "reference": "Encounter/73758e67-4aaf-4e80-982b-8821f0b6fdfb"
    },
    "onsetDateTime": "2019-06-13T20:37:40-07:00",
    "abatementDateTime": "2020-01-23T19:37:40-08:00",
    "recordedDate": "2019-06-13T20:37:40-07:00"
   },
   "search": {
    "mode": "match"
   }
  }
 ]
}
```

DocumentReference

Pour voir les résultats du traitement HealthLake du langage naturel intégré (NLP) lorsque vous faites une POST demande sur le type de `DocumentReference` ressource, formatez une demande comme suit.

```
POST https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/
DocumentReference/_search
```

Pour spécifier l' DocumentReference élément auquel vous souhaitez faire référence, consultezParamètres de recherche. Vous les spécifierez dans le corps de la demande sous la formeJSON.

```
_lastUpdated=le2021-12-19&infer-icd10cm-entity-text-concept-score;=streptococcal|
0.6&infer-rxnorm-entity-text-concept-score=Amoxicillin|0.8
```

Cette chaîne de requête utilise plusieurs paramètres de recherche pour effectuer une recherche sur les opérations Amazon Comprehend API Medical utilisées pour générer les résultats NLP médicaux intégrés.

Location

Utilisez ce qui suit pour effectuer une POST demande sur le type de `Location` ressource. Cette recherche permet de trouver dans votre HealthLake banque de données les emplacements dont l'adresse contient le nom de la ville de Boston.

Vous devez spécifier une demande URL et un corps de demande. Voici un exemple de demandeURL.

```
POST https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/
Location/_search
```

Pour spécifier Boston dans la recherche, ajoutez l'élément suivant dans le corps de la requête :

```
address=Boston
```

Réponse d'JSON

En cas de succès, vous recevrez un code de 200 HTTP réponse. La JSON réponse a été tronquée pour des raisons de clarté.


```
{
 "resourceType": "Bundle",
 "type": "searchset",
 "entry": [{
   "resource": {
    "resourceType": "Location",
    "id": "0a6903c7-25c5-4ae4-8354-be88f9c5f2ee",
    "meta": {
     "lastUpdated": "2022-08-23T00:24:24.570Z"
    },
    "status": "active",
    "name": "BRIGHAM AND WOMEN'S HOSPITAL",
    "telecom": [{
     "system": "phone",
     "value": "6177325500"
    }],
    "address": {
     "line": [
      "75 FRANCIS STREET"
     ],
     "city": "BOSTON",
     "state": "MA",
     "postalCode": "02115",
     "country": "US"
    },
    "position": {
     "longitude": -71.020173,
```

```
     "latitude": 42.33196
    },
    "managingOrganization": {
     "reference": "Organization/27379046-608b-32f0-9df7-8c833cf5d11d",
     "display": "BRIGHAM AND WOMEN'S HOSPITAL"
    }
   },
   "search": {
    "mode": "match"
   }
  },

  {
   "resource": {
    "resourceType": "Location",
    "id": "ca5e7f65-4eb5-4bff-9a6f-07bc80acf8d0",
    "meta": {
     "lastUpdated": "2022-08-23T00:20:47.100Z"
    },
    "status": "active",
    "name": "BETH ISRAEL DEACONESS MEDICAL CENTER",
    "telecom": [{
     "system": "phone",
     "value": "6176677000"
    }],
    "address": {
     "line": [
      "330 BROOKLINE AVENUE"
     ],
     "city": "BOSTON",
     "state": "MA",
     "postalCode": "02215",
     "country": "US"
    },
    "position": {
     "longitude": -71.020173,
     "latitude": 42.33196
    },
    "managingOrganization": {
     "reference": "Organization/cb6a50e0-af76-3758-99ad-3200ede03fff",
     "display": "BETH ISRAEL DEACONESS MEDICAL CENTER"
    }
   },
   "search": {
```

```
    "mode": "match"
   }
  }
 ]
}
```

Observation

Utilisez ce qui suit pour effectuer une demande de recherche POST basée sur le type de `Observation` ressource. Cette recherche utilise le paramètre `value-concept` de recherche pour rechercher le code médical,`266919005`. Ce statut indique`Never smoker`.

Vous devez spécifier une demande URL et un corps de demande. Voici un exemple de demandeURL.

```
POST https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/
Observation/_search
```

Pour spécifier le statut`Never smoker`, `value-concept=266919005` définissez-le dans le corps duJSON.

```
value-concept=266919005
```

Réponse d'JSON

En cas de succès, vous recevrez un code de `200` HTTP réponse. La JSON réponse suivante a été tronquée pour des raisons de clarté.

```
{
 "resourceType": "Bundle",
 "type": "searchset",
 "link": [{
  "relation": "next",
  "url": "https://healthlake.us-west-2.amazonaws.com/
datastore/3651c6d3c1e81e785adba06b710b52a9/r4/Observation?value-
concept=266919005&=AAMA-
EFRSURBSGlpcGIyN250ZG9WRXVnTTFOdmtxQk9Bb3Y0YjhVcVdUMGV0eVozNmdjQU9nRjRNUUtscjhCZ1NMUG84VGNqN
 }],
 "entry": [{
   "resource": {
    "resourceType": "Observation",
    "id": "000038e0-71c6-4cc0-9c6c-50c8b1c53309",
```

```
    "meta": {
     "lastUpdated": "2022-11-03T01:02:38.981Z"
    },
    "status": "final",
    "category": [{
     "coding": [{
      "system": "http://terminology.hl7.org/CodeSystem/observation-category",
      "code": "survey",
      "display": "survey"
     }]
    }],
    "code": {
     "coding": [{
      "system": "http://loinc.org",
      "code": "72166-2",
      "display": "Tobacco smoking status NHIS"
     }],
     "text": "Tobacco smoking status NHIS"
    },
    "subject": {
     "reference": "Patient/598c9d7a-0494-448e-a81e-d50e3606e8db"
    },
    "encounter": {
     "reference": "Encounter/86bdee4a-2aa9-474a-b43f-6237cd68e512"
    },
    "effectiveDateTime": "2019-12-11T19:44:57-08:00",
    "issued": "2019-12-11T19:44:57.438-08:00",
    "valueCodeableConcept": {
     "coding": [{
      "system": "http://snomed.info/sct",
      "code": "266919005",
      "display": "Never smoker"
     }],
     "text": "Never smoker"
    }
   },
   "search": {
    "mode": "match"
   }
  },

  {
   "resource": {
    "resourceType": "Observation",
```

```
    "id": "0c2f6260-e671-4cfd-ac3d-e75f073fa3cd",
    "meta": {
     "lastUpdated": "2022-11-03T01:05:21.488Z"
    },
    "status": "final",
    "category": [{
     "coding": [{
      "system": "http://terminology.hl7.org/CodeSystem/observation-category",
      "code": "survey",
      "display": "survey"
     }]
    }],
    "code": {
     "coding": [{
      "system": "http://loinc.org",
      "code": "72166-2",
      "display": "Tobacco smoking status NHIS"
     }],
     "text": "Tobacco smoking status NHIS"
    },
    "subject": {
     "reference": "Patient/89d9a9b7-9720-4881-a2ab-d7907544b26f"
    },
    "encounter": {
     "reference": "Encounter/8ebba7b0-fdfc-4ec1-a9aa-907cccf60925"
    },
    "effectiveDateTime": "2018-11-17T03:59:36-08:00",
    "issued": "2018-11-17T03:59:36.550-08:00",
    "valueCodeableConcept": {
     "coding": [{
      "system": "http://snomed.info/sct",
      "code": "266919005",
      "display": "Never smoker"
     }],
     "text": "Never smoker"
    }
   },
   "search": {
    "mode": "match"
   }
  }
 ]
}
```

Recherche à l'aide GET d'exemples

Vous pouvez effectuer une recherche dans un magasin de HealthLake données en effectuant des GET demandes. HealthLake prend uniquement en charge la fourniture de paramètres de requête dans le cadre de la requêteURI, et non dans le corps d'une requête.

> **Note**
>
> Pour les requêtes impliquant des informations personnellement identifiables (PII) ou des informations de santé protégées (PHI), il est recommandé d'utiliser les `POST` demandes. Dans une `POST` demande, PII ou PHI est ajouté dans le corps de la demande et est crypté pendant le transit.

Cette rubrique fournit des exemples de recherche à l'`GET`aide des types de ressources pris en charge dans HealthLake.

- Âge : L'âge n'est pas un type de ressource défini dansFHIR. L'âge est plutôt saisi dans le cadre du type de ressource du patient. Pour rechercher un groupe de patients en fonction d'un âge ou d'une tranche d'âge spécifique, vous devez utiliser unthe section called “Comparateurs de recherche pris en charge”. Pour plus de détails, voir Type de ressource : Patient dans l'index de FHIR documentation.
- État : ce type de ressource contient des informations relatives à des concepts cliniques tels qu'un diagnostic, des situations, un état clinique et des problèmes devenus préoccupants. Pour en savoir plus, voir Type de ressource : Condition dans l'index de FHIR documentation. HealthLake crée de nouvelles conditions sur la base des documents trouvés dans le DocumentReference. Ces ajouts sont exclus par défaut lors d'une `POST` demande. Pour les inclure, vous devez spécifier un identifiant valide pour une ressource conditionnelle dans votre recherche.
- DocumentReference:Ce type de ressource est pris en charge par HealthLake. Ce type de ressource prend en charge le référencement de documents de tout type. Pour en savoir plus, voir Type de ressource : DocumentReference dans l'index de la FHIR documentation. HealthLake fournit également un traitement intégré du langage naturel (NLP) des documents trouvés dans le DocumentReference. Pour en savoir plus, consultez Utilisation de la génération automatique de ressources basée sur le traitement du langage naturel (NLP) du type de FHIR DocumentReference ressource dans AWS HealthLake.
- Emplacement : Ce type de ressource inclut à la fois les sites secondaires (un lieu utilisé pour les soins de santé sans désignation ni autorisation préalable) et les sites dédiés officiellement

désignés. Pour plus de détails, voir Type de ressource : Emplacement dans l'index de FHIR documentation.

- Observation : mesures et assertions simples faites à propos d'un patient, d'un appareil ou d'un autre sujet. HealthLake crée de nouvelles ressources d'observation à partir des documents trouvés dans la DocumentReference ressource. Pour en savoir plus sur la HealthLake création de nouvelles ressources, consultezUtilisation de la génération automatique de ressources basée sur le traitement du langage naturel (NLP) du type de FHIR DocumentReference ressource dans AWS HealthLake. Ces ajouts sont exclus par défaut lors d'une `POST` demande. Pour les inclure, vous devez spécifier un identifiant valide pour une ressource d'observation dans votre recherche. Pour en savoir plus, voir Type de ressource : Observation dans l'index de la FHIR documentation.

Chaque onglet montre un exemple de recherche sur le type de ressource spécifié. Il inclut un exemple de la manière de spécifier la demande dans URI le et la JSON réponse associée.

Age

Utilisez ce qui suit pour effectuer une demande de recherche `GET` basée sur le type de `Patient` ressource. Cette recherche utilise le `eq` comparateur de recherche pour rechercher des personnes nées en 1997.

```
GET https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4//
Patient?birthdate=eq1997
```

Réponse d'JSON

En cas de succès, vous recevrez un code de `200` HTTP réponse.

Condition

Utilisez ce qui suit pour effectuer une `GET` demande sur le type de `Condition` ressource. Cette recherche permet de trouver les emplacements de votre banque de HealthLake données qui contiennent le code médical`72892002`.

Vous devez spécifier une demande URL et un corps de demande. Voici un exemple de demandeURL.

```
GET https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/
Condition?code=72892002
```

Réponse d'JSON

En cas de succès, vous recevrez un code de 200 HTTP réponse. La JSON réponse suivante a été tronquée pour des raisons de clarté.


```
{
 "resourceType": "Bundle",
 "type": "searchset",
 "entry": [{
   "resource": {
    "resourceType": "Condition",
    "id": "0063326c-6b42-4d13-af2f-1efe0a65f016",
    "meta": {
     "lastUpdated": "2022-08-23T00:22:49.681Z"
    },
    "clinicalStatus": {
     "coding": [{
      "system": "http://terminology.hl7.org/CodeSystem/condition-clinical",
      "code": "resolved"
     }]
    },
    "verificationStatus": {
     "coding": [{
      "system": "http://terminology.hl7.org/CodeSystem/condition-ver-status",
      "code": "confirmed"
     }]
    },
    "code": {
     "coding": [{
      "system": "http://snomed.info/sct",
      "code": "72892002",
      "display": "Normal pregnancy"
     }],
     "text": "Normal pregnancy"
    },
    "subject": {
     "reference": "Patient/5fc0070a-696a-4855-94a9-175f1c641a33"
    },
    "encounter": {
     "reference": "Encounter/44078ab9-7ac7-4731-9ac8-4b3ff21a7bdb"
    },
    "onsetDateTime": "2019-08-15T01:19:17-07:00",
    "abatementDateTime": "2020-03-26T01:19:17-07:00",
    "recordedDate": "2019-08-15T01:19:17-07:00"
   },
```

```
  "search": {
   "mode": "match"
  }
 },
 {
  "resource": {
   "resourceType": "Condition",
   "id": "d00afdb2-1d2c-44fe-9f3b-033c0fe751a3",
   "meta": {
    "lastUpdated": "2022-08-23T00:20:47.100Z"
   },
   "clinicalStatus": {
    "coding": [{
     "system": "http://terminology.hl7.org/CodeSystem/condition-clinical",
     "code": "resolved"
    }]
   },
   "verificationStatus": {
    "coding": [{
     "system": "http://terminology.hl7.org/CodeSystem/condition-ver-status",
     "code": "confirmed"
    }]
   },
   "code": {
    "coding": [{
     "system": "http://snomed.info/sct",
     "code": "72892002",
     "display": "Normal pregnancy"
    }],
    "text": "Normal pregnancy"
   },
   "subject": {
    "reference": "Patient/d0a5cd1e-8da7-41bd-9b2f-41eef45246e5"
   },
   "encounter": {
    "reference": "Encounter/73758e67-4aaf-4e80-982b-8821f0b6fdfb"
   },
   "onsetDateTime": "2019-06-13T20:37:40-07:00",
   "abatementDateTime": "2020-01-23T19:37:40-08:00",
   "recordedDate": "2019-06-13T20:37:40-07:00"
  },
  "search": {
   "mode": "match"
  }
```

```
  }
 ]
}
```

DocumentationReference

Cet exemple montre comment créer une demande de recherche sur le type de `DocumentReference` ressource pour les patients ayant reçu un diagnostic de streptocoque et à qui l'amoxicilline a également été prescrite.

```
GET https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/
DocumentReference?_lastUpdated=le2021-12-19&infer-icd10cm-entity-text-concept-
score;=streptococcal|0.6&infer-rxnorm-entity-text-concept-score=Amoxicillin|0.8
```

En cas de succès, vous obtiendrez la JSON réponse suivante.

```
{
    "resourceType": "Bundle",
    "type": "searchset",
    "entry": [
    {
       "resource": {
       "resourceType": "DocumentReference",
       "id": "985c3e94-4219-4c79-97a1-c94694525e24",
       "meta": {
          "lastUpdated": "2020-11-23T06:09:10.719Z"
        },
        "extension": [
          {
            "url": "http://healthlake.amazonaws.com/aws-cm/",
            "extension": [
              {
                "url": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/",
                "extension": [
                  {
                    "url": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/raw-
response",
                    "valueString": "{Entities: [{Id: 0,Text: otitis media,Category:
 MEDICAL_CONDITION,Type: DX_NAME,Score: 0.9815994,BeginOffset: 151,EndOffset:
 163,Attributes: [],Traits: [{Name: DIAGNOSIS,Score: 0.95042425}],ICD10CMConcepts:
 [{Description: Otitis media, unspecified, unspecified ear,Code: H66.90,Score:
 0.7176407}, {Description: Otitis media, unspecified,Code: H66.9,Score:
 0.6930445}, {Description: Otitis media, unspecified, left ear,Code: H66.92,Score:
```

```
 0.688161}, {Description: Otitis media, unspecified, bilateral,Code: H66.93,Score:
 0.6748094}, {Description: Otitis media, unspecified, right ear,Code:
 H66.91,Score: 0.6645618}]}, {Id: 1,Text: streptococcal sore throat,Category:
 MEDICAL_CONDITION,Type: DX_NAME,Score: 0.92208487,BeginOffset: 461,EndOffset:
 486,Attributes: [],Traits: [],ICD10CMConcepts: [{Description: Streptococcal
 pharyngitis,Code: J02.0,Score: 0.55638546}, {Description: Acute streptococcal
 tonsillitis, unspecified,Code: J03.00,Score: 0.53159785}, {Description:
 Streptococcal sepsis, unspecified,Code: A40.9,Score: 0.51865804}, {Description:
 Acute pharyngitis, unspecified,Code: J02.9,Score: 0.45085955}, {Description:
 Streptococcal infection, unspecified site,Code: A49.1,Score: 0.41550553}]},
 {Id: 3,Text: disorder,Category: MEDICAL_CONDITION,Type: DX_NAME,Score:
 0.9191257,BeginOffset: 488,EndOffset: 496,Attributes: [],Traits: [{Name:
 DIAGNOSIS,Score: 0.93372077}],ICD10CMConcepts: [{Description: Parkinson's
 disease,Code: G20,Score: 0.6959145}, {Description: Illness, unspecified,Code:
 R69,Score: 0.68428487}, {Description: Disorder of bone, unspecified,Code:
 M89.9,Score: 0.6542605}, {Description: Unspecified mental disorder due to known
 physiological condition,Code: F09,Score: 0.6240179}, {Description: Mental disorder,
 not otherwise specified,Code: F99,Score: 0.61046}]}],ModelVersion: 0.1.0}"
                  },
                  {
                    "url": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/
model-version",
                    "valueString": "0.1.0"
                  },
                  {
                    "url": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/aws-
cm-icd10-entity",
                    "extension": [
                      {
                        "url": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/
aws-cm-icd10-entity-id",
                        "valueInteger": 0
                      },
                      {
                        "url": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/
aws-cm-icd10-entity-text",
                        "valueString": "otitis media"
                      },
                      {
                        "url": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/
aws-cm-icd10-entity-begin-offset",
                        "valueInteger": 151
                      },
                      {
```

```
                        "url": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/
aws-cm-icd10-entity-end-offset",
                        "valueInteger": 163
                      },
                      {
                        "url": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/
aws-cm-icd10-entity-score",
                        "valueDecimal": 0.9815994
                      },
                      {
                        "url": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/
aws-cm-icd10-entity-ConceptList",
                        "extension": [
                          {
                            "url": "http://healthlake.amazonaws.com/aws-cm/infer-
icd10/aws-cm-icd10-entity-Concept",
                            "extension": [
                              {
                                "url": "http://healthlake.amazonaws.com/aws-cm/
infer-icd10/aws-cm-icd10-entity-Concept-Code",
                                "valueString": "H66.90"
                              },
                              {
                                "url": "http://healthlake.amazonaws.com/aws-cm/
infer-icd10/aws-cm-icd10-entity-Concept-Description",
                                "valueString": "Otitis media, unspecified,
 unspecified ear"
                              },
                              {
                                "url": "http://healthlake.amazonaws.com/aws-cm/
infer-icd10/aws-cm-icd10-entity-Concept-Score",
                                "valueDecimal": 0.7176407
                              }
                            ]
                          },
                          {
                            "url": "http://healthlake.amazonaws.com/aws-cm/infer-
icd10/aws-cm-icd10-entity-Concept",
                            "extension": [
                              {
                                "url": "http://healthlake.amazonaws.com/aws-cm/
infer-icd10/aws-cm-icd10-entity-Concept-Code",
                                "valueString": "H66.9"
                              },
```

```
                                {
                                  "url": "http://healthlake.amazonaws.com/aws-cm/
infer-icd10/aws-cm-icd10-entity-Concept-Description",
                                  "valueString": "Otitis media, unspecified"
                                },
                                {
                                  "url": "http://healthlake.amazonaws.com/aws-cm/
infer-icd10/aws-cm-icd10-entity-Concept-Score",
                                  "valueDecimal": 0.6930445
                                }
                              ]
                            },
                            {
                              "url": "http://healthlake.amazonaws.com/aws-cm/infer-
icd10/aws-cm-icd10-entity-Concept",
                              "extension": [
                                {
                                  "url": "http://healthlake.amazonaws.com/aws-cm/
infer-icd10/aws-cm-icd10-entity-Concept-Code",
                                  "valueString": "H66.92"
                                }
                              ]
                            }
```

Location

Utilisez ce qui suit pour effectuer une GET demande sur le type de Location ressource. Cette recherche permet de trouver dans votre HealthLake banque de données les emplacements dont l'adresse contient le nom de la ville de Boston.

```
GET https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4//
Location?address=boston
```

Réponse d'JSON

En cas de succès, vous recevrez un code de 200 HTTP réponse. La JSON réponse a été tronquée pour des raisons de clarté.

```
{
    "resourceType": "Bundle",
    "type": "searchset",
    "entry": [
```

```
        {
            "resource": {
                "resourceType": "Location",
                "id": "0a6903c7-25c5-4ae4-8354-be88f9c5f2ee",
                "meta": {
                    "lastUpdated": "2022-08-23T00:24:24.570Z"
                },
                "status": "active",
                "name": "BRIGHAM AND WOMEN'S HOSPITAL",
                "telecom": [
                    {
                        "system": "phone",
                        "value": "6177325500"
                    }
                ],
                "address": {
                    "line": [
                        "75 FRANCIS STREET"
                    ],
                    "city": "BOSTON",
                    "state": "MA",
                    "postalCode": "02115",
                    "country": "US"
                },
                "position": {
                    "longitude": -71.020173,
                    "latitude": 42.33196
                },
                "managingOrganization": {
                    "reference":
"Organization/27379046-608b-32f0-9df7-8c833cf5d11d",
                    "display": "BRIGHAM AND WOMEN'S HOSPITAL"
                }
            },
            "search": {
                "mode": "match"
            }
        },
        {
            "resource": {
                "resourceType": "Location",
                "id": "3cc3ad99-e0ff-48b4-b277-052abfc41058",
                "meta": {
                    "lastUpdated": "2022-08-23T00:19:37.029Z"
```

```
                },
                "status": "active",
                "name": "NEW ENGLAND BAPTIST HOSPITAL",
                "telecom": [
                    {
                        "system": "phone",
                        "value": "6177545800"
                    }
                ],
                "address": {
                    "line": [
                        "125 PARKER HILL AVENUE"
                    ],
                    "city": "BOSTON",
                    "state": "MA",
                    "postalCode": "02120",
                    "country": "US"
                },
                "position": {
                    "longitude": -71.020173,
                    "latitude": 42.33196
                },
                "managingOrganization": {
                    "reference": "Organization/9a7149fa-49fc-3c87-b935-
d29c55808717",
                    "display": "NEW ENGLAND BAPTIST HOSPITAL"
                }
            },
            "search": {
                "mode": "match"
            }
        },
        {
            "resource": {
                "resourceType": "Location",
                "id": "3f956715-3890-4235-85be-3fba5e3488ee",
                "meta": {
                    "lastUpdated": "2022-08-23T00:23:38.981Z"
                },
                "status": "active",
                "name": "MASSACHUSETTS GENERAL HOSPITAL",
                "telecom": [
                    {
                        "system": "phone",
```

```
                        "value": "6177262000"
                    }
                ],
                "address": {
                    "line": [
                        "55 FRUIT STREET"
                    ],
                    "city": "BOSTON",
                    "state": "MA",
                    "postalCode": "02114",
                    "country": "US"
                },
                "position": {
                    "longitude": -71.020173,
                    "latitude": 42.33196
                },
                "managingOrganization": {
                    "reference": "Organization/d78e84ec-30aa-3bba-a33a-
f29a3a454662",
                    "display": "MASSACHUSETTS GENERAL HOSPITAL"
                }
            },
            "search": {
                "mode": "match"
            }
        },
        {
            "resource": {
                "resourceType": "Location",
                "id": "6cc07b51-7287-443c-b772-c864f7831e13",
                "meta": {
                    "lastUpdated": "2022-08-23T00:21:11.045Z"
                },
                "status": "active",
                "name": "TUFTS MEDICAL CENTER",
                "telecom": [
                    {
                        "system": "phone",
                        "value": "6176365000"
                    }
                ],
                "address": {
                    "line": [
                        "800 WASHINGTON STREET"
```

```
                    ],
                    "city": "BOSTON",
                    "state": "MA",
                    "postalCode": "02111",
                    "country": "US"
                },
                "position": {
                    "longitude": -71.020173,
                    "latitude": 42.33196
                },
                "managingOrganization": {
                    "reference": "Organization/b7175ab4-
bde5-3848-891b-579bccb77c7c",
                    "display": "TUFTS MEDICAL CENTER"
                }
            },
            "search": {
                "mode": "match"
            }
        },
        {
            "resource": {
                "resourceType": "Location",
                "id": "8101300f-f685-49e7-b428-43b7855c39ee",
                "meta": {
                    "lastUpdated": "2022-08-23T00:22:06.474Z"
                },
                "status": "active",
                "name": "BOSTON CHILDREN'S HOSPITAL",
                "telecom": [
                    {
                        "system": "phone",
                        "value": "6177356000"
                    }
                ],
                "address": {
                    "line": [
                        "300 LONGWOOD AVENUE"
                    ],
                    "city": "BOSTON",
                    "state": "MA",
                    "postalCode": "02115",
                    "country": "US"
                },
```

```
                "position": {
                    "longitude": -71.020173,
                    "latitude": 42.33196
                },
                "managingOrganization": {
                    "reference": "Organization/d7b11827-25f2-350b-
bcd8-939fc59851b0",
                    "display": "BOSTON CHILDREN'S HOSPITAL"
                }
            },
            "search": {
                "mode": "match"
            }
        },
        {
            "resource": {
                "resourceType": "Location",
                "id": "8b7641d3-6997-48bb-bd60-23e35dfaae9d",
                "meta": {
                    "lastUpdated": "2022-08-23T00:20:47.099Z"
                },
                "status": "active",
                "name": "BRIGHAM AND WOMEN'S FAULKNER HOSPITAL",
                "telecom": [
                    {
                        "system": "phone",
                        "value": "6179837000"
                    }
                ],
                "address": {
                    "line": [
                        "1153 CENTRE STREET"
                    ],
                    "city": "BOSTON",
                    "state": "MA",
                    "postalCode": "02130",
                    "country": "US"
                },
                "position": {
                    "longitude": -71.020173,
                    "latitude": 42.33196
                },
                "managingOrganization": {
```

```
                    "reference": "Organization/d733d4a9-080d-3593-
b910-2366e652b7ea",
                    "display": "BRIGHAM AND WOMEN'S FAULKNER HOSPITAL"
                }
            },
            "search": {
                "mode": "match"
            }
        },
        {
            "resource": {
                "resourceType": "Location",
                "id": "998ef80b-7b58-4dc3-99ac-c440ec9e282d",
                "meta": {
                    "lastUpdated": "2022-08-23T00:21:11.046Z"
                },
                "status": "active",
                "name": "BRIGHAM AND WOMEN'S FAULKNER HOSPITAL",
                "telecom": [
                    {
                        "system": "phone",
                        "value": "6179837000"
                    }
                ],
                "address": {
                    "line": [
                        "1153 CENTRE STREET"
                    ],
                    "city": "BOSTON",
                    "state": "MA",
                    "postalCode": "02130",
                    "country": "US"
                },
                "position": {
                    "longitude": -71.020173,
                    "latitude": 42.33196
                },
                "managingOrganization": {
                    "reference": "Organization/d733d4a9-080d-3593-
b910-2366e652b7ea",
                    "display": "BRIGHAM AND WOMEN'S FAULKNER HOSPITAL"
                }
            },
            "search": {
```

```
                "mode": "match"
            }
        },
        {
            "resource": {
                "resourceType": "Location",
                "id": "c454bed3-7013-4376-81cf-4f49342f1402",
                "meta": {
                    "lastUpdated": "2022-08-23T00:24:24.573Z"
                },
                "status": "active",
                "name": "MASSACHUSETTS GENERAL HOSPITAL",
                "telecom": [
                    {
                        "system": "phone",
                        "value": "6177262000"
                    }
                ],
                "address": {
                    "line": [
                        "55 FRUIT STREET"
                    ],
                    "city": "BOSTON",
                    "state": "MA",
                    "postalCode": "02114",
                    "country": "US"
                },
                "position": {
                    "longitude": -71.020173,
                    "latitude": 42.33196
                },
                "managingOrganization": {
                    "reference": "Organization/d78e84ec-30aa-3bba-a33a-
f29a3a454662",
                    "display": "MASSACHUSETTS GENERAL HOSPITAL"
                }
            },
            "search": {
                "mode": "match"
            }
        },
        {
            "resource": {
                "resourceType": "Location",
```

```
                "id": "ca5e7f65-4eb5-4bff-9a6f-07bc80acf8d0",
                "meta": {
                    "lastUpdated": "2022-08-23T00:20:47.100Z"
                },
                "status": "active",
                "name": "BETH ISRAEL DEACONESS MEDICAL CENTER",
                "telecom": [
                    {
                        "system": "phone",
                        "value": "6176677000"
                    }
                ],
                "address": {
                    "line": [
                        "330 BROOKLINE AVENUE"
                    ],
                    "city": "BOSTON",
                    "state": "MA",
                    "postalCode": "02215",
                    "country": "US"
                },
                "position": {
                    "longitude": -71.020173,
                    "latitude": 42.33196
                },
                "managingOrganization": {
                    "reference": "Organization/cb6a50e0-
af76-3758-99ad-3200ede03fff",
                    "display": "BETH ISRAEL DEACONESS MEDICAL CENTER"
                }
            },
            "search": {
                "mode": "match"
            }
        }
    ]
}
```

Observation

Utilisez ce qui suit pour effectuer une demande de recherche GET basée sur le type de `Observation` ressource. Cette recherche utilise le paramètre `value-concept` de recherche pour rechercher le code médical,`266919005`. Ce statut indique`Never smoker`.

Vous devez spécifier une requête URL et une chaîne de requête. Voici un exemple de demandeURL.

```
POST https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/
Observation?value-concept=266919005
```

Pour spécifier le statutNever smoker, définissez-le value-concept=266919005 comme chaîne de requête.

Réponse d'JSON

En cas de succès, vous recevrez un code de 200 HTTP réponse. La JSON réponse suivante a été tronquée pour des raisons de clarté.

```
{
 "resourceType": "Bundle",
 "type": "searchset",
 "link": [{
  "relation": "next",
  "url": "https://healthlake.us-west-2.amazonaws.com/
datastore/3651c6d3c1e81e785adba06b710b52a9/r4/Observation?value-
concept=266919005&=AAMA-
EFRSURBSGlpcGIyN250ZG9WRXVnTTFOdmtxQk9Bb3Y0YjhVcVdUMGV0eVozNmdjQU9nRjRNUUtscjhCZ1NMUG84VGNqN
 }],
 "entry": [{
   "resource": {
    "resourceType": "Observation",
    "id": "000038e0-71c6-4cc0-9c6c-50c8b1c53309",
    "meta": {
     "lastUpdated": "2022-11-03T01:02:38.981Z"
    },
    "status": "final",
    "category": [{
     "coding": [{
      "system": "http://terminology.hl7.org/CodeSystem/observation-category",
      "code": "survey",
      "display": "survey"
     }]
    }],
    "code": {
     "coding": [{
      "system": "http://loinc.org",
      "code": "72166-2",
```

```
      "display": "Tobacco smoking status NHIS"
     }],
     "text": "Tobacco smoking status NHIS"
    },
    "subject": {
     "reference": "Patient/598c9d7a-0494-448e-a81e-d50e3606e8db"
    },
    "encounter": {
     "reference": "Encounter/86bdee4a-2aa9-474a-b43f-6237cd68e512"
    },
    "effectiveDateTime": "2019-12-11T19:44:57-08:00",
    "issued": "2019-12-11T19:44:57.438-08:00",
    "valueCodeableConcept": {
     "coding": [{
      "system": "http://snomed.info/sct",
      "code": "266919005",
      "display": "Never smoker"
     }],
     "text": "Never smoker"
    }
   },
   "search": {
    "mode": "match"
   }
  },

  {
   "resource": {
    "resourceType": "Observation",
    "id": "0c2f6260-e671-4cfd-ac3d-e75f073fa3cd",
    "meta": {
     "lastUpdated": "2022-11-03T01:05:21.488Z"
    },
    "status": "final",
    "category": [{
     "coding": [{
      "system": "http://terminology.hl7.org/CodeSystem/observation-category",
      "code": "survey",
      "display": "survey"
     }]
    }],
    "code": {
     "coding": [{
      "system": "http://loinc.org",
```

```
      "code": "72166-2",
      "display": "Tobacco smoking status NHIS"
     }],
     "text": "Tobacco smoking status NHIS"
    },
    "subject": {
     "reference": "Patient/89d9a9b7-9720-4881-a2ab-d7907544b26f"
    },
    "encounter": {
     "reference": "Encounter/8ebba7b0-fdfc-4ec1-a9aa-907cccf60925"
    },
    "effectiveDateTime": "2018-11-17T03:59:36-08:00",
    "issued": "2018-11-17T03:59:36.550-08:00",
    "valueCodeableConcept": {
     "coding": [{
      "system": "http://snomed.info/sct",
      "code": "266919005",
      "display": "Never smoker"
     }],
     "text": "Never smoker"
    }
   },
   "search": {
    "mode": "match"
   }
  }
 ]
}
```

Lire l'historique FHIR des ressources

L'FHIR`history`interaction permet de récupérer l'historique d'une FHIR ressource particulière dans un magasin de HealthLake données. Grâce à cette interaction, vous pouvez déterminer l'évolution du contenu d'une FHIR ressource au fil du temps. Il est également utile, en coordination avec les journaux d'audit, de voir l'état d'une ressource avant et après modification.

> **Note**
>
> FHIR`history`La ressource est activée par défaut pour tous les magasins de HealthLake données créés après le 25/10/2024. Si votre banque de données a été créée avant cette date, vous pouvez envoyer un ticket d'assistance pour activer FHIR `history` l'interaction.

Créez un dossier en utilisant AWS Support Center Console. Pour créer votre dossier, connectez-vous à votre dossier Compte AWS et choisissez Créer un dossier.

L'`history`interaction est réalisée à l'aide de la `HTTP GET` commande. Les FHIR interactions `create`, `update`, et `delete` aboutissent à une version historique de la ressource à enregistrer. HealthLake prend en charge les paramètres de recherche suivants pour l'FHIR`history`interaction.

HealthLake paramètres de recherche pris en charge pour FHIR **`history`** l'interaction

Paramètre de recherche	Description
`_count : integer`	Le nombre maximum de résultats de recherche sur une page. Le serveur renverra le nombre demandé ou le nombre maximum de résultats de recherche autorisés par défaut pour le magasin de données, le plus faible des deux étant retenu.
`_since : instant`	N'incluez que les versions de ressources créées à l'instant donné ou après.
`_at : date(Time)`	N'incluez que les versions de ressources qui étaient à jour à un moment donné pendant la période spécifiée dans la valeur de date et d'heure. Pour plus d'informations, consultez `date`la HL7FHIRRESTfulAPIdocumentation.

L'exemple suivant renvoie 100 résultats de recherche historiques par page pour une FHIR `Patient` ressource dans HealthLake. Pour afficher le URL chemin complet, faites défiler le curseur sur le bouton Copier. URLIl s'agit de la forme :

```
GET https://healthlake.region.amazonaws.com/datastore/datastore-id/r4/Patient/id/
_history?_count=100
```

Le contenu renvoyé par une interaction historique est contenu dans une FHIR ressource `Bundle`, dont le type est défini sur`history`. Il contient l'historique des versions spécifié, trié avec les versions

les plus anciennes en dernier, et inclut les ressources supprimées. Pour plus d'informations sur l'`history`interaction, consultez `history`la HL7 FHIR RESTful API documentation.

> **Note**
>
> Vous pouvez choisir de ne pas `history` utiliser certains types de FHIR ressources. Pour vous désinscrire, créez un dossier à l'aide de AWS Support Center Console. Pour créer votre dossier, connectez-vous à votre dossier Compte AWS et choisissez Créer un dossier.

Lire l'historique des ressources spécifiques à une version FHIR

L'FHIR`vread`interaction effectue une lecture spécifique à la version d'une ressource dans un magasin de HealthLake données. Grâce à cette interaction, vous pouvez visualiser le contenu d'une FHIR ressource tel qu'il était à un moment donné dans le passé.

HealthLake déclare qu'il prend en charge le versionnement `CapabilityStatement.rest.resource.versioning`pour chaque ressource prise en charge. Tous les magasins de HealthLake données incluent `Resource.meta.versionId` (`vid`) sur toutes les ressources.

Lorsque FHIR `history` l'interaction est activée (par défaut pour les magasins de données créés après le 25/10/2024 ou par demande pour les anciens magasins de données), la `Bundle` réponse inclut `vid` le dans le. `location` Dans l'exemple suivant, le `vid` s'affiche sous forme de numéro`1`. Pour voir l'exemple complet, voir Example bundle/bundle-response (). JSON

```
"response" : {
    "status" : "201 Created",
    "location" : "Patient/12423/_history/1",
    ...}
```

L'`vread`interaction est réalisée à l'aide de la `HTTP GET` commande. L'`vread`interaction suivante renvoie une instance unique avec le contenu spécifié pour la FHIR `Patient` ressource pour la version des métadonnées de ressource spécifiée par le`vid`. Pour afficher le URL chemin complet dans l'exemple suivant, faites défiler le curseur sur le bouton Copier. URLIl s'agit de la forme :

```
GET https://healthlake.region.amazonaws.com/datastore/datastore-id/r4/Patient/id/
_history/vid
```

> **Note**
>
> Si vous utilisez l'`history`interaction sans `vread` lire une FHIR ressource, renvoie HealthLake toujours la dernière version des métadonnées de la ressource.

Pour plus d'informations sur l'`vread`interaction, consultez `vread`la APIdocumentation HL7 FHIR Restful.

Obtenir les données des patients grâce à l'opération Patient $everything FHIR REST API

L'opération Patient $everything est utilisée pour interroger une ressource FHIR Patient ainsi que toute autre ressource liée à ce patient. Cette opération peut être utilisée pour permettre à un patient d'accéder à l'intégralité de son dossier ou pour qu'un fournisseur effectue un téléchargement massif de données relatives à un patient. HealthLake prend en charge $everything pour un identifiant de patient spécifique.

> **Note**
>
> L'opération Patient $everything est actuellement prise en charge sur les banques de données créées après le 27 février 2024.

Accédez à toutes les ressources relatives à un patient

Patient $everything est une REST API opération qui peut être invoquée comme indiqué dans les exemples ci-dessous.

GET Request

```
GET https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/
Patient/patient-id/$everything
```

> **Note**
>
> Les ressources en réponse sont triées par type de ressource et par identifiant de ressource. La réponse est toujours renseignée avec Bundle.TOTAL.

Paramètres du patient $everything

HealthLake prend en charge les paramètres de requête suivants

Paramètre	Détails
démarrer	Obtenez toutes les données du patient après une date de début spécifiée.
end	Obtenez toutes les données du patient avant une date de fin spécifiée.
since	Mettez à jour toutes les données des patients après une date spécifiée.
_type	Obtenez des données sur les patients pour des types de ressources spécifiques.
_compter	Obtenez les données du patient et spécifiez le format de page.

Example - Obtenez toutes les données du patient après une date de début spécifiée

Le patient $everything peut utiliser le `start` filtre pour n'interroger des données qu'après une date précise.

```
GET https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/
Patient/patient-id/$everything?start=2024-03-15T00:00:00.000Z
```

Example - Obtenez toutes les données du patient avant une date de fin spécifiée

Patient $everything peut utiliser le end filtre pour n'interroger que des données antérieures à une date précise.

```
GET https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/
Patient/patient-id/$everything?end=2024-03-15T00:00:00.000Z
```

Example - Mettez à jour toutes les données des patients après une date spécifiée

Le patient $everything peut utiliser le `since` filtre pour n'interroger que les données mises à jour après une date précise.

```
GET https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/
Patient/patient-id/$everything?since=2024-03-15T00:00:00.000Z
```

Example - Obtenez des données sur les patients pour des types de ressources spécifiques

Le patient $everything peut utiliser le `_type` filtre pour spécifier des types de ressources spécifiques à inclure dans la réponse. Plusieurs types de ressources peuvent être spécifiés dans une liste séparée par des virgules.

```
GET https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/
Patient/patient-id/$everything?_type=Observation,Condition
```

Example - Obtenez les données du patient et spécifiez le format de page

Le patient $everything peut utiliser le `_count` pour définir le format de page.

```
GET https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/
Patient/patient-id/$everything?_count=15
```

Tout **`start`** et **`end`** attributs du patient

HealthLake prend en charge les attributs de ressource suivants pour les paramètres de début et de fin de requête.

Ressource	Élément de ressource
Compte	Compte. servicePeriod.démarrer
AdverseEvent	AdverseEvent.date

Ressource	Élément de ressource
AllergyIntolerance	AllergyIntolerance.recordedDate
Rendez-vous	Rendez-vous.Start
AppointmentResponse	AppointmentResponse.démarrer
AuditEvent	AuditEvent.period.start
Base	Basic. Créé
BodyStructure	NON_ DATE
CarePlan	CarePlan.period.start
CareTeam	CareTeam.period.start
ChargeItem	ChargeItem. occurrenceDateTime, ChargeItem. occurrencePeriod.démarrer, ChargeItem. occurrenceTiming.événement
Demander	Réclamation. billablePeriod.démarrer
ClaimResponse	ClaimResponse.créé
ClinicalImpression	ClinicalImpression.date
Communication	Communication. Envoyée
CommunicationRequest	CommunicationRequest. occurrenceDateTime, CommunicationRequest. occurrencePeriod.démarrer

Ressource	Élément de ressource
Montage	Date de composition
Condition	État. recordedDate
Consentement	Consentement. dateTime
Couverture	Couverture.Period.Start
CoverageEligibilityRequest	CoverageEligibilityRequest.créé
CoverageEligibilityResponse	CoverageEligibilityResponse.créé
DetectedIssue	DetectedIssue.identifié
DeviceRequest	DeviceRequest.authoredOn
DeviceUseStatement	DeviceUseStatement.recordedOn
DiagnosticReport	DiagnosticReport.efficace
DocumentManifest	DocumentManifest.créé
DocumentReference	DocumentReference.context.period.start
Rencontre	Encounter.Period.Start

Ressource	Élément de ressource
EnrollmentRequest	EnrollmentRequest.créé
EpisodeOfCare	EpisodeOfCare.period.start
ExplanationOfBenefit	ExplanationOfBenefit. billablePeriod.démarrer
FamilyMemberHistory	NON_ DATE
Indicateur	Flag.Period.Start
Objectif	Objectif. statusDate
Groupe	NON_ DATE
ImagingStudy	ImagingStudy.démarré
Immunisation	Vaccination enregistrée
ImmunizationEvaluation	ImmunizationEvaluation.date
ImmunizationRecommendation	ImmunizationRecommendation.date
Facture	Date de facturation
Liste	Date de la liste

Ressource	Élément de ressource
MeasureReport	MeasureReport.period.start
Multimédia	Publié dans les médias
MedicationAdministration	MedicationAdministration.efficace
MedicationDispense	MedicationDispense.whenPrepared
MedicationRequest	MedicationRequest.authoredOn
MedicationStatement	MedicationStatement.dateAsserted
MolecularSequence	NON_ DATE
NutritionOrder	NutritionOrder.dateTime
Observation	Observation. Efficace
Patient	NON_ DATE
Personne	NON_ DATE
Procédure	Procédure. Exécutée
Provenance	Provenance. occurredPeriod.start, Provenance. occurredDateTime

Ressource	Élément de ressource
QuestionnaireResponse	QuestionnaireResponse.écrit
RelatedPerson	NON_ DATE
RequestGroup	RequestGroup.authoredOn
ResearchSubject	ResearchSubject.période
RiskAssessment	RiskAssessment. occurrenceDateTime, RiskAssessment. occurrencePeriod.démarrer
Planificateur	Horaire. planningHorizon
ServiceRequest	ServiceRequest.authoredOn
Spécimen	Spécimen. receivedTime
SupplyDelivery	SupplyDelivery. occurrenceDateTime, SupplyDelivery. occurrencePeriod.démarrer, SupplyDelivery. occurrenceTiming.événement
SupplyRequest	SupplyRequest.authoredOn
VisionPrescription	VisionPrescription.dateWritten

Exportation de données depuis votre magasin de HealthLake données à l'aide de $export

Pour effectuer une demande d'exportation en utilisant le paramètre FHIR REST API `$export` Specify dans le cadre de la `POST` demande et en incluant les paramètres de la demande dans le corps de votre demande. Selon les FHIR spécifications, le FHIR serveur doit prendre en charge les `GET` demandes et peut prendre en charge les `POST` demandes. Afin de prendre en charge des paramètres supplémentaires, un corps est nécessaire pour démarrer l'exportation et HealthLake prend donc en charge les POST demandes.

> **Important**
>
> HealthLake les banques de données créées avant le 1er juin 2023 ne prennent en charge que les demandes de travail d'exportation FHIR REST API basées sur les exportations à l'échelle du système.
> HealthLake les magasins de données créés avant le 1er juin 2023 ne permettent pas d'obtenir le statut d'une exportation à l'aide d'une `GET` demande sur le point de terminaison d'un magasin de données.

Toutes les demandes d'exportation que vous effectuez à l'aide du FHIR REST API sont renvoyées au `ndjson` format et exportées vers un compartiment Amazon S3. Chaque objet S3 ne contiendra qu'un seul type de FHIR ressource.

Vous pouvez effectuer une seule demande d'exportation pour chaque AWS compte à la fois. Pour en savoir plus sur les Quotas de Service associés à HealthLake, consultezAWS HealthLake points de terminaison et quotas.

Pour en savoir plus sur la création d'une demande d'exportation à l'aide du FHIR RESTAPI, voirExportation de données depuis votre HealthLake banque de données avec FHIR REST API opérations.

Interrogez les magasins de AWS HealthLake données SQL à l'aide d'Amazon Athena

Lorsque vous créez un magasin de HealthLake données, la structure de FHIR données hautement imbriquée est ingérée dans Amazon Athena et automatiquement transformée en tables Iceberg interrogeables avec. SQL L'octroi de l'accès à cette nouvelle ressource est géré à l'aide de AWS Lake Formation. Chaque type de FHIR ressource est représenté sous forme de table individuelle dans Athena.

Important

Pour les magasins de données créés avant le 14 novembre 2022, vous devez migrer votre magasin de données existant vers un nouveau pour l'interroger à l'aide de celui-ciSQL. Pour obtenir de l'aide, veuillez consulter Migration d'un magasin de données existant pour utiliser Amazon Athena.

Note

Après le 20 février 2023, les magasins de HealthLake données n'utilisent pas le traitement du langage naturel intégré (NLP) par défaut. Si vous souhaitez activer cette fonctionnalité sur votre banque de données, consultez le Comment activer la fonction intégrée HealthLake de traitement du langage naturel ? chapitre Dépannage.

Pour créer un magasin de HealthLake données, vous devez ajouter des IAM politiques supplémentaires et un rôle de service à votre IAM utilisateur ou un rôle d' HealthLake administrateur. Pour plus d'informations sur la configuration des autorisations, consultezConfiguration des autorisations pour commencer à utiliser AWS HealthLake.

HealthLake les magasins de données sont ingérés dans Athena sous forme de tables Iceberg. Pour en savoir plus sur le fonctionnement des tables Iceberg dans Athena, consultez la section Utilisation des tables Iceberg dans le guide de l'utilisateur d'Athena.

HealthLake prend en charge le `READ` fonctionnement de vos magasins de HealthLake données (magasins de données) dans Athena. Pour en savoir plus sur les opérations de création, de lecture,

de mise à jour et de suppression (CRUD) utilisant ces FHIR REST API opérations, consultez la section Utilisation FHIR REST API des interactions avec un magasin HealthLake de données pour en savoir plus sur la manière dont les CRUD opérations affectent vos données dans Athena.

Les rubriques de ce chapitre décrivent comment connecter votre banque de HealthLake données à Athena, comment l'interroger à l'aide et comment connecter SQL les résultats à d'autres AWS services pour une analyse plus approfondie.

Table des matières

Connecter votre banque de données à Amazon Athena

> **Important**
>
> Après le 14 novembre 2022, les IAM conditions d'accès HealthLake ont changé. Pour créer des banques de données et pour autoriser l'accès à celles-ci dans Athena, vous devez ajouter la politique `AWSLakeFormationDataAdmin` gérée à votre IAM utilisateur, groupe ou rôle. Vous pouvez utiliser cette `AWSLakeFormationDataAdmin` politique pour créer des administrateurs de lacs de données et autoriser l'accès aux magasins de données d'Athena.

Cette rubrique décrit les étapes nécessaires pour créer un utilisateur, un groupe ou un rôle Athena et leur accorder l'accès aux FHIR ressources d'un magasin de HealthLake données.

- Accorder à un utilisateur, un groupe ou un rôle l'accès à un magasin de HealthLake données (AWS Lake Formation Console)
- Création d'un compte Athena

Accorder à un utilisateur, un groupe ou un rôle l'accès à un magasin de HealthLake données (AWS Lake Formation Console)

> Persona : administrator HealthLake
>
> Le personnage d' HealthLake administrateur est un administrateur de lac de données dans AWS Lake Formation. Ils donnent accès aux magasins de HealthLake données de Lake Formation.

Pour chaque magasin de données créé, deux entrées sont visibles dans la console AWS Lake Formation. L'une des entrées est un lien vers une ressource. Les noms des liens vers les ressources sont toujours affichés en italique. Chaque lien de ressource est affiché avec le nom et le propriétaire de la ressource partagée associée. Pour tous les magasins de HealthLake données, le propriétaire de la ressource partagée est le compte HealthLake de service. L'autre entrée est le magasin HealthLake de données dans le compte HealthLake de service. Les étapes de cette procédure utilisent le magasin de données qui est le lien de ressource.

Pour en savoir plus sur les liens vers des ressources, voir Comment fonctionnent les liens vers des ressources dans Lake Formation in the AWS Lake Formation Developer Guide.

Pour qu'un utilisateur, un groupe ou un rôle puisse interroger des données dans Athena, vous devez accorder l'autorisation Describe sur la base de données de ressources. Ensuite, vous devez autoriser Select et Describe sur les tables.

STEP1 : Pour accorder DESCRIBEdes autorisations sur une base de HealthLake données contenant des liens vers des ressources

1. Ouvrez la console AWS Lake Formation : https://console.aws.amazon.com/lakeformation/
2. Dans la barre de navigation principale, sélectionnez Bases de données.
3. Sur la page Bases de données, cliquez sur le bouton radio à côté du nom du magasin de données en italique.
4. Choisissez Actions (▾).
5. Choisissez Grant (Accorder).
6. Sur la page Autoriser les données d'octroi, sous Principaux, choisissez des IAMutilisateurs ou des rôles.

7. Sous IAMutilisateurs ou rôles, utilisez la flèche vers le bas (▼) ou recherchez l'IAMutilisateur, le rôle ou le groupe sur lequel vous souhaitez pouvoir effectuer des requêtes dans Athena.
8. Sous Balises LF ou carte de ressources de catalogue, choisissez l'option Ressources de catalogue de données nommées.
9. Sous Bases de données, utilisez la flèche vers le bas (▼) pour choisir la HealthLake base de données à laquelle vous souhaitez partager l'accès.
10. Dans la fiche d'autorisation des liens vers les ressources, sous Autorisations des liens vers les ressources, sélectionnez Décrire.

Lorsque l'autorisation est accordée, la bannière de réussite de l'autorisation apparaît. Pour consulter l'autorisation que vous venez d'accorder, choisissez Data lake permissions. Recherchez l'utilisateur, le groupe et le rôle dans le tableau. Sous la colonne Autorisations, vous verrez la liste Décrire.

Vous devez maintenant utiliser Grant on target pour autoriser Select et Describe sur toutes les tables de la base de données.

STEP2 : Autoriser l'accès à toutes les tables d'un lien de ressource d'une banque de HealthLake données

1. Ouvrez la console AWS Lake Formation : https://console.aws.amazon.com/lakeformation/
2. Dans la barre de navigation principale, sélectionnez Bases de données.
3. Sur la page Bases de données, cliquez sur le bouton radio à côté du nom du magasin de données en italique.
4. Choisissez Actions (▼).
5. Choisissez Grant on target.
6. Sur la page Autoriser les données d'octroi, sous Principaux, choisissez des IAMutilisateurs ou des rôles.
7. Sous IAMutilisateurs ou rôles, utilisez la flèche vers le bas (▼) ou recherchez l'IAMutilisateur, le groupe ou le rôle sur lequel vous souhaitez pouvoir effectuer des requêtes dans Athena.
8. Sous Balises LF ou carte de ressources de catalogue, choisissez l'option Ressources de catalogue de données nommées.
9. Sous Bases de données, utilisez la flèche vers le bas (▼) pour choisir la HealthLake base de données à laquelle vous souhaitez accorder l'accès.
10. Sous Tables, choisissez Toutes les tables pour partager toutes les tables avec un HealthLake utilisateur.

11. Dans la fiche Autorisations du tableau, sous Autorisations du tableau, choisissez Décrire et sélectionner.
12. Choisissez Grant (Accorder).

Après avoir choisi l'octroi, une bannière de réussite de l'octroi des autorisations apparaît. L'utilisateur spécifié peut désormais effectuer des requêtes sur un magasin de HealthLake données dans Athena.

Commencer à utiliser Athena

> **HealthLake utilisateur**
>
> L' HealthLake utilisateur utilisera la console Athena ou AWS SDKs pour interroger un magasin de HealthLake données partagé avec lui par l' HealthLake administrateur. AWS CLI

Pour interroger un magasin de données à l'aide d'Athena, vous devez effectuer les trois opérations suivantes.

- Accordez à l'IAMutilisateur ou au rôle l'accès au magasin de HealthLake données via Lake Formation. Pour en savoir plus, consultez Accorder à un utilisateur, un groupe ou un rôle l'accès à un magasin de HealthLake données (AWS Lake Formation Console).
- Créez un groupe de travail pour votre banque HealthLake de données.
- Désignez un compartiment Amazon S3 pour stocker les résultats de vos requêtes.

Pour commencer à utiliser Athena, ajoutez les politiques FullAccess AWS gérées AmazonAthenaFullAccesset AmazonS3 à votre utilisateur, groupe ou rôle. L'utilisation d'une politique AWS gérée est un excellent moyen de commencer à utiliser un nouveau service. Gardez à l'esprit que les AWS politiques gérées peuvent ne pas accorder les autorisations de moindre privilège pour vos cas d'utilisation spécifiques, car elles peuvent être utilisées par tous les clients AWS. Lorsque vous définissez des autorisations à IAM l'aide de politiques, accordez uniquement les autorisations nécessaires à l'exécution d'une tâche. Pour en savoir plus sur le moindre privilège IAM et son application, voir Appliquer les autorisations du moindre privilège dans le Guide de l'utilisateur. IAM

> **Important**
>
> Pour interroger un magasin de HealthLake données dans Athena, vous devez utiliser le moteur Athena version 3.

Les groupes de travail sont des ressources. Vous pouvez donc utiliser des politiques IAM basées pour contrôler l'accès à des groupes de travail spécifiques. Pour en savoir plus, consultez la section Utilisation de groupes de travail pour contrôler l'accès aux requêtes et les coûts dans le Guide de l'utilisateur d'Athena.

Pour en savoir plus sur la configuration des groupes de travail, consultez le guide https://docs.aws.amazon.com/athena/latest/ug/workgroups-procedure.html de l'utilisateur d'Athena.

> **Note**
>
> La région dans laquelle se trouve votre compartiment Amazon S3 et la console Athena doivent correspondre.

Pour pouvoir exécuter une requête, vous devez spécifier un emplacement de compartiment de résultats de requête dans Simple Storage Service (Amazon S3) ou utiliser un groupe de travail qui a spécifié un compartiment et dont la configuration remplace les paramètres du client. Les fichiers de sortie sont enregistrés automatiquement pour chaque requête qui s'exécute.

Pour plus de détails sur la spécification de l'emplacement des résultats de requête dans la console Athena, consultez la section Spécification d'un emplacement de résultat de requête à l'aide de la console Athena dans le guide de l'utilisateur d'Amazon Athena.

Pour voir des exemples illustrant la manière d'interroger votre HealthLake banque de données dans Athena, reportez-vous à. Interrogez votre banque HealthLake de données à l'aide de SQL

Interrogez votre banque HealthLake de données à l'aide de SQL

> **Note**
>
> Après le 20 février 2023, les magasins de HealthLake données n'utilisent pas le traitement automatique du langage naturel intégré (NLP) par défaut. Si vous souhaitez activer cette

fonctionnalité sur votre banque de données, consultez le Comment activer la fonction intégrée HealthLake de traitement du langage naturel ? chapitre Dépannage.
Tous les exemples présentés dans cette rubrique utilisent des données fictives créées à l'aide de Synthea. Pour en savoir plus sur la création d'un magasin de données préchargé avec des données Synthea, consultez. Création d'un magasin de données dans AWS HealthLake

Lorsque vous importez votre HealthLake banque de données dans Athena, chaque type de ressource de votre banque de HealthLake données est converti en table. Ces tables peuvent être interrogées individuellement ou en groupe à l'aide de requêtes SQL basées. En raison de la structure des magasins de données, vos données sont importées dans Athena sous la forme de plusieurs types de données différents. Pour en savoir plus sur la création de SQL requêtes pouvant accéder à ces types de données, consultez la section Interrogation de tableaux dotés de types complexes et de structures imbriquées dans le guide de l'utilisateur d'Amazon Athena.

Pour chaque élément d'un type de ressource, la FHIR spécification définit une cardinalité. La cardinalité d'un élément définit les limites inférieure et supérieure du nombre de fois que cet élément peut apparaître. Lorsque vous créez une SQL requête, vous devez en tenir compte. Par exemple, examinons certains éléments du champ Type de ressource : Patient.

- Elément : Nom La FHIR spécification définit la cardinalité comme`0..*`.

  L'élément est capturé sous forme de tableau.

```
[{
 id = null,
 extension = null,
 use = official,
 _use = null,
 text = null,
 _text = null,
 family = Wolf938,
 _family = null,
 given = [Noel608],
 _given = null,
 prefix = null,
 _prefix = null,
 suffix = null,
 _suffix = null,
```

```
 period = null
}]
```

Dans Athena, pour voir comment un type de ressource a été ingéré, recherchez-le sous Tables et vues. Pour accéder aux éléments de ce tableau, vous pouvez utiliser la notation par points. Voici un exemple simple qui permettrait d'accéder aux valeurs de `given` et`family`.

```
SELECT
    name[1].given as FirstName,
    name[1].family as LastName
FROM Patient
```

- Élément : MaritalStatus La FHIR spécification définit la cardinalité comme`0..1`.

  Cet élément est capturé en tant queJSON.

```
{
 id = null,
 extension = null,
 coding = [
  {
   id = null,
   extension = null,
   system = http: //terminology.hl7.org/CodeSystem/v3-MaritalStatus,
    _system = null,
   version = null,
   _version = null,
   code = S,
   _code = null,
   display = Never Married,
   _display = null,
   userSelected = null,
   _userSelected = null
  }

 ],
 text = Never Married,
 _text = null
}
```

Dans Athena, pour voir comment un type de ressource a été ingéré, recherchez-le sous Tables et vues. Pour accéder aux paires clé-valeur dans leJSON, vous pouvez utiliser la notation par points.

Comme il ne s'agit pas d'un tableau, aucun index de tableau n'est requis. Voici un exemple simple qui permettrait d'accéder à la valeur `detext`.

```
SELECT
    maritalstatus.text as MaritalStatus
FROM Patient
```

Pour en savoir plus sur l'accès et la rechercheJSON, consultez la section Interrogation JSON dans le guide de l'utilisateur d'Athena.

Les instructions de requête Athena Data Manipulation Language (DML) sont basées sur Trino. Athena ne prend pas en charge toutes les fonctionnalités de Trino, et il existe des différences importantes. Pour en savoir plus, consultez les DMLrequêtes, les fonctions et les opérateurs dans le guide de l'utilisateur d'Amazon Athena.

En outre, Athena prend en charge plusieurs types de données que vous pouvez rencontrer lors de la création de requêtes dans votre banque de HealthLake données. Pour en savoir plus sur les types de données dans Athena, consultez la section Types de données dans Amazon Athena dans le guide de l'utilisateur d'Amazon Athena.

Pour en savoir plus sur le fonctionnement SQL des requêtes dans Athena, consultez la SQLréférence relative à Amazon Athena dans le guide de l'utilisateur d'Amazon Athena.

Chaque onglet présente des exemples de recherche sur les types de ressources spécifiés et les éléments associés à l'aide d'Athena.

Element: Extension

L'élément `extension` est utilisé pour créer des champs personnalisés dans un magasin de données.

Cet exemple montre comment accéder aux fonctionnalités de l'`extension`élément présent dans le type de `Patient` ressource.

Lorsque votre banque de HealthLake données est importée dans Athena, les éléments d'un type de ressource sont analysés différemment. La structure de la variable `element` is ne peut pas être entièrement spécifiée dans le schéma. Pour gérer cette variabilité, les éléments du tableau sont transmis sous forme de chaînes.

Dans la description du tableau de`Patient`, vous pouvez voir l'élément `extension` décrit comme`array<string>`, ce qui signifie que vous pouvez accéder aux éléments du tableau en utilisant une valeur d'index. Pour accéder aux éléments de la chaîne, vous devez toutefois utiliser`json_extract`.

Voici une seule entrée de l'`extension`élément trouvé dans le tableau des patients.

```
[{
  "valueString": "Kerry175 Cummerata161",
  "url": "http://hl7.org/fhir/StructureDefinition/patient-mothersMaidenName"
 },
 {
  "valueAddress": {
   "country": "DE",
   "city": "Hamburg",
   "state": "Hamburg"
  },
  "url": "http://hl7.org/fhir/StructureDefinition/patient-birthPlace"
 },
 {
  "valueDecimal": 0.0,
  "url": "http://synthetichealth.github.io/synthea/disability-adjusted-life-years"
 },
 {
  "valueDecimal": 5.0,
  "url": "http://synthetichealth.github.io/synthea/quality-adjusted-life-years"
 }
]
```

Même si cela est valideJSON, Athéna le traite comme une chaîne.

Cet exemple de SQL requête montre comment créer une table contenant les `patient-birthPlace` éléments `patient-mothersMaidenName` et. Pour accéder à ces éléments, vous devez utiliser différents indices de tableau et `json_extract.`

```
SELECT
    extension[1],
    json_extract(extension[1], '$.valueString') AS MothersMaidenName,
    extension[2],
    json_extract(extension[2], '$.valueAddress.city') AS birthPlace
FROM patient
```

Pour en savoir plus sur les requêtes impliquant des requêtesJSON, consultez la section Extraction de données JSON dans le guide de l'utilisateur d'Amazon Athena.

Element: birthDate (Age)

L'âge n'est pas un élément du type de ressource Patient dansFHIR. Voici deux exemples de recherches filtrées en fonction de l'âge.

Comme l'âge n'est pas un élément, nous utilisons le `birthDate` pour les SQL requêtes. Pour savoir comment un élément a été ingéréFHIR, recherchez le nom de la table sous Tables et vues. Vous pouvez voir qu'il est de type chaîne.

Exemple 1 : Calcul d'une valeur pour l'âge

Dans cet exemple de SQL requête, nous utilisons un SQL outil intégré `year` pour extraire ces composants. `current_date` Ensuite, nous les soustrayons pour renvoyer l'âge réel du patient sous la forme d'une colonne appeléeage.

```
SELECT
 (year(current_date) - year(date(birthdate))) as age
FROM patient
```

Exemple 2 : Filtrage pour les patients nés avant `2019-01-01` et ceux qui le sontmale.

La SQL requête indique comment utiliser la `CAST` fonction pour convertir l'`birthDate`élément en type `DATE` et comment filtrer en fonction des deux critères de la `WHERE` clause. Étant donné que l'élément est ingéré en tant que chaîne de caractères par défaut, nous devons l'`CAST`ingérer en tant que type`DATE`. Ensuite, vous pouvez utiliser l'<opérateur pour le comparer à une autre date,`2019-01-01`. En utilisant`AND`, vous pouvez ajouter un deuxième critère à la `WHERE` clause.

```
SELECT birthdate
FROM patient
-- we convert birthdate (varchar) to date  > cast that as date too
WHERE CAST(birthdate AS DATE) < CAST('2019-01-01' AS DATE) AND gender = 'male'
```

Resource type: Location

Cet exemple montre les recherches de lieux dans le type de ressource Location dont le nom de ville est Attleboro.

```
SELECT *
```

```
FROM Location
WHERE address.city='ATTLEBORO'
LIMIT 10;
```

Element: Age

```
SELECT birthdate
FROM patient
-- we convert birthdate (varchar) to date  > cast that as date too
WHERE CAST(birthdate AS DATE) < CAST('2019-01-01' AS DATE) AND gender = 'male'
```

Resource type: Condition

La condition du type de ressource stocke les données de diagnostic relatives à des problèmes devenus préoccupants. HealthLakele traitement médical intégré du langage naturel (NLP) génère de nouvelles `Condition` ressources en fonction des détails trouvés dans le type de DocumentReference ressource. Lorsque de nouvelles ressources sont générées, HealthLake ajoute la balise `SYSTEM_GENERATED` à l'`meta`élément. Cet exemple de SQL requête montre comment effectuer une recherche dans la table des conditions et renvoyer des résultats lorsque les `SYSTEM_GENERATED` résultats ont été supprimés.

Pour en savoir plus sur HealthLake le traitement intégré du langage naturel (NLP), voirUtilisation de la génération automatique de ressources basée sur le traitement du langage naturel (NLP) du type de FHIR DocumentReference ressource dans AWS HealthLake.

```
SELECT *
FROM condition
WHERE meta.tag[1] is NULL
```

Vous pouvez également effectuer une recherche dans un élément de chaîne spécifique pour filtrer davantage votre requête. L'`modifierextension`élément contient des détails sur la `DocumentReference` ressource qui a été utilisée pour générer un ensemble de conditions. Encore une fois, vous devez utiliser `json_extract` pour accéder aux JSON éléments imbriqués qui sont introduits dans Athena sous forme de chaîne.

Cet exemple de SQL requête montre comment vous pouvez rechercher tout `Condition` ce qui a été généré en fonction d'un élément spécifique`DocumentReference`. Permet `CAST` de définir l'JSONélément sous forme de chaîne afin de pouvoir l'utiliser `LIKE` pour le comparer.

```
SELECT
```

```
    meta.tag[1].display as SystemGenerated,
    json_extract(modifierextension[4], '$.valueReference.reference') as
 DocumentReference
FROM condition
WHERE meta.tag[1].display = 'SYSTEM_GENERATED'

AND CAST(json_extract(modifierextension[4], '$.valueReference.reference') as
 VARCHAR) LIKE '%DocumentReference/67aa0278-8111-40d0-8adc-43055eb9d18d%'
```

Resource type: Observation

Le type de ressource Observation stocke les mesures et les assertions simples faites à propos d'un patient, d'un appareil ou d'un autre sujet. HealthLakele traitement du langage naturel intégré (NLP) génère de nouvelles `Observation` ressources en fonction des détails trouvés dans une `DocumentReference` ressource. Cet exemple de SQL requête inclut des `WHERE meta.tag[1] is NULL` commentaires, ce qui signifie que les `SYSTEM_GENERATED` résultats sont inclus.

```
SELECT valueCodeableConcept.coding[1].code
FROM Observation
WHERE  valueCodeableConcept.coding[1].code = '266919005'
-- WHERE meta.tag[1] is NULL
```

Cette colonne a été importée en tant que `struct`. Par conséquent, vous pouvez accéder aux éléments qu'il contient en utilisant la notation par points.

Resource type: MedicationStatement

MedicationStatement est un type de FHIR ressource que vous pouvez utiliser pour stocker des informations sur les médicaments qu'un patient a pris, prend ou prendra à l'avenir. HealthLakele traitement médical intégré du langage naturel (NLP) génère de nouvelles MedicationStatement ressources sur la base des documents trouvés dans le type de DocumentReference ressource. Lorsque de nouvelles ressources sont générées, HealthLake ajoute la balise `SYSTEM_GENERATED` à l'`meta`élément. Cet exemple de SQL requête montre comment créer une requête filtrant en fonction d'un seul patient en utilisant son identifiant et recherchant les ressources qui ont été ajoutées par HealthLake l'intégrationNLP.

```
SELECT *
FROM medicationstatement
WHERE meta.tag[1].display = 'SYSTEM_GENERATED' AND subject.reference =
 'Patient/0679b7b7-937d-488a-b48d-6315b8e7003b';
```

Pour en savoir plus sur HealthLake la médecine intégréeNLP, consultezUtilisation de la génération automatique de ressources basée sur le traitement du langage naturel (NLP) du type de FHIR DocumentReference ressource dans AWS HealthLake.

Exemples de SQL requêtes avec filtrage complexe

> **Note**
>
> Après le 20 février 2023, les magasins de HealthLake données n'utilisent pas le traitement automatique du langage naturel intégré (NLP) par défaut. Si vous souhaitez activer cette fonctionnalité sur votre banque de données, consultez le Comment activer la fonction intégrée HealthLake de traitement du langage naturel ? chapitre Dépannage.

Les exemples présentés dans cette rubrique incluent des SQL requêtes pour HealthLake l'intégration avec Athena qui utilisent un filtrage complexe.

Example Création de critères de filtrage basés sur des données démographiques

Il est important d'identifier les données démographiques correctes lors de la création d'une cohorte de patients. Cet exemple de requête montre comment utiliser la notation par points Trino et `json_extract` comment filtrer les données dans votre banque de HealthLake données.

```
SELECT
    id
    , CONCAT(name[1].family, ' ', name[1].given[1]) as name
    , (year(current_date) - year(date(birthdate))) as age
    , gender as gender
    , json_extract(extension[1], '$.valueString') as MothersMaidenName
    , json_extract(extension[2], '$.valueAddress.city') as birthPlace
    , maritalstatus.coding[1].display as maritalstatus
    , address[1].line[1] as addressline
    , address[1].city as city
    , address[1].district as district
    , address[1].state as state
    , address[1].postalcode as postalcode
    , address[1].country as country
    , json_extract(address[1].extension[1], '$.extension[0].valueDecimal') as latitude
    , json_extract(address[1].extension[1], '$.extension[1].valueDecimal') as longitude
    , telecom[1].value as telNumber
```

```
    , deceasedboolean as deceasedIndicator
    , deceaseddatetime
FROM database.patient;
```

Avec la console Athena, vous pouvez mieux trier et télécharger les résultats.

Example Création de filtres pour un patient et ses pathologies associées

Cet exemple de requête montre comment vous pouvez rechercher et trier toutes les affections associées aux patients trouvés dans une banque de HealthLake données.

```
SELECT
 patient.id as patientId
    , condition.id  as conditionId
    , CONCAT(name[1].family, ' ', name[1].given[1]) as name
    , condition.meta.tag[1].display
    , json_extract(condition.modifierextension[1], '$.valueDecimal') AS confidenceScore
    , category[1].coding[1].code as categoryCode
    , category[1].coding[1].display as categoryDescription
    , code.coding[1].code as diagnosisCode
    , code.coding[1].display as diagnosisDescription
    , onsetdatetime
    , severity.coding[1].code as severityCode
    , severity.coding[1].display as severityDescription
    , verificationstatus.coding[1].display as verificationStatus
    , clinicalstatus.coding[1].display as clinicalStatus
    , encounter.reference as encounterId
    , encounter.type as encountertype
FROM database.patient, condition
WHERE CONCAT('Patient/', patient.id) = condition.subject.reference
ORDER BY name;
```

Vous pouvez utiliser la console Athena pour trier davantage ces résultats ou les télécharger pour une analyse plus approfondie.

Example Création de filtres pour un patient et ses observations associées

Cet exemple de requête montre comment vous pouvez rechercher et trier toutes les observations associées relatives aux patients trouvées dans un magasin de HealthLake données.

```
SELECT
 patient.id as patientId
```

```
    , observation.id as observationId
    , CONCAT(name[1].family, ' ', name[1].given[1]) as name
    , meta.tag[1].display
    , json_extract(modifierextension[1], '$.valueDecimal') AS confidenceScore
    , status
    , category[1].coding[1].code as categoryCode
    , category[1].coding[1].display as categoryDescription
    , code.coding[1].code as observationCode
    , code.coding[1].display as observationDescription
    , effectivedatetime
    , CASE
  WHEN valuequantity.value IS NOT NULL THEN CONCAT(CAST(valuequantity.value AS
 VARCHAR),' ',valuequantity.unit)
       WHEN valueCodeableConcept.coding [ 1 ].code IS NOT NULL THEN
 CAST(valueCodeableConcept.coding [ 1 ].code AS VARCHAR)
       WHEN valuestring IS NOT NULL THEN CAST(valuestring AS VARCHAR)
       WHEN valueboolean IS NOT NULL THEN CAST(valueboolean AS VARCHAR)
       WHEN valueinteger IS NOT NULL THEN CAST(valueinteger AS VARCHAR)
       WHEN valueratio IS NOT NULL THEN CONCAT(CAST(valueratio.numerator.value AS
 VARCHAR),'/',CAST(valueratio.denominator.value AS VARCHAR))
       WHEN valuerange IS NOT NULL THEN CONCAT(CAST(valuerange.low.value AS
 VARCHAR),'-',CAST(valuerange.high.value AS VARCHAR))
       WHEN valueSampledData IS NOT NULL THEN CAST(valueSampledData.data AS VARCHAR)
       WHEN valueTime IS NOT NULL THEN CAST(valueTime AS VARCHAR)
       WHEN valueDateTime IS NOT NULL THEN CAST(valueDateTime AS VARCHAR)
       WHEN valuePeriod IS NOT NULL THEN valuePeriod.start
       WHEN component[1] IS NOT NULL THEN CONCAT(CAST(component[2].valuequantity.value
 AS VARCHAR),' ',CAST(component[2].valuequantity.unit AS VARCHAR),
 '/', CAST(component[1].valuequantity.value AS VARCHAR),'
 ',CAST(component[1].valuequantity.unit AS VARCHAR))
    END AS observationvalue
 , encounter.reference as encounterId
    , encounter.type as encountertype
FROM database.patient, observation
WHERE CONCAT('Patient/', patient.id) = observation.subject.reference
ORDER BY name;
```

Example Création de conditions de filtrage pour un patient et les procédures associées

Connecter les procédures aux patients est un aspect important des soins de santé. Cette SQL requête montre comment vous pouvez utiliser les types de ressources patient et de procédure pour ce faire dans Athena. Cette SQL requête renverra tous les patients et les procédures associées trouvés dans votre banque de HealthLake données.

```
SELECT
 patient.id  as patientId
 , PROCEDURE.id as procedureId
 , CONCAT(name[1].family, ' ', name[1].given[1]) as name
 , status
 , category.coding[1].code as categoryCode
 , category.coding[1].display as categoryDescription
 , code.coding[1].code as procedureCode
 , code.coding[1].display as procedureDescription
 , performeddatetime
 , performer[1]
 , encounter.reference as encounterId
 , encounter.type as encountertype
FROM database.patient, procedure
WHERE CONCAT('Patient/', patient.id) = procedure.subject.reference
ORDER BY name;
```

Vous pouvez désormais utiliser la console Athena pour télécharger les résultats pour une analyse plus approfondie ou les trier afin de mieux comprendre les résultats.

Example Création de conditions de filtrage pour un patient et ses prescriptions connexes

Il est important de consulter la liste à jour des médicaments que les patients prennent. À l'aide d'Athena, vous pouvez écrire une SQL requête qui utilise à la fois les types de patients et de MedicationRequest ressources trouvés dans votre banque de HealthLake données.

Cette SQL requête joint le patient et les MedicationRequest tables importées dans Athena. Il organise également les prescriptions en entrées individuelles à l'aide de la notation par points.

```
SELECT
 patient.id  as patientId
 , medicationrequest.id  as medicationrequestid
 , CONCAT(name[1].family, ' ', name[1].given[1]) as name
 , status
 , statusreason.coding[1].code as categoryCode
 , statusreason.coding[1].display as categoryDescription
 , category[1].coding[1].code as categoryCode
 , category[1].coding[1].display as categoryDescription
 , priority
 , donotperform
 , encounter.reference as encounterId
 , encounter.type as encountertype
 , medicationcodeableconcept.coding[1].code as medicationCode
```

```
 , medicationcodeableconcept.coding[1].display as medicationDescription
 , dosageinstruction[1].text as dosage
FROM database.patient, medicationrequest
WHERE CONCAT('Patient/', patient.id ) = medicationrequest.subject.reference
ORDER BY name
```

Vous pouvez utiliser la console Athena pour trier les résultats ou les télécharger pour une analyse plus approfondie.

Example Afficher les médicaments trouvés dans le type de MedicationStatement ressource

L'exemple de requête vous montre comment organiser l'imbriqué JSON importé dans SQL Athena à l'aide de. La requête utilise l'`meta`élément pour indiquer quand un médicament a été ajouté par HealthLake le traitement du langage naturel intégré (NLP). Pour en savoir plus sur HealthLake l'intégration avec Amazon Comprehend Medical, Utilisation de la génération automatique de ressources basée sur le traitement du langage naturel (NLP) du type de FHIR DocumentReference ressource dans AWS HealthLake consultez. Il est également utilisé `json_extract` pour rechercher des données dans le tableau de JSON chaînes.

```
SELECT
 medicationcodeableconcept.coding[1].code as medicationCode
 , medicationcodeableconcept.coding[1].display as medicationDescription
 , meta.tag[1].display
 , json_extract(modifierextension[1], '$.valueDecimal') AS confidenceScore
FROM medicationstatement;
```

Vous pouvez utiliser la console Athena pour télécharger ces résultats ou les trier.

Example Filtre pour un type de maladie spécifique

L'exemple montre comment vous pouvez trouver un groupe de patients âgés de 18 à 75 ans chez qui on a diagnostiqué un diabète.

```
SELECT patient.id as patientId,
 condition.id as conditionId,
 CONCAT(name [ 1 ].family, ' ', name [ 1 ].given [ 1 ]) as name,
 (year(current_date) - year(date(birthdate))) AS age,
 CASE
  WHEN condition.encounter.reference IS NOT NULL THEN condition.encounter.reference
  WHEN observation.encounter.reference IS NOT NULL THEN observation.encounter.reference
 END as encounterId,
 CASE
```

```
  WHEN condition.encounter.type IS NOT NULL THEN observation.encounter.type
  WHEN observation.encounter.type IS NOT NULL THEN observation.encounter.type
 END AS encountertype,
 condition.code.coding [ 1 ].code as diagnosisCode,
 condition.code.coding [ 1 ].display as diagnosisDescription,
 observation.category [ 1 ].coding [ 1 ].code as categoryCode,
 observation.category [ 1 ].coding [ 1 ].display as categoryDescription,
 observation.code.coding [ 1 ].code as observationCode,
 observation.code.coding [ 1 ].display as observationDescription,
 effectivedatetime AS observationDateTime,
 CASE
      WHEN valuequantity.value IS NOT NULL THEN CONCAT(CAST(valuequantity.value AS
 VARCHAR),' ',valuequantity.unit)
      WHEN valueCodeableConcept.coding [ 1 ].code IS NOT NULL THEN
 CAST(valueCodeableConcept.coding [ 1 ].code AS VARCHAR)
      WHEN valuestring IS NOT NULL THEN CAST(valuestring AS VARCHAR)
      WHEN valueboolean IS NOT NULL THEN CAST(valueboolean AS VARCHAR)
      WHEN valueinteger IS NOT NULL THEN CAST(valueinteger AS VARCHAR)
      WHEN valueratio IS NOT NULL THEN CONCAT(CAST(valueratio.numerator.value AS
 VARCHAR),'/',CAST(valueratio.denominator.value AS VARCHAR))
      WHEN valuerange IS NOT NULL THEN CONCAT(CAST(valuerange.low.value AS
 VARCHAR),'-',CAST(valuerange.high.value AS VARCHAR))
      WHEN valueSampledData IS NOT NULL THEN CAST(valueSampledData.data AS VARCHAR)
      WHEN valueTime IS NOT NULL THEN CAST(valueTime AS VARCHAR)
      WHEN valueDateTime IS NOT NULL THEN CAST(valueDateTime AS VARCHAR)
      WHEN valuePeriod IS NOT NULL THEN valuePeriod.start
      WHEN component[1] IS NOT NULL THEN CONCAT(CAST(component[2].valuequantity.value
 AS VARCHAR),' ',CAST(component[2].valuequantity.unit AS VARCHAR),
 '/', CAST(component[1].valuequantity.value AS VARCHAR),'
 ',CAST(component[1].valuequantity.unit AS VARCHAR))
    END AS observationvalue,
 CASE
  WHEN condition.meta.tag [ 1 ].display = 'SYSTEM GENERATED' THEN 'YES'
  WHEN condition.meta.tag [ 1 ].display IS NULL THEN 'NO'
  WHEN observation.meta.tag [ 1 ].display = 'SYSTEM GENERATED' THEN 'YES'
  WHEN observation.meta.tag [ 1 ].display IS NULL THEN 'NO'
   END AS IsSystemGenerated,
  CAST(
    json_extract(
      condition.modifierextension [ 1 ],
      '$.valueDecimal'
    ) AS int
  ) AS confidenceScore
FROM database.patient,
```

```
 database.condition,
 database.observation
WHERE CONCAT('Patient/', patient.id) = condition.subject.reference
 AND CONCAT('Patient/', patient.id) = observation.subject.reference
   AND (year(current_date) - year(date(birthdate))) >= 18
   AND (year(current_date) - year(date(birthdate))) <= 75
   AND condition.code.coding [ 1 ].display like ('%diabetes%');
```

Vous pouvez désormais utiliser la console Athena pour trier les résultats ou les télécharger pour une analyse plus approfondie.

AWS HealthLake et VPC points de terminaison d'interface ()AWS PrivateLink

Vous pouvez établir une connexion privée entre votre VPC et AWS HealthLake en créant un point de VPCterminaison d'interface. Les VPC points de terminaison de l'interface sont alimentés par AWS PrivateLinkune technologie à laquelle vous pouvez accéder HealthLake en privé, APIs sans passerelle Internet, NAT appareil, VPN connexion ou AWS Direct Connect connexion. Les instances de votre VPC site n'ont pas besoin d'adresses IP publiques pour communiquer avec HealthLake APIs ; Le trafic entre votre VPC et HealthLake ; ne quitte pas le réseau Amazon.

Chaque point de terminaison d'interface est représenté par une ou plusieurs interfaces réseau Elastic dans vos sous-réseaux.

Pour plus d'informations, consultez Interface VPC endpoints (AWS PrivateLink) dans le guide de l'VPCutilisateur Amazon.

Considérations relatives aux HealthLake VPC terminaux

Avant de configurer un point de VPC terminaison d'interface pour HealthLake, assurez-vous de consulter les propriétés et les limites du point de terminaison d'interface dans le guide de VPC l'utilisateur Amazon.

HealthLake prend en charge l'envoi d'appels à toutes ses API actions depuis votreVPC.

Création d'un point de VPC terminaison d'interface pour HealthLake ;

Vous pouvez créer un VPC point de terminaison pour le service HealthLake ; à l'aide de la VPC console Amazon ou du AWS Command Line Interface (AWS CLI). Pour plus d'informations, consultez la section Création d'un point de terminaison d'interface dans le guide de VPC l'utilisateur Amazon.

Créez un VPC point de terminaison pour HealthLake ; en utilisant le nom de service suivant :

- com.amazonaws. *region*.healthlake

Si vous activez le mode privé DNS pour le point de terminaison, vous pouvez faire des API demandes HealthLake en utilisant son DNS nom par défaut pour la région. Par exemple, *healthlake.us-east-1.amazonaws.com*.

Pour plus d'informations, consultez la section Accès à un service via un point de terminaison d'interface dans le guide de VPC l'utilisateur Amazon.

Création d'une politique de VPC point de terminaison pour HealthLake

Vous pouvez associer une politique de point de terminaison à votre VPC point de terminaison qui contrôle l'accès à HealthLake. La politique spécifie les informations suivantes :

- Le principal qui peut exécuter des actions.
- Les actions qui peuvent être effectuées.
- Les ressources sur lesquelles les actions peuvent être exécutées.

Pour plus d'informations, consultez la section Contrôle de l'accès aux services avec des VPC points de terminaison dans le guide de VPC l'utilisateur Amazon.

Exemple : politique des VPC points de terminaison pour les HealthLake actions

Voici un exemple de politique de point de terminaison pour HealthLake. Lorsqu'elle est attachée à un point de terminaison, cette politique accorde l'accès à l' HealthLake`CreateFHIRDatastore`action à tous les principaux sur toutes les ressources.

```
{
   "Statement":[
      {
         "Principal":"*",
         "Effect":"Allow",
         "Action":[
            "healthlake:create-fhir-datastore"
         ],
         "Resource":"*"
      }
   ]
}
```

Marquage des ressources dans AWS HealthLake

Vous pouvez attribuer des métadonnées à vos ressources AWS sous la forme de balises. Chaque balise est une étiquette composée d'une clé définie par l'utilisateur et d'une valeur. Les balises peuvent vous aider à gérer, identifier, organiser, rechercher et filtrer des ressources.

Cette rubrique décrit les catégories et stratégies de balisage couramment utilisées pour vous aider à mettre en œuvre une stratégie de balisage cohérente et efficace. Les sections suivantes supposent des connaissances de base en matière de AWS ressources, de balisage, de facturation détaillée et AWS d'Identity and Access Management (IAM).

Chaque balise se compose de deux parties :

- Une clé de balise (par exemple CostCenter, Environnement ou Projet). Les clés de balises sont sensibles à la casse.
- Une valeur de balise (par exemple, 111122223333 ou Production). Les valeurs de balise sont sensibles à la casse, tout comme les clés de balise.

Vous pouvez utiliser les balises pour classer vos ressources par objectif, propriétaire, environnement ou selon d'autres critères. Pour de plus amples informations, consultez Politiques de balisage AWS.

Vous pouvez ajouter, modifier ou supprimer des balises ressource par ressource depuis la console de service, le service API ou le AWSCLI.

Pour activer le balisage, assurez-vous que vous TagResources êtes autorisé. Vous pouvez autoriser TagResources en joignant une IAM politique comme dans l'exemple suivant.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "healthlake:CreateFHIRDatastore",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "healthlake:TagResource",
            "Resource": "*"
```

```
        }
    ]
}
```

Avis important

AWS HealthLake protège les données des clients conformément aux politiques du modèle de responsabilité AWS partagée. Cela signifie que toutes les données des clients sont cryptées à la fois pendant la transition et au repos. Cependant, les noms saisis par le client pour les magasins de données ou les opérations basées sur des tâches ne sont pas tous cryptés. Ils ne doivent jamais contenir d'informations personnelles identifiables ou d'informations de santé protégées. Pour plus d'informations, consultez le chapitre sur la AWS HealthLake sécurité.

Bonnes pratiques

Lorsque vous créez une stratégie de balisage pour les ressources AWS, suivez les meilleures pratiques :

- Ne stockez pas d'informations personnellement identifiables (PII), d'informations de santé personnelles (PHI) ou d'autres informations sensibles dans des balises.
- Utilisez un format standardisé et sensible à la casse pour les balises et appliquez-le de manière cohérente à tous les types de ressources.
- Optez pour des directives de balisage qui prennent en charge plusieurs objectifs, comme la gestion du contrôle d'accès aux ressources, le suivi des coûts, l'automatisation et l'organisation.
- Utilisez des outils automatisés pour gérer les balises de ressources. AWS Resource Groups et Resource Groups Tagging API permettent le contrôle programmatique des balises, ce qui permet de gérer, de rechercher et de filtrer automatiquement les balises et les ressources.
- Le balisage est plus efficace lorsque vous utilisez un plus grand nombre de balises.
- Les balises peuvent être éditées ou modifiées en fonction des besoins des utilisateurs. Toutefois, pour mettre à jour les balises de contrôle d'accès, vous devez également mettre à jour les politiques qui font référence à ces balises afin de contrôler l'accès à vos ressources.

Balisage des exigences

Les balises possèdent les exigences suivantes :

- Les clés ne peuvent pas être préfixées par aws :.
- Les clés doivent être uniques par ensemble de balises.
- Une clé doit comporter entre 1 et 128 caractères autorisés.
- Une valeur doit comprendre entre 0 et 256 caractères autorisés.
- Les valeurs ne doivent pas être uniques par ensemble de balises.
- Les caractères autorisés pour les clés et les valeurs sont les lettres Unicode, les chiffres, les espaces et les symboles suivants : _ . : / = + - @.
- Les clés et les valeurs sont sensibles à la casse.

Ajouter une étiquette à un magasin de données

L'ajout de balises à un magasin de données peut vous aider à identifier et à organiser vos AWS ressources et à gérer l'accès à celles-ci. Tout d'abord, vous ajoutez une ou plusieurs balises (paires clé-valeur) à un magasin de données. Vous pouvez utiliser jusqu'à cinquante balises par utilisateur. Il existe également des restrictions quant aux caractères que vous pouvez utiliser dans les champs de clé et de valeur.

Une fois que vous avez des balises, vous pouvez créer des IAM politiques pour gérer l'accès au magasin de données en fonction de ces balises. Vous pouvez utiliser la HealthLake console ou le AWS CLI pour ajouter des balises à un magasin de données. L'ajout de balises à un référentiel peut avoir un impact sur l'accès à ce référentiel. Avant d'ajouter une balise à un magasin de données, assurez-vous de consulter IAM les politiques susceptibles d'utiliser des balises pour contrôler l'accès aux ressources telles que les magasins de données.

Procédez comme suit pour utiliser le AWS CLI pour ajouter une balise à un magasin de HealthLake données. Pour ajouter une balise à un magasin de données lorsque vous le créez, voirCréation d'un magasin de données dans AWS HealthLake.

Sur le terminal ou sur la ligne de commande, exécutez la commande tag-resource en spécifiant le nom de ressource Amazon (ARN) du magasin de données dans lequel vous souhaitez ajouter des balises, ainsi que la clé et la valeur de la balise que vous souhaitez ajouter. Vous pouvez ajouter plusieurs balises à un magasin de données. Il existe également des restrictions quant aux caractères que vous pouvez utiliser dans les champs de clé et de valeur, comme indiqué dans Par exemple, Balisage des exigences pour ajouter des balises à un magasin de données lors de sa création, vous devez utiliser la commande suivante dans le AWS CLI. Le nom du magasin de données est

Test_Data_Store, et les deux balises ajoutées avec des clés sont key1 et key2 avec des valeurs value1 et value2 respectivement :

```
aws healthlake create-fhir-datastore --datastore-type-version R4 --preload-data-config
 PreloadDataType="SYNTHEA" --datastore-name "Test_Data_Store" --tags '[{"Key": "key1",
 "Value": "value1"}, {"Key": "key2", "Value": "value2"}]' --region us-east-1
```

Pour ajouter des balises à un magasin de données existant, vous devez exécuter l'exemple de commande suivant :

```
aws healthlake tag-resource --resource-arn "arn:aws:healthlake:us-
east-1:691207106566:datastore/fhir/0725c83f4307f263e16fd56b6d8ebdbe" --tags '[{"Key":
 "key1", "Value": "value1"}]' --region us-east-1
```

En cas de succès, cette commande ne renvoie aucune réponse.

Répertorier les balises d'un magasin de données

Procédez comme suit pour utiliser le AWS CLI pour afficher la liste des AWS balises d'un magasin de HealthLake données. Si aucune balise n’a été ajoutée, la liste renvoyée est vide.

Sur le terminal ou sur la ligne de commande, exécutez la list-tags-for-resource commande comme indiqué dans l'exemple suivant.

```
aws healthlake-test list-tags-for-resource --resource-arn "arn:aws:healthlake:us-
east-1:674914422125:datastore/fhir/0725c83f4307f263e16fd56b6d8ebdbe" --region us-
east-1
```

```
{
    "tags": {
        "key": "value",
        "key1": "value1"
    }
}
```

Supprimer des balises d'un magasin de données

Vous pouvez supprimer une ou plusieurs balises associées à un magasin de données. La suppression d’une balise ne supprime pas la balise d’autres ressources AWS qui sont associées à cette balise.

Sur le terminal ou sur la ligne de commande, exécutez la commande untag-resource en spécifiant le nom de ressource Amazon (ARN) du magasin de données dans lequel vous souhaitez supprimer les balises et la clé de balise de la balise que vous souhaitez supprimer.

```
aws healthlake untag-resource --resource-arn "arn:aws:healthlake:us-
east-1:674914422125:datastore/fhir/b91723d65c6fdeb1d26543a49d2ed1fa" --tag-keys
 '["key1"]' --region us-east-1
```

En cas de succès, cette commande ne renvoie aucune réponse. Pour vérifier les balises associées au magasin de données, exécutez la list-tags-for-resource commande.

Surveillance HealthLake

La surveillance joue un rôle important dans le maintien de la fiabilité, de la disponibilité HealthLake et des performances de vos autres AWS solutions. AWSfournit les outils de surveillance suivants pour surveiller HealthLake, signaler tout problème et prendre des mesures automatiques le cas échéant :

- Amazon CloudWatch surveille vos AWS ressources et les applications que vous utilisez AWS en temps réel. Vous pouvez collecter et suivre les métriques, créer des tableaux de bord personnalisés et définir des alarmes qui vous avertissent ou prennent des mesures lorsqu'une métrique spécifiée atteint un seuil spécifique. Par exemple, vous pouvez CloudWatch suivre CPU l'utilisation ou d'autres indicateurs de vos EC2 instances Amazon et lancer automatiquement de nouvelles instances en cas de besoin. Pour plus d'informations, consultez le guide de CloudWatch l'utilisateur Amazon.
- AWS CloudTrailenregistre API les appels et les événements connexes effectués par ou au nom de votre AWS compte. Puis, il livre les fichiers journaux à un compartiment Amazon S3 que vous spécifiez. Vous pouvez identifier les utilisateurs et les comptes appelés AWS, l'adresse IP source de ces appels et la date à laquelle ils ont eu lieu. Pour plus d’informations, consultez le AWS CloudTrail Guide de l’utilisateur .

Rubriques

- Surveillance HealthLake avec Amazon CloudWatch

Surveillance HealthLake avec Amazon CloudWatch

Vous pouvez surveiller HealthLake l'utilisation CloudWatch, qui collecte les données brutes et les transforme en indicateurs lisibles en temps quasi réel. Ces statistiques sont conservées pendant 15 mois, afin que vous puissiez utiliser ces informations historiques et avoir une meilleure idée des performances de votre application ou service Web. Vous pouvez également définir des alarmes qui surveillent certains seuils et envoient des notifications ou prennent des mesures lorsque ces seuils sont atteints. Pour plus d'informations, consultez le guide de CloudWatch l'utilisateur Amazon.

Les indicateurs sont présentés pour tous HealthLake APIs, y compris les suivants.

- Gestion du magasin de données APIs —CreateFHIRDatastore, DeleteFHIRDatastore, DescribeFHIRDatastore, L istFHIRDatastores

- Importation et exportation APIs —S tartFHIRImport Job, L istFHIRImport Job, D escribeFHIRImport Job, S tartFHIRExport Job, L istFHIRExport Jobs, D escribeFHIRExport Job
- HTTPRESTGestion des clients et des ressources APIs — CreateResource DeleteResource, GetCapabilities, ReadResource, SearchAll, SearchWithGet, SearchWithPost, UpdateResource.
- Balisage APIs — ListTagsForResource, TagResource UntagResource

Les tableaux suivants répertorient les métriques et les dimensions d' HealthLake.

Les métriques suivantes sont rapportées. Chacun est présenté sous forme de comptage de fréquences pour une plage de données spécifiée par l'utilisateur.

Métriques

Métriques	Description
Nombre d'appels	Le nombre d'appels versAPIs. Cela peut être signalé soit pour le compte, soit pour un magasin de données spécifique. Unités : nombre Statistiques valides : somme, nombre Dimensions : fonctionnement, identifiant du magasin de données, type de magasin de données
Demandes réussies	Le nombre de API demandes réussies. Unités : nombre Statistiques valides : somme, moyenne Dimensions : fonctionnement, magasin de données, type de magasin de données
Erreurs d'utilisateur	Le nombre de demandes qui ont échoué en raison d'une erreur de l'utilisateur. Unités : nombre

Métriques	Description
	Statistiques valides : somme, moyenne Dimensions : fonctionnement, identifiant du magasin de données, type de magasin de données
Erreurs de serveur	Le nombre de demandes qui ont échoué en raison d'une erreur du serveur. Unités : nombre Statistiques valides : somme, moyenne Dimensions : fonctionnement, identifiant du magasin de données, type de magasin de données
Demandes restreintes	Le nombre de demandes qui ont été limitées. Cette métrique n'est pas incluse dans le nombre d'erreurs des utilisateurs ou des serveurs. Unités : nombre Statistiques valides : somme, moyenne Dimensions : fonctionnement, identifiant du magasin de données, type de magasin de données

Métriques	Description
Latence	Le temps nécessaire, en millisecondes, pour traiter la demande de l'utilisateur. Unité : millisecondes Statistiques valides : Minimum, Maximum, Average Dimensions : fonctionnement, identifiant du magasin de données, type de magasin de données

Les dimensions suivantes sont indiquées.

Dimensions

Dimensions	Description
Opération	Quelle API opération a été utilisée
DataStoreID	Le magasin de données inclus dans la API demande
DataStoreType	Type de magasin de données (actuellement, seul le FHIR R4 est pris en charge)

Vous pouvez obtenir des métriques pour HealthLake la console AWS de gestion, le AWS CLI, ou le CloudWatch API. Vous pouvez l'utiliser CloudWatch API via l'un des kits de développement AWS logiciel Amazon (SDKs) ou des CloudWatch API outils. La HealthLake console affiche des graphiques basés sur les données brutes du CloudWatch API.

Vous devez disposer des CloudWatch autorisations appropriées pour effectuer la surveillance HealthLake CloudWatch. Pour plus d'informations, consultez Authentification et contrôle d'accès pour Amazon CloudWatch dans le guide de CloudWatch l'utilisateur Amazon.

Afficher HealthLake les métriques

Pour consulter les métriques (CloudWatch console)

1. Connectez-vous à AWSGestion de la console et ouvrez CloudWatch de la console.
2. Choisissez Metrics, choisissez All Metrics, puis choisissez AWS/HealthLake.
3. Choisissez la dimension, le nom de la métrique, puis Ajouter au graphique.
4. Choisissez une valeur pour la plage de dates. Le décompte de la métrique pour la plage de dates sélectionnée est affiché dans le graphique.

Création d'une alarme à l'aide de CloudWatch

Une CloudWatch alarme surveille une seule métrique sur une période spécifiée et exécute une ou plusieurs actions : envoyer une notification à une rubrique Amazon Simple Notification Service (AmazonSNS) ou à une politique Auto Scaling. L'action ou les actions sont basées sur la valeur de la métrique par rapport à un seuil donné sur un certain nombre de périodes que vous spécifiez. CloudWatch peut également vous envoyer un SNS message Amazon lorsque l'alarme change d'état.

CloudWatch les alarmes appellent des actions uniquement lorsque l'état change et persiste pendant la période que vous spécifiez.

Pour consulter les métriques (CloudWatch console)

1. Connectez-vous à AWSGestion de la console et ouvrez CloudWatch de la console.
2. Choisissez Alarmes, puis Créer une alarme.
3. Choisissez AWS/HealthLake, puis choisissez une métrique.
4. Pour Période, choisissez un intervalle de temps à surveiller, puis cliquez sur Suivant.
5. Saisissez un Name (Nom) et une Description.
6. Pour Whenever, choisissez **>=**, puis saisissez une valeur maximale.
7. Si vous souhaitez CloudWatch envoyer un e-mail lorsque l'état d'alarme est atteint, dans la section Actions, pour Whenever this alarm, choisissez State is ALARM. Pour Envoyer une notification à, choisissez une liste de diffusion ou choisissez Nouvelle liste et créez une nouvelle liste de diffusion.
8. Affichez un aperçu de l’alarme dans la section Aperçu de l’alarme. Si elle vous convient, choisissez Créer une alarme.

Intégration SMART FHIR avec AWS HealthLake

Une application médicale substituable et des technologies réutilisables (SMART) sur un magasin de HealthLake données FHIR activé permet aux SMART applications FHIR conformes d'accéder aux données stockées dans un magasin de HealthLake données. HealthLake les données sont accessibles en authentifiant et en autorisant les demandes à l'aide d'un serveur d'autorisation tiers et en configurant des ressources supplémentaires dans. AWS

Pour utiliser cette SMART option FHIR avec votre banque de HealthLake données, vous devez fournir les informations suivantes dans votre reateFHIRDatastore API demande C.

- Définissez l'AuthorizationStrategyégal à`SMART_ON_FHIR_V1`.
- Définissez l'IdpLambdaArn équivalent ARN de celui que AWS Lambda vous avez créé pour gérer le décodage des jetons avec votre serveur d'autorisation.
- Définissez les éléments de métadonnées spécifiés dans votre serveur d'autorisation. Ces éléments de métadonnées sont renvoyés dans le document de découverte. Pour en savoir plus, consultez Récupération du document SMART de découverte d'un magasin de HealthLake données non FHIR activé.
- Facultatif : Activez cette option FineGrainedAuthorizationEnabledsi vous avez configuré une autorisation précise sur votre serveur d'autorisation.

Vous pouvez créer un magasin de données SMART FHIR activé en utilisant le AWS Command Line Interface (AWS CLI) ou via l'un des supports AWS pris en chargeSDKs. La création d'un magasin de HealthLake données SMART FHIR activé n'est pas prise en charge à l'aide de la HealthLake console. Pour en savoir plus, consultez Création d'un SMART magasin de données non FHIR activé.

Pour définir ces paramètres dans la demande, vous devez configurer des ressources dans d'autres AWS services (AWS Secrets Manager et AWS Lambda), créer de nouveaux rôles de IAM service et configurer un serveur SMART d'autorisation non FHIR conforme. Utilisez la section Configuration des ressources requises pour implémenter un SMART magasin de données FHIR non conforme pour en savoir plus sur la configuration des ressources requises et pour obtenir une vue d'ensemble de la manière dont une SMART FHIR application active interagit avec HealthLake.

Cela signifie qu'au lieu de gérer les informations d'identification des utilisateurs par AWS Identity and Access Management votre intermédiaire, vous utilisez un SMART serveur d'autorisation non FHIR conforme.

HealthLake prend en charge SMART la FHIR version 1.0. Pour en savoir plus sur ce framework, consultez le Guide de mise en œuvre du framework de lancement d'SMARTapplications, version 1.0.

Pour autoriser et authentifier les demandes de stockage de données à l'aide de SMART onFHIR, HealthLake prend en charge l'utilisation de :

- Intégration OpenID (AuthN) : utilisée pour authentifier que cette personne ou cette application cliente est la personne (ou ce que) elle prétend être.
- OAuthIntégration 2.0 (AuthZ) : utilisée pour autoriser les FHIR ressources de votre magasin de HealthLake données auxquelles une demande authentifiée peut également lire ou écrire des données. Ceci est défini par les étendues configurées sur votre serveur d'autorisation.

Table des matières

Exigences d'authentification SMART pour FHIR

Pour accéder aux FHIR ressources d'un SMART magasin de FHIR HealthLake données intégré, une application cliente doit être autorisée par un serveur d'autorisation OAuth compatible 2.0 et présenter un jeton OAuth Bearer dans le cadre d'une demande. FHIR REST API Pour trouver le point de terminaison du serveur d'autorisation, utilisez le document HealthLake SMART on FHIR Discovery via un identifiant de ressource uniforme bien connu. Pour en savoir plus sur ce processus, consultez Récupération du document SMART de découverte d'un magasin de HealthLake données non FHIR activé.

Lorsque vous créez un SMART magasin de FHIR HealthLake données intégré, vous devez définir le point de terminaison du serveur d'autorisation et le point de terminaison du jeton dans l'`metadata`élément de la reateFHIRDatastore requête C. Pour en savoir plus sur la définition de l'`metadata`élément, voirCréation d'un SMART magasin de HealthLake données non FHIR activé.

À l'aide des points de terminaison du serveur d'autorisation, l'application cliente authentifie un utilisateur auprès du service d'autorisation. Une fois autorisé et authentifié, un jeton JSON Web (JWT) est généré par le service d'autorisation et transmis à l'application cliente. Ce jeton contient FHIR des étendues de ressources que l'application cliente est autorisée à utiliser, ce qui limite les données auxquelles l'utilisateur peut accéder. Facultativement, si le périmètre de lancement a été fourni, la réponse contiendra ces détails. Pour en savoir plus SMART sur les FHIR scopes pris en charge par HealthLake, voirSupporté SMART sur les FHIR OAuth oscilloscopes par HealthLake.

À l'aide de l'autorisation JWT accordée par le serveur d'autorisation, une application cliente FHIR REST API appelle un magasin de HealthLake données SMART FHIR activé. Pour les valider et les décoderJWT, vous devez créer une fonction Lambda. HealthLake invoque cette fonction Lambda en

votre nom lorsqu'FHIRRESTAPIune demande est reçue. Pour voir un exemple de fonction Lambda de démarrage, reportez-vous à. Utilisation AWS Lambda pour la validation des jetons avec un SMART magasin de HealthLake données FHIR activé

Éléments du serveur d'autorisation requis pour créer un magasin de HealthLake données SMART FHIR activé

Dans la reateFHIRDatastore requête C, vous devez fournir le point de terminaison d'autorisation et le point de terminaison du jeton dans le cadre de l'`metadata`élément de l'`IdentityProviderConfiguration`objet. Le point de terminaison d'autorisation et le point de terminaison jeton sont tous deux requis. Pour voir un exemple de la façon dont cela est spécifié dans reateFHIRDatastore une requête C, voirCréation d'un SMART magasin de HealthLake données non FHIR activé.

Réclamations requises pour compléter une FHIR REST API demande SMART sur un magasin de HealthLake données non FHIR activé

Pour qu'il s'agisse d'une FHIR REST API demande valide sur un magasin de HealthLake données non FHIR activé, votre AWS Lambda fonction doit contenir les affirmations suivantes. SMART

- `nbf`: (Pas avant) Réclamation — La réclamation « nbf » (pas avant) indique le délai avant lequel elle JWT MUST NOT doit être acceptée pour traitement. Le traitement de la réclamation « nbf » nécessite que le courant indiqué dans la réclamation « nbf » soit le courant date/time MUST be after or equal to the not-before date/time indiqué dans la réclamation « nbf ». L'exemple de fonction Lambda que nous fournissons convertit la réponse `iat` du serveur en. `nbf`
- `exp`: (Date d'expiration) Demande — La demande « exp » (délai d'expiration) indique le délai d'expiration à partir duquel la demande ne JWT doit pas être acceptée pour traitement.
- `isAuthorized`: un booléen défini sur. `True` Indique que la demande a été autorisée sur le serveur d'autorisation.
- `aud`: Demande (audience) — La réclamation « aud » (audience) identifie les destinataires auxquels JWT elle est destinée. Il doit s'agir d'un point SMART de terminaison du magasin de HealthLake données FHIR activé.
- `scope`: Il doit s'agir d'au moins une étendue liée à une FHIR ressource. Cette étendue est définie sur votre serveur d'autorisation. Pour en savoir plus sur les étendues liées aux FHIR ressources acceptées par HealthLake, voirHealthLake portées spécifiques aux FHIR ressources du magasin de données.

Supporté SMART sur les FHIR OAuth oscilloscopes par HealthLake

HealthLake utilise le OAuth 2.0 comme protocole d'autorisation. L'utilisation de ce protocole sur votre serveur d'autorisation vous permet de définir les FHIR ressources de votre magasin de HealthLake données auxquelles une application cliente peut également avoir un accès en lecture et/ou en écriture.

Le FHIR framework SMART on définit un ensemble de portées qui peuvent être demandées au serveur d'autorisation. Pour consulter les définitions du champ d'application dans le FHIR framework SMART on, voir SMARTle guide FHIR des HL7 FHIR ressources sur les champs d'application.

Par exemple, une application client conçue uniquement pour permettre aux patients de consulter leurs résultats de laboratoire ou de consulter leurs coordonnées ne doit être autorisée qu'à demander (sur FHIR REST demande) `read` des scopes. Pour les définir comme scope, vous devez fournir une chaîne comme celle-ci`patient/Observation.read`. Cela permettrait à l'application cliente de demander l'accès au type de `Observation` ressource en lecture seule sur le type de `Patient` ressource.

Périmètre de lancement autonome

HealthLake prend en charge le champ `launch/patient` d'application du mode de lancement autonome.

En mode de lancement autonome, une application client demande l'accès aux données cliniques du patient car l'utilisateur et le patient ne sont pas connus de l'application cliente. Ainsi, la demande d'autorisation de l'application cliente demande explicitement que le dossier du patient soit renvoyé. Une fois l'authentification réussie, le serveur d'autorisation émet un jeton d'accès contenant le champ d'application du patient de lancement demandé. Le contexte patient nécessaire est fourni avec le jeton d'accès dans la réponse du serveur d'autorisation.

Champs d'application du mode de lancement pris en charge

Portée	Description
`launch/patient`	Paramètre d'une demande d'autorisation OAuth 2.0 demandant que les données du patient soient renvoyées dans la réponse d'autorisation.

HealthLake portées spécifiques aux FHIR ressources du magasin de données

HealthLake définit trois niveaux de portée.

- Les scopes spécifiques au patient donnent accès à des données spécifiques concernant un seul patient. Quel patient est spécifié dans le contexte de lancement.
- Les étendues au niveau de l'utilisateur accordent l'accès à des données spécifiques auxquelles un utilisateur peut accéder.
- Les étendues au niveau du système accordent un accès en lecture/écriture à toutes les FHIR ressources présentes dans le magasin de données. HealthLake

Le tableau suivant indique la syntaxe permettant de créer des étendues liées aux FHIR ressources prises en charge par HealthLake. Le format général est le suivant :

```
( 'patient' | 'user' | 'system' ) '/' ( fhir-resource | '*' ) '.' ( 'read' | 'write' |
 '*' )
```

Étendue d'autorisation prise en charge sur les magasins HealthLake de données

Syntaxe Scope	Exemple de portée	Résultat
`patient/(fhir-resource | '*').('read' | 'write' | '*')`	`patient/AllergyIntolerance.*`	Une application cliente aurait un accès en lecture/écriture aux allergies.
`user/(fhir-resource | '*').('read' | 'write' | '*')`	`user/Observation.read`	Une application cliente aurait un accès en lecture à toutes les observations enregistrées.
`system/('read' | 'write' | *)`	`system/*.*`	Une application cliente aurait un accès en lecture/écriture à toutes les données.

Utilisation AWS Lambda pour la validation des jetons avec un SMART magasin de HealthLake données FHIR activé

Lorsque vous créez un magasin de HealthLake données non SMART FHIR activé, vous devez fournir ARN la AWS Lambda fonction dans la `CreateFHIRDatastore` demande. La fonction Lambda ARN est spécifiée dans l'`IdentityProviderConfiguration`objet à l'aide du `IdpLambdaArn` paramètre.

Vous devez créer la fonction Lambda avant de créer votre propre banque de HealthLake données SMART FHIR activée. Une fois que vous avez créé le magasin de données, le Lambda ARN ne peut pas être modifié. Pour voir le Lambda ARN que vous avez spécifié lors de la création du magasin de données, utilisez l'`DescribeFHIRDatastore`APIopération.

Pour qu'une FHIR REST demande aboutisse SMART sur un magasin de HealthLake données FHIR activé, votre fonction Lambda doit effectuer les opérations suivantes :

- La fonction Lambda doit renvoyer une réponse en moins d'une seconde au point de terminaison du magasin de HealthLake données.
- Décodez le jeton d'accès fourni dans l'en-tête d'autorisation de la REST API demande envoyée par l'application cliente.
- Attribuez un rôle de IAM service disposant des autorisations suffisantes pour exécuter la FHIR REST API demande.
- Les demandes suivantes sont requises pour compléter une FHIR REST API demande. Pour en savoir plus, consultez Réclamations requises.
  - `nbf`
  - `exp`
  - `isAuthorized`
  - `aud`
  - `scope`

Lorsque vous travaillez avec Lambda, vous devez créer un rôle d'exécution et une politique basée sur les ressources en plus de votre fonction Lambda. Le rôle d'exécution d'une fonction Lambda est un IAM rôle qui accorde à la fonction l'autorisation d'accéder aux AWS services et aux ressources nécessaires au moment de l'exécution. La politique basée sur les ressources que vous fournissez doit HealthLake permettre d'invoquer votre fonction en votre nom.

Les sections de cette rubrique décrivent un exemple de demande provenant d'une application cliente et une réponse décodée, les étapes nécessaires à la création d'une fonction AWS Lambda et la manière de créer une politique basée sur les ressources qui peut être assumée. HealthLake

- Partie 1 : Création d'une fonction Lambda
- Partie 2 : Création d'un rôle HealthLake de service utilisé par la fonction AWS Lambda
- Partie 3 : Mise à jour du rôle d'exécution de la fonction Lambda
- Partie 4 : Ajouter une politique de ressources à votre fonction Lambda
- Partie 5 : Configuration de la simultanéité pour votre fonction Lambda

Création d'une fonction AWS Lambda

La fonction Lambda créée dans cette rubrique est déclenchée lors de la HealthLake réception d'une demande à un magasin de HealthLake données SMART FHIR activé. La demande de l'application cliente contient un REST API appel et un en-tête d'autorisation contenant un jeton d'accès.

```
GET https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/
Authorization: Bearer i8hweunweunweofiwweoijewiwe
```

L'exemple de fonction Lambda présenté dans cette rubrique permet de masquer AWS Secrets Manager les informations d'identification liées au serveur d'autorisation. Nous vous recommandons vivement de ne pas fournir les informations de connexion au serveur d'autorisation directement dans une fonction Lambda.

Example validation d'une FHIR REST demande contenant un jeton porteur d'autorisation

L'exemple de fonction Lambda vous montre comment valider une FHIR REST demande envoyée à un magasin de HealthLake données SMART FHIR activé. Pour savoir step-by-steps comment implémenter cette fonction Lambda, consultez. Création d'une fonction Lambda à l'aide du AWS Management Console

Si la FHIR REST API demande ne contient pas de point de terminaison, de jeton d'accès et d'RESTopération valides pour le magasin de données, la fonction Lambda échouera. Pour en savoir plus sur les éléments du serveur d'autorisation requis, consultezRéclamations requises.

```
import base64
import boto3
import logging
```

```
import json
import os
from urllib import request, parse

logger = logging.getLogger()
logger.setLevel(logging.INFO)

## Uses Secrets manager to gain access to the access key ID and secret access key for
 the authorization server
client = boto3.client('secretsmanager', region_name="region-of-datastore")
response = client.get_secret_value(SecretId='name-specified-by-customer-in-
secretsmanager')
secret = json.loads(response['SecretString'])
client_id = secret['client_id']
client_secret = secret['client_secret']


unencoded_auth = f'{client_id}:{client_secret}'
headers = {
  'Authorization': f'Basic {base64.b64encode(unencoded_auth.encode()).decode()}',
  'Content-Type': 'application/x-www-form-urlencoded'
}

auth_endpoint = os.environ['auth-server-base-url'] # Base URL of the Authorization
 server
user_role_arn = os.environ['iam-role-arn'] # The IAM role client application will use
 to complete the HTTP request on the datastore

def lambda_handler(event, context):
    if 'datastoreEndpoint' not in event or 'operationName' not in event or
 'bearerToken' not in event:
    return {}

    datastore_endpoint = event['datastoreEndpoint']
    operation_name = event['operationName']
    bearer_token = event['bearerToken']
    logger.info('Datastore Endpoint [{}], Operation Name:
 [{}]'.format(datastore_endpoint, operation_name))

    ## To validate the token
    auth_response = auth_with_provider(bearer_token)
    logger.info('Auth response: [{}]'.format(auth_response))
    auth_payload = json.loads(auth_response)
```

```
    ## Required parameters needed to be sent to the datastore endpoint for the HTTP
 request to go through
    auth_payload["isAuthorized"] = bool(auth_payload["active"])
    auth_payload["nbf"] = auth_payload["iat"]
    return {"authPayload": auth_payload, "iamRoleARN": user_role_arn}

## access the server
def auth_with_provider(token):
    data = {'token': token, 'token_type_hint': 'access_token'}
    req = request.Request(url=auth_endpoint + '/v1/introspect',
 data=parse.urlencode(data).encode(), headers=headers)
    with request.urlopen(req) as resp:
    return resp.read().decode()
```

Création d'une fonction Lambda à l'aide du AWS Management Console

Cette procédure suppose que vous avez déjà créé le rôle de service que vous HealthLake souhaitez assumer lors du traitement d'une FHIR REST API demande SMART sur un magasin de HealthLake données FHIR activé. Si vous n'avez pas créé le rôle de service, vous pouvez toujours créer la fonction Lambda. Vous devez ajouter le rôle ARN de service pour que la fonction Lambda fonctionne. Pour en savoir plus sur la création d'un rôle de service et sa spécification dans la fonction Lambda, reportez-vous à Création d'un rôle HealthLake de service à utiliser dans la fonction AWS Lambda utilisée pour décoder un JWT

Pour créer une fonction Lambda ()AWS Management Console

1. Ouvrez la page Functions (Fonctions) de la console Lambda.
2. Sélectionnez Create function (Créer une fonction).
3. Sélectionnez Créer à partir de zéro.
4. Dans Informations de base, entrez le nom de la fonction. Sous Runtime, choisissez un environnement d'exécution basé sur Python.
5. Pour Execution role (Rôle d'exécution), choisissez Create a new role with basic Lambda permissions (Créer un nouveau rôle avec les autorisations Lambda de base).

   Lambda crée un rôle d'exécution qui accorde à la fonction l'autorisation de télécharger des journaux sur Amazon. CloudWatch La fonction Lambda assume le rôle d'exécution lorsque vous appelez votre fonction et utilise le rôle d'exécution pour créer des informations d'identification pour le. AWS SDK
6. Choisissez l'onglet Code et ajoutez l'exemple de fonction Lambda.

Si vous n'avez pas encore créé le rôle de service que la fonction Lambda doit utiliser, vous devez le créer pour que l'exemple de fonction Lambda fonctionne. Pour en savoir plus sur la création d'un rôle de service pour la fonction Lambda, consultez. Création d'un rôle HealthLake de service à utiliser dans la fonction AWS Lambda utilisée pour décoder un JWT

```
import base64
import boto3
import logging
import json
import os
from urllib import request, parse

logger = logging.getLogger()
logger.setLevel(logging.INFO)

## Uses Secrets manager to gain access to the access key ID and secret access key
 for the authorization server
client = boto3.client('secretsmanager', region_name="region-of-datastore")
response = client.get_secret_value(SecretId='name-specified-by-customer-in-
secretsmanager')
secret = json.loads(response['SecretString'])
client_id = secret['client_id']
client_secret = secret['client_secret']


unencoded_auth = f'{client_id}:{client_secret}'
headers = {
  'Authorization': f'Basic {base64.b64encode(unencoded_auth.encode()).decode()}',
  'Content-Type': 'application/x-www-form-urlencoded'
}

auth_endpoint = os.environ['auth-server-base-url'] # Base URL of the Authorization
 server
user_role_arn = os.environ['iam-role-arn'] # The IAM role client application will
 use to complete the HTTP request on the datastore

def lambda_handler(event, context):
    if 'datastoreEndpoint' not in event or 'operationName' not in event or
 'bearerToken' not in event:
    return {}

    datastore_endpoint = event['datastoreEndpoint']
```

```
    operation_name = event['operationName']
    bearer_token = event['bearerToken']
    logger.info('Datastore Endpoint [{}], Operation Name:
 [{}]'.format(datastore_endpoint, operation_name))

    ## To validate the token
    auth_response = auth_with_provider(bearer_token)
    logger.info('Auth response: [{}]'.format(auth_response))
    auth_payload = json.loads(auth_response)
    ## Required parameters needed to be sent to the datastore endpoint for the HTTP
 request to go through
    auth_payload["isAuthorized"] = bool(auth_payload["active"])
    auth_payload["nbf"] = auth_payload["iat"]
    return {"authPayload": auth_payload, "iamRoleARN": user_role_arn}

## Access the server
def auth_with_provider(token):
    data = {'token': token, 'token_type_hint': 'access_token'}
    req = request.Request(url=auth_endpoint + '/v1/introspect',
 data=parse.urlencode(data).encode(), headers=headers)
    with request.urlopen(req) as resp:
    return resp.read().decode()
```

Modifier le rôle d'exécution d'une fonction Lambda

Après avoir créé la fonction Lambda, vous devez mettre à jour le rôle d'exécution afin d'inclure les autorisations nécessaires pour appeler Secrets Manager. Dans Secrets Manager, chaque secret que vous créez possède unARN. Pour appliquer le moindre privilège, le rôle d'exécution doit uniquement avoir accès aux ressources nécessaires à l'exécution de la fonction Lambda.

Vous pouvez modifier le rôle d'exécution d'une fonction Lambda en la recherchant dans la IAM console ou en choisissant Configuration dans la console Lambda. Pour en savoir plus sur la gestion de votre rôle d'exécution des fonctions Lambda, consultez. Rôle d'exécution Lambda

Example Rôle d'exécution de la fonction Lambda qui donne accès à **`GetSecretValue`**

L'ajout de l'IAMaction `GetSecretValue` au rôle d'exécution accorde l'autorisation nécessaire au fonctionnement de l'exemple de fonction Lambda.

```
{
  "Version": "2012-10-17",
```

```
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "secretsmanager:GetSecretValue",
      "Resource": "arn:aws:secretsmanager:your-region:your-aws-account-
id:secret:secret-name-DKodTA"
    }
  ]
}
```

À ce stade, vous avez créé une fonction Lambda qui peut être utilisée pour valider le jeton d'accès fourni dans le cadre de la FHIR REST demande envoyée à votre banque de HealthLake données SMART FHIR activée.

Création d'un rôle HealthLake de service à utiliser dans la fonction AWS Lambda utilisée pour décoder un JWT

Persona : Administrator IAM

Un utilisateur qui peut ajouter ou supprimer IAM des politiques et créer de nouvelles IAM identités.

Rôle de service

Un rôle de service est un IAMrôle qu'un service assume pour effectuer des actions en votre nom. Un administrateur IAM peut créer, modifier et supprimer une fonction du service à partir de IAM. Pour plus d'informations, consultez la section Créer un rôle pour déléguer des autorisations à un Service AWS dans le guide de IAM l'utilisateur.

Une fois le jeton JSON Web (JWT) décodé, l'autorisation dont Lambda a besoin doit également renvoyer IAM un rôle. ARN Ce rôle doit disposer des autorisations nécessaires pour exécuter la REST API demande, sinon il échouera en raison d'autorisations insuffisantes.

Lorsque vous configurez une politique personnalisée, IAM il est préférable d'accorder les autorisations minimales requises. Pour en savoir plus, consultez la section Appliquer les autorisations de moindre privilège dans le Guide de l'IAMutilisateur.

La création d'un rôle de HealthLake service à désigner dans la fonction Lambda d'autorisation nécessite deux étapes.

- Tout d'abord, vous devez créer une IAM politique. La politique doit spécifier l'accès aux FHIR ressources pour lesquelles vous avez fourni des étendues sur le serveur d'autorisation.
- Ensuite, vous devez créer le rôle de service. Lorsque vous créez le rôle, vous désignez une relation de confiance et vous y associez la politique que vous avez créée à la première étape. La relation de confiance désigne HealthLake le principal du service. Vous devez spécifier un magasin de HealthLake données ARN et un identifiant de AWS compte lors de cette étape.

Création d'une nouvelle IAM politique

Les étendues que vous définissez dans votre serveur d'autorisation déterminent les FHIR ressources auxquelles un utilisateur authentifié a accès dans un magasin de HealthLake données.

La IAM politique que vous créez peut être adaptée aux étendues que vous avez définies.

Les actions suivantes peuvent être définies dans l'`Action`élément d'une déclaration de IAM politique. Pour chacun `Action` des éléments du tableau, vous pouvez définir un`Resource types`. Dans HealthLake un magasin de données, le seul type de ressource pris en charge peut être défini dans l'`Resource`élément d'une déclaration de politique d'IAMautorisation.

Les FHIR ressources individuelles ne sont pas des ressources que vous pouvez définir en tant qu'élément d'une politique d'IAMautorisation.

Actions définies par HealthLake

Actions	Description	Nivea d'acc	Type de ressource (obligatoire)
CreateRes ource	Accorde l'autorisation de créer une ressource	Écrire	Banque de données ARN : arn:aws:h ealthlake : ::datastore/fhir/ **your-region 111122223333 your-datastore-id**
DeleteRes ource	Accorde l'autorisation de supprimer une ressource	Écrire	Banque de données ARN : arn:aws:h ealthlake : ::datastore/fhir/ **your-region 111122223333 your-datastore-id**
ReadResou rce	Accorde l'autorisation de lire une ressource	Lectu	Banque de données ARN : arn:aws:h ealthlake : ::datastore/fhir/ **your-region 111122223333 your-datastore-id**

Actions	Description	Nivea d'acc	Type de ressource (obligatoire)
SearchWithGet	Accorde l'autorisation de rechercher des ressources avec la GET méthode	Lectu	Banque de données ARN : arn:aws:healthlake : ::datastore/fhir/ **your-region 111122223333 your-datastore-id**
SearchWithPost	Accorde l'autorisation de rechercher des ressources avec la POST méthode	Lectu	Banque de données ARN : arn:aws:healthlake : ::datastore/fhir/ **your-region 111122223333 your-datastore-id**
StartFHIRExportJobWithPost	Accorde l'autorisation de commencer une tâche d'FHIRexportation avec GET	Écrire	Banque de données ARN : arn:aws:healthlake : ::datastore/fhir/ **your-region 111122223333 your-datastore-id**
UpdateResource	Accorde l'autorisation de mettre à jour des ressources	Écrire	Banque de données ARN : arn:aws:healthlake : ::datastore/fhir/ **your-region 111122223333 your-datastore-id**

Pour commencer, vous pouvez utiliser`AmazonHealthLakeFullAccess`. Cette politique autoriserait la lecture, l'écriture, la recherche et l'exportation de toutes les FHIR ressources présentes dans un magasin de données. Pour accorder des autorisations de lecture seule sur un magasin de données, utilisez. `AmazonHealthLakeReadOnlyAccess`

Pour en savoir plus sur la création d'une politique personnalisée à l'aide de AWS Management Console AWS CLI, ou IAMSDKs, voir Création de IAM politiques dans le Guide de IAM l'utilisateur.

Création d'un rôle de service pour HealthLake (IAMconsole)

Utilisez cette procédure pour créer un rôle de service. Lorsque vous créez un service, vous devez également définir une IAM politique.

Pour créer le rôle de service pour HealthLake (IAMconsole)

1. Connectez-vous à la IAM console AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/iam/.

2. Dans le panneau de navigation de la console IAM, choisissez Rôles.
3. Puis, choisissez Créer un rôle.
4. Sur la page Sélectionner une entité de confiance, choisissez Politique de confiance personnalisée.
5. Ensuite, sous Politique de confiance personnalisée, mettez à jour l'exemple de politique comme suit. Remplacez-le **`your-account-id`** par votre numéro de compte et ajoutez le magasin ARN de données que vous souhaitez utiliser dans vos tâches d'importation ou d'exportation.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Principal": {
            "Service": "healthlake.amazonaws.com"
        },
        "Condition": {
          "StringEquals": {
            "aws:SourceAccount": "your-account-id"
            },
            "ArnEquals": {
              "aws:SourceArn": "arn:aws:healthlake:your-region:your-account-
id:datastore/fhir/your-datastore-id"
            }
          }
        }
  ]
}
```

6. Ensuite, choisissez Suivant.
7. Sur la page Ajouter des autorisations, choisissez la politique que vous souhaitez que le HealthLake service adopte. Pour trouver votre politique, recherchez-la dans Politiques d'autorisations.
8. Choisissez ensuite Attach policy.
9. Ensuite, sur la page Nom, révision et création sous Nom du rôle, entrez un nom.
10. (Facultatif) Ensuite, sous Description, ajoutez une brève description de votre rôle.
11. Si possible, saisissez un nom de rôle ou le suffixe d'un nom de rôle vous permettant d'identifier l'objectif du rôle. Les noms de rôles doivent être uniques au sein de votre Compte AWS. Ils

ne sont pas sensibles à la casse. Par exemple, vous ne pouvez pas créer deux rôles nommés **`PRODROLE`** et **`prodrole`**. Différentes entités peuvent référencer le rôle et il n'est donc pas possible de modifier son nom après sa création.

12. Passez en revue les détails du rôle, puis choisissez Créer un rôle.

Pour savoir comment spécifier le rôle ARN dans l'exemple de fonction Lambda, consultez. Création d'une fonction AWS Lambda

Rôle d'exécution Lambda

Le rôle d'exécution d'une fonction Lambda est un IAM rôle qui accorde à la fonction l'autorisation d'accéder aux AWS services et aux ressources. Cette page fournit des informations sur la façon de créer, d'afficher et de gérer le rôle d'exécution d'une fonction Lambda.

Par défaut, Lambda crée un rôle d'exécution avec des autorisations minimales lorsque vous créez une nouvelle fonction Lambda à l'aide du. AWS Management Console Pour gérer les autorisations accordées dans le rôle d'exécution, consultez la section Création d'un rôle d'exécution dans la IAM console dans le guide du développeur Lambda.

L'exemple de fonction Lambda fourni dans cette rubrique utilise Secrets Manager pour masquer les informations d'identification du serveur d'autorisation.

Comme pour tout IAM rôle que vous créez, il est important de suivre les meilleures pratiques du moindre privilège. Au cours de la phase de développement, vous pouvez parfois accorder des autorisations au-delà de ce qui est requis. Avant de publier votre fonction dans l'environnement de production, une bonne pratique consiste à ajuster la stratégie de manière à inclure uniquement les autorisations requises. Pour plus d'informations, voir Appliquer le moindre privilège dans le guide de l'utilisateur. IAM

HealthLake Autoriser le déclenchement de votre fonction Lambda

HealthLake Vous pouvez donc invoquer la fonction Lambda en votre nom, vous devez procéder comme suit :

- Vous devez définir la `IdpLambdaArn` valeur égale à ARN la fonction Lambda que vous HealthLake souhaitez invoquer dans la `CreateFHIRDatastore` requête.
- Vous avez besoin d'une politique basée sur les ressources permettant HealthLake d'appeler la fonction Lambda en votre nom.

Lorsqu'il HealthLake reçoit une FHIR REST API demande SMART sur un magasin de HealthLake données FHIR activé, il a besoin d'autorisations pour appeler en votre nom la fonction Lambda spécifiée lors de la création du magasin de données. Pour accorder HealthLake l'accès, vous allez utiliser une politique basée sur les ressources. Pour en savoir plus sur la création d'une politique basée sur les ressources pour une fonction Lambda, voir Autoriser un AWS service à appeler une fonction Lambda dans le Guide du développeur.AWS Lambda

Provisionnement de la simultanéité pour votre fonction Lambda

> **Important**
>
> HealthLake exige que la durée d'exécution maximale de votre fonction Lambda soit inférieure à une seconde (1 000 millisecondes).
> Si votre fonction Lambda dépasse la limite de temps d'exécution, vous obtenez une TimeOutexception.

Pour éviter cette exception, nous vous recommandons de configurer la simultanéité provisionnée. En allouant la simultanéité provisionnée avant une augmentation des appels, vous pouvez vous assurer que toutes les demandes sont servies par des instances initialisées avec une faible latence. Pour en savoir plus sur la configuration de la simultanéité provisionnée, voir Configuration de la simultanéité provisionnée dans le Guide du développeur Lambda

Pour connaître la durée d'exécution moyenne de votre fonction Lambda, utilisez actuellement la page de surveillance de votre fonction Lambda sur la console Lambda. Par défaut, la console Lambda fournit un graphique de durée qui indique le temps moyen, minimum et maximum consacré par votre code de fonction au traitement d'un événement. Pour en savoir plus sur la surveillance des fonctions Lambda, consultez la section Surveillance des fonctions dans la console Lambda dans le guide du développeur Lambda.

Si vous avez déjà configuré la simultanéité pour votre fonction Lambda et que vous souhaitez la surveiller, consultez la section Surveillance de la simultanéité dans le guide du développeur Lambda.

Création d'un SMART magasin de HealthLake données non FHIR activé

Pour utiliser le FHIR framework SMART on HealthLake, créez un magasin de HealthLake données avec le `IdentityProviderConfiguration` paramètre spécifié dans votre reateFHIRDatastore

requête C. Dans le `IdentityProviderConfiguration` paramètre, vous spécifiez les informations suivantes :

- Définissez l'AuthorizationStrategyégal à`SMART_ON_FHIR_V1`.
- Définissez l'IdpLambdaArnéquivalent ARN de celui que AWS Lambda vous avez créé pour gérer le décodage des jetons avec votre serveur d'autorisation.
- Définissez les éléments de métadonnées spécifiés dans le serveur d'autorisation sous forme de JSON bloc. Ces éléments de métadonnées sont renvoyés dans le document de découverte.
- Facultatif : Activer FineGrainedAuthorizationEnabled. Spécifiez `True` l'utilisation de l'autorisation détaillée fournie par HealthLake

Vous pouvez créer un magasin de données SMART FHIR activé en utilisant le AWS Command Line Interface (AWS CLI) ou via l'un des supports AWS pris en chargeSDKs. La création d'un magasin de HealthLake données SMART FHIR activé n'est pas prise en charge à l'aide de la HealthLake console.

Utilisation du AWS CLI pour créer un SMART magasin de HealthLake données FHIR activé

Vous pouvez utiliser l'exemple de code suivant pour créer un magasin de HealthLake données SMART FHIR activé à l'aide du AWS CLI. Lorsque vous créez un magasin de HealthLake données SMART FHIR activé, vous devez spécifier le `identity-provider-configuration`paramètre.

Dans le `identity-provider-configuration` paramètre, vous pouvez éventuellement activer une autorisation précise en définissant la `FineGrainedAuthorizationEnabled` valeur égale à. `True` Pour en savoir plus sur l'autorisation détaillée, voirUtilisation d'une autorisation précise avec un magasin de SMART données FHIR activé HealthLake . L'exemple ci-dessous contient un caractère spécial \ pour indiquer les sauts de ligne ou comme caractère d'échappement. C'est pour des raisons de clarté.

```
aws healthlake create-fhir-datastore \
  --region us-east-1 \
  --datastore-name "your-data-store-name" \
  --datastore-type-version R4 \
  --preload-data-config PreloadDataType="SYNTHEA" \
  --sse-configuration '{ "KmsEncryptionConfig": { \
    "CmkType": "customer-managed-kms-key1",
    "KmsKeyId": "arn:aws:kms:us-east-1:your-account-id:key/your-key-id" } }' \
```

```
  --identity-provider-configuration  \
      '{"AuthorizationStrategy": "SMART_ON_FHIR_V1", \
      "FineGrainedAuthorizationEnabled": boolean-false-by-default, \
      "IdpLambdaArn": "arn:aws:lambda:your-region:your-account-id:function:your-lambda-
name" \
      "Metadata": "{\"issuer\":\"https://ehr.example.com\",\"jwks_uri\":\"https://
ehr.example.com/.well-known/jwks.json\",\"authorization_endpoint\":\"https://
ehr.example.com/auth/authorize\",\"token_endpoint\":\"https://ehr.token.com/auth/
token\",\"token_endpoint_auth_methods_supported\":[\"client_secret_basic\",\"foo\"],
\"grant_types_supported\":[\"client_credential\",\"foo\"],\"registration_endpoint\":
\"https://ehr.example.com/auth/register\",\"scopes_supported\":[\"openId\",\"profile\",
\"launch\"],\"response_types_supported\":[\"code\"],\"management_endpoint\":\"https://
ehr.example.com/user/manage\",\"introspection_endpoint\":\"https://ehr.example.com/
user/introspect\",\"revocation_endpoint\":\"https://ehr.example.com/user/revoke\",
\"code_challenge_methods_supported\":[\"S256\"],\"capabilities\":[\"launch-ehr\",\"sso-
openid-connect\",\"client-public\"]}"}'
```

En cas de succès, vous obtenez la JSON réponse suivante :

```
{
  "DatastoreArn": "arn:aws:healthlake:your-region:111122223333:datastore/fhir/your-
datastore-id",
  "DatastoreEndpoint": "https://healthlake.your-region.amazonaws.com/datastore/your-
datastore-id/r4/",
  "DatastoreId": "your-data-store-id",
  "DatastoreStatus": "data-store-creation-status"
}
```

Utilisation d'une autorisation précise avec un magasin de SMART données FHIR activé HealthLake

Les champs d'application à eux seuls ne vous fournissent pas les précisions nécessaires quant aux données auxquelles un demandeur est autorisé à accéder dans un magasin de données. L'utilisation d'une autorisation précise permet d'obtenir un niveau de spécificité plus élevé lors de l'octroi de l'accès à un magasin de HealthLake données FHIR non SMART activé. Pour utiliser une autorisation précise, définissez la `FineGrainedAuthorizationEnabled` valeur égale à `True` dans le `IdentityProviderConfiguration` paramètre de votre requête C. reateFHIRDatastore

Si vous avez activé l'autorisation détaillée, votre serveur d'autorisation renvoie une `fhirUser` étendue en `id_token` même temps que le jeton d'accès. Cela permet de récupérer des informations

sur l'utilisateur par l'application cliente. L'application cliente doit traiter la `fhirUser` réclamation comme celle URI d'une FHIR ressource représentant l'utilisateur actuel. Cette valeur peut être `Patient`, `Practitioner` ou `RelatedPerson`. La réponse du serveur d'autorisation inclut également une `user/` étendue qui définit les données auxquelles l'utilisateur peut accéder. Cela utilise la syntaxe définie pour les étendues liées aux étendues spécifiques aux FHIR ressources :

```
user/(fhir-resource | '*').('read' | 'write' | '*')
```

Vous trouverez ci-dessous des exemples de la manière dont l'autorisation précise peut être utilisée pour préciser davantage les types de FHIR ressources liés à l'accès aux données.

- Quand `fhirUser` une `Practitioner` autorisation précise détermine-t-elle le nombre de patients auxquels l'utilisateur peut accéder. L'accès à n'`fhirUser`est autorisé qu'aux patients auxquels le patient fait référence en `fhirUser` tant que médecin généraliste.

  ```
  Patient.generalPractitioner : [{Reference(Practitioner)}]
  ```

- Quand `fhirUser` un `Patient` ou `RelatedPerson` le patient référencé dans la demande est différent de `fhirUser` l'autorisation précise détermine l'accès `fhirUser` du patient demandé. L'accès est autorisé lorsqu'une relation est spécifiée dans la `Patient` ressource demandée.

  ```
  Patient.link.other : {Reference(Patient|RelatedPerson)}
  ```

Récupération du document SMART de découverte d'un magasin de HealthLake données non FHIR activé

Pour qu'une application cliente puisse effectuer une FHIR REST demande avec succès, elle doit rassembler les exigences d'autorisation définies dans le magasin de HealthLake données. Aucune autorisation (jeton porteur) n'est requise pour que cette demande aboutisse.

Pour ce faire, faites une `GET` demande et ajoutez-la `/.well-known/smart-configuration` au point de terminaison du magasin de données

```
GET https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/.well-known/smart-configuration
```

Cela renvoie le document de découverte du magasin de HealthLake données sous forme de JSON blob. Vous y trouverez le `authorization_endpoint` et les `token_endpoint` ainsi que les spécifications et capacités définies dans le magasin de HealthLake données.

```
{
  "authorization_endpoint": "https://oidc.example.com/authorize",
  "token_endpoint": "https://oidc.example.com/oauth/token",
  "capabilities": [
    "launch-ehr",
    "client-public"
  ]
}
```

URLsnécessaire au lancement réussi d'une application client

- Point de terminaison d'autorisation : URL nécessaire pour autoriser une application cliente ou un utilisateur.
- Point de terminaison du jeton : point de terminaison du serveur d'autorisation utilisé par l'application cliente pour communiquer avec elle.

Effectuer une FHIR REST API demande sur un magasin de HealthLake données SMART activé

Vous pouvez effectuer des FHIR REST API demandes SMART sur un magasin de HealthLake données FHIR activé. L'exemple suivant montre une demande provenant d'une application cliente contenant un JWT dans l'en-tête d'autorisation et montre comment Lambda doit décoder la réponse. Une fois que la demande d'application cliente est autorisée et authentifiée, elle doit recevoir un jeton porteur du serveur d'autorisation. Utilisez le jeton porteur dans l'en-tête d'autorisation lorsque vous envoyez une FHIR REST API demande SMART sur un magasin de HealthLake données FHIR activé.

```
GET https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/
Patient/[ID]
Authorization: Bearer auth-server-provided-bearer-token
```

Comme un jeton porteur a été trouvé dans l'en-tête d'autorisation et qu'aucune AWS IAM identité n'a été détectée, la HealthLake fonction Lambda spécifiée lors de la création du magasin de HealthLake

données SMART FHIR activé a été créé. Lorsque le jeton est décodé avec succès par votre fonction Lambda, voici un exemple de réponse envoyée à. HealthLake

```
{
  "authPayload": {
    "iss": "https://authorization-server-endpoint/oauth2/token", # The issuer
 identifier of the authorization server
    "aud": "https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/
r4/", # Required, data store endpoint
    "iat": 1677115637,  # Identifies the time at which the token was issued
    "nbf": 1677115637,  # Required, the earliest time the JWT would be valid
    "exp": 1997877061,  # Required, the time at which the JWT is no longer valid
    "isAuthorized": "true",  # Required, boolean indicating the request has been
 authorized
    "uid": "100101",  # Unique identifier returned by the auth server
    "scope": "system/*.*" # Required, the scope of the request
  },
  "iamRoleARN": "iam-role-arn" #Required, IAM role to complete the request
}
```

Configuration des ressources nécessaires à la mise en œuvre d'un SMART magasin de données non FHIR conforme

Cette rubrique décrit les ressources que vous devez fournir dans votre AWS compte externe HealthLake, la création d'un magasin de HealthLake données SMART FHIR activé et la manière dont une SMART application FHIR cliente interagirait avec un serveur d'autorisation et un magasin de HealthLake données.

Les étapes de ce flux de travail définissent les étapes de base relatives à SMART la manière dont les FHIR demandes sont traitées et aux ressources nécessaires pour qu'elles aboutissent.

Dans le cadre d'un processus SMART sur FHIR demande, trois applications fonctionnent ensemble :

- L'utilisateur final : En général, un patient ou un clinicien utilisant une FHIR application tierce SMART pour accéder aux données d'un magasin de HealthLake données.
- L'SMARTFHIRapplication active (appelée application cliente) : application qui souhaite accéder aux données présentes dans le magasin de HealthLake données.
- Le serveur d'autorisation : un serveur compatible OpenID Connect capable d'authentifier les utilisateurs et d'émettre des jetons d'accès.

- Le magasin de HealthLake données : un magasin de HealthLake données SMART FHIR activé qui utilise une fonction Lambda pour répondre aux FHIR REST demandes fournissant un jeton porteur.

Pour que ces applications fonctionnent ensemble, vous devez créer les ressources suivantes.

Nous vous recommandons de créer SMART le magasin de HealthLake données FHIR activé après avoir configuré le serveur d'autorisation, défini les étendues nécessaires et créé une AWS Lambda fonction pour gérer l'introspection des jetons.

1. Configuration d'un point de terminaison de serveur d'autorisation — Serveur d'autorisation

Pour utiliser le FHIR framework SMART on, vous devez configurer un serveur d'autorisation tiers capable de valider les FHIR REST demandes effectuées sur un magasin de données. Pour en savoir plus sur la configuration d'un point de terminaison de serveur d'autorisation compatible HealthLake, consultezExigences d'authentification SMART pour FHIR.

2. Définissez des étendues pour contrôler qui peut accéder à quelles données de votre magasin de HealthLake données sur votre serveur d'autorisation — Serveur d'autorisation

Le FHIR framework SMART on utilise des OAuth étendues pour déterminer à quelles FHIR ressources une demande authentifiée a accès et dans quelle mesure. La définition des étendues est un moyen de concevoir avec le moindre privilège. Pour en savoir plus sur les portées définies par le FHIR framework SMART on et prises en charge par, HealthLake voir,Supporté SMART sur les FHIR OAuth oscilloscopes par HealthLake.

3. Configurez une AWS Lambda fonction capable d'effectuer une introspection de jetons : votre compte AWS

Une FHIR REST demande envoyée par l'application cliente SMART sur un magasin de données non FHIR activé contiendra un jeton JSON Web (JWT). Pour en savoir plus sur la configuration d'une fonction Lambda capable de la décoder et de la valider, voir Décodage a. JWT

4. Créez un magasin SMART de HealthLake données non FHIR activé : votre AWS compte

Pour créer un magasin SMART de FHIR HealthLake données, vous devez fournir un`IdentityProviderConfiguration`. Pour en savoir plus sur les `IdentityProviderConfiguration` paramètres requis dans une reateFHIRDatastore requête C, voir Création d'un magasin de HealthLake données SMART FHIR activé.

Comment une application cliente lance et demande des données à partir d'un magasin de HealthLake données SMART FHIR activé

Cette section explique comment une application cliente se lance dans un FHIR contexte et est capable d'effectuer une FHIR REST demande avec succès sur un magasin de HealthLake données. SMART

1. L'application cliente envoie une **`GET`** demande à l'identifiant de ressource uniforme bien connu

Une application cliente SMART activée doit effectuer une `GET` demande pour trouver les points de terminaison d'autorisation de votre magasin de HealthLake données. Cela se fait via une demande d'identifiant uniforme de ressource bien connu (URI). Pour en savoir plus à ce sujet, consultez la section Extraction d'un document SMART de découverte d'un magasin de HealthLake données non FHIR activé.

2. Demande d'accès et champs d'application

L'application cliente utilise le point de terminaison d'autorisation du serveur d'autorisation afin que l'utilisateur puisse se connecter. Ce processus authentifie l'utilisateur. Les étendues sont utilisées pour définir les FHIR ressources de votre magasin de HealthLake données auxquelles une application cliente peut accéder. Pour en savoir plus sur la définition des étendues, consultezSupporté SMART sur les FHIR OAuth oscilloscopes par HealthLake.

3. Jetons d'accès

Maintenant que l'utilisateur est authentifié, une application cliente reçoit un jeton d'JWTaccès du serveur d'autorisation. Ce jeton est fourni lorsque l'application cliente envoie une FHIR REST demande à HealthLake. Pour en savoir plus sur le décodage à l'JWTaide d'une fonction Lambda, reportez-vous à la section. Exécution de la validation des jetons

4. Faire une FHIR REST demande SMART sur un magasin de HealthLake données FHIR activé

L'application cliente peut désormais envoyer une FHIR REST demande à un point de terminaison du magasin de HealthLake données à l'aide du jeton d'accès fourni par le serveur d'autorisation. Pour consulter un exemple de FHIR REST demande, consultezEffectuer une FHIR REST API demande sur un magasin de HealthLake données SMART activé.

5. Validation du jeton JWT d'accès

Pour valider le jeton d'accès envoyé dans la FHIR REST demande, utilisez une fonction Lambda. Pour savoir comment créer une fonction Lambda capable d'effectuer une introspection de jetons, consultez. Création d'une fonction AWS Lambda

Utilisation de la génération automatique de ressources basée sur le traitement du langage naturel (NLP) du type de FHIR DocumentReference ressource dans AWS HealthLake

> **Note**
>
> Après le 20 février 2023, les magasins de HealthLake données n'utilisent pas le traitement automatique du langage naturel intégré (NLP) par défaut. Si vous souhaitez activer cette fonctionnalité sur votre banque de données, consultez le Comment activer la fonction intégrée HealthLake de traitement du langage naturel ? chapitre Dépannage.
> Si vous avez activé la fonction intégrée d'Amazon Comprehend MedicalNLP, des frais vous seront facturés lors de la création ou de la mise à jour de `DocumentReference` ressources sur votre compte. AWS Pour plus de détails, consultez AWS HealthLake les tarifs.
> Amazon Comprehend Medical n'est pas disponible en Asie-Pacifique (Mumbai). HealthLake les banques de données créées dans la région Asie-Pacifique (Mumbai) ne prennent pas en charge le traitement intégré du langage naturel (NLP).

HealthLake vous fournit automatiquement un traitement du langage naturel intégré (NLP) à l'aide d'Amazon Comprehend Medical pour le traitement non structuré des données stockées dans `DocumentReference` le type de ressource. Pour ce faire, HealthLake appelez Amazon Comprehend `DetectEntities-V2` Medical`InferICD10-CM`, `InferRxNorm` API et Operations. Les résultats sont automatiquement ajoutés à la `DocumentReference` ressource sous forme d'extension. Lorsque les opérations d'Amazon Comprehend API Medical détectent `SIGN` des traits`SYMPTOM`, `DIAGNOSIS` `Linkage` un type de ressource est généré automatiquement. De nouvelles ressources de condition et d'observation sont créées à partir d'entités identifiées par les traits de `SIGN` `SYMPTOMDIAGNOSIS`, ou, et elles sont liées au document source avec cette ressource de liaison.

Pour les ressources générées par l'intégrationNLP, vous pouvez faire des `GET` demandes, mais la recherche de ces nouvelles ressources n'est pas prise en charge.

Pour en savoir plus sur la recherche dans ces extensions à l'aide HealthLake de l'intégration avec Athena, consultez. Interrogez votre banque HealthLake de données à l'aide de SQL

Table des matières

Comment Amazon Comprehend Medical est intégré à HealthLake

HealthLake déduit les données trouvées dans le type de `DocumentReference` ressource à l'aide d'Amazon Comprehend Medical. Les `DetectEntities-V2` opérations d'Amazon Comprehend API Medical `InferRxNorm` et détecte `InferICD10-CM` les problèmes de santé en tant que traits de caractère. Chaque opération fournit des informations différentes.

> ⚠ Prise en charge des langages
>
> Les opérations d'Amazon Comprehend API Medical détectent uniquement les entités médicales dans les textes en anglais.

- DetectEntities-V2 : inspecte le texte clinique de diverses entités médicales et renvoie des informations spécifiques les concernant, telles que la catégorie d'entité, l'emplacement et le score de confiance.
- Déduire ICD1 0-CM : détecte les affections médicales présentes dans le dossier d'un patient sous forme d'entités, et relie ces entités aux identificateurs conceptuels normalisés de la base de connaissances à ICD -10 CM du CDC National Center for Health Statistics, avec l'autorisation de l'Organisation mondiale de la santé (). WHO
- InferRxNorm: Détecte les médicaments en tant qu'entités répertoriées dans le dossier d'un patient et les lie aux identificateurs conceptuels normalisés de la RxNorm base de données de la National Library of Medicine.

Les caractéristiques prises en charge pour chaque API opération sont `SIGNSYMPTOM`, et`DIAGNOSIS`. Si des traits sont détectés, ils sont ajoutés en tant qu'extensions FHIR conformes à différents emplacements de votre banque de HealthLake données.

Emplacements où les extensions sont ajoutées.

- `DocumentReference`: Les résultats des opérations Amazon Comprehend API Medical sont ajoutés sous forme `extension` de fichier à chaque document trouvé dans DocumentReference le type de ressource. Les résultats de l'extension sont divisés en deux groupes. Vous pouvez les trouver dans les résultats en fonction de leurs`URL`.
  - `http://healthlake.amazonaws.com/system-generated-resources/`
    - Il s'agit de types de ressources qui ont été créés ou ajoutés par HealthLake.
  - `http://healthlake.amazonaws.com/aws-cm/`
    - Où le résultat brut des opérations d'Amazon Comprehend API Medical est ajouté à HealthLake votre banque de données.
- `Linkage`: Ce type de ressource est soit ajouté, soit créé à la suite de l'intégrationNLP. Une `GET` requête portant sur un sujet spécifique `Linkage` renvoie une liste de ressources liées. Pour savoir si a `Linkage` été ajouté par HealthLake, recherchez la paire `"tag": [{"display": "SYSTEM_GENERATED"}]` clé-valeur ajoutée. Pour en savoir plus sur les FHIR spécifications relatives à Linkage, voir Type de ressource : Liaison dans l'index de FHIRdocumentation.
- FHIRtypes de ressources générés à la suite des opérations d'Amazon Comprehend API Medical.
  - `Observation`: Les résultats des DetectEntities opérations Amazon Comprehend API Medical - V2 et ICD1 Infer 0-CM y sont ajoutés lorsque les traits sont ou. `SIGN` `SYMPTOM`
  - `Condition`: Les résultats des DetectEntities opérations -V2 d'Amazon Comprehend API Medical et ICD1 Infer 0-CM y sont ajoutés lorsque les traits le sont. `DIAGNOSIS`
  - `MedicationStatement`: Des résultats de l' InferRxNorm opération Amazon Comprehend API Medical y ont été ajoutés.

Intégration aux FHIR REST API opérations

Par défaut, les caractéristiques détectées par les opérations d'Amazon Comprehend API Medical ne sont pas renvoyées lors `GET` d'une demande.

Pour voir les résultats des NLP opérations intégrées pour ces types de ressources, vous devez spécifier une ressource connue`ID`.

- `Linkage`
- `Observation`
- `Condition`

- `MedicationStatement`

Les résultats des NLP opérations intégrées autres que le type de DocumentReference ressource ne sont disponibles qu'à l'aide d'une `GET` demande dont le contenu spécifié `ID` est connu pour contenir les résultats des opérations Amazon Comprehend API Medical.

Exemples de la manière dont les opérations d'Amazon Comprehend API Medical sont intégrées dans HealthLake

Exemple 1 : dossier patient ingéré dans une banque de HealthLake données

Voici un exemple de note clinique basée sur la rencontre d'un patient avec un professionnel de santé.

> **⚠ Données synthétiques**
>
> Le texte de cet exemple est un contenu synthétique et ne contient pas d'informations médicales personnelles (PHI).

```
1991-08-31

# Chief Complaint
- Headache
- Sinus Pain
- Nasal Congestion
- Sore Throat
- Pain with Bright Lights
- Nasal Discharge
- Cough


# History of Present Illness
Jerónimo599
 is a 4 month-old non-hispanic white male.

# Social History
 Patient has never smoked.


Patient comes from a middle socioeconomic background.
```

```
Patient currently has Aetna.

# Allergies
No Known Allergies.

# Medications
No Active Medications.

# Assessment and Plan
Patient is presenting with bee venom (substance), mold (organism), house dust
 mite (organism), animal dander (substance), grass pollen (substance), tree pollen
 (substance), lisinopril, sulfamethoxazole / trimethoprim, fish (substance).

## Plan

The patient was prescribed the following medications:
- astemizole 10 mg oral tablet
- nda020800 0.3 ml epinephrine 1 mg/ml auto-injector
The patient was placed on a careplan:
- self-care interventions (procedure)
```

Pour rappel, ces informations sont encodées au format base64 dans la DocumentReference ressource. Lorsque ce document est intégré HealthLake et que les API opérations Amazon Comprehend Medical sont terminées, pour voir les résultats, vous pouvez commencer par la `GET` demande sur le `DocumentReference` type de ressource.

```
 GET https://https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/
r4/eeb8005725ae22b35b4edbdc68cf2dfd/r4/DocumentReference
```

Lorsque les opérations d'Amazon Comprehend API Medical sont réussies, recherchez ces paires clé-valeur dans `extension` le lien suivant `"url": "http://healthlake.amazonaws.com/aws-cm/"`

```
{
  "url": "http://healthlake.amazonaws.com/aws-cm/status/",
  "valueString": "SUCCESS"
 },
 {
  "url": "http://healthlake.amazonaws.com/aws-cm/message/",
  "valueString": "The Amazon HealthLake integrated medical NLP operation was
 successful."
```

```
}
```

Les onglets suivants indiquent comment le dossier médical ingéré est enregistré dans votre banque de HealthLake données en fonction du type de ressource.

DocumentReference

Pour voir les résultats pour un seul type de `DocumentReference` ressource, faites une `GET` demande dans laquelle le `id` nom d'une ressource spécifique est fourni.

```
GET https://https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/eeb8005725ae22b35b4edbdc68cf2dfd/r4/DocumentReference/0e938f03-da7f-4178-acd8-eea9586c46ed
```

En cas de succès, vous obtenez un code de `200` HTTP réponse et la JSON réponse suivante (qui a été tronquée pour des raisons de clarté).

Voici la `http://healthlake.amazonaws.com/system-generated-resources/` portion. Vous pouvez voir qu'un nouveau `Linkage/`*`e366d29f-2c22-4c19-866e-09603937935a`* a été ajouté. Vous pouvez également voir où des résultats basés sur des inférences ont HealthLake été ajoutés à des types `Observation` de `Condition` ressources spécifiques.

Pour voir comment ces types de ressources ont été modifiés, sélectionnez les onglets correspondants.

```
{
  "extension": [
   {
    "url": "http://healthlake.amazonaws.com/linkage",
    "valueReference": {
     "reference": "Linkage/e366d29f-2c22-4c19-866e-09603937935a"
    }
   },
   {
    "url": "http://healthlake.amazonaws.com/nlp-entity",
    "valueReference": {
     "reference": "Observation/c6e0a3ff-7a17-4d8b-bfd0-d02d7da090c5"
    }
   },
   {
    "url": "http://healthlake.amazonaws.com/nlp-entity",
    "valueReference": {
```

```
     "reference": "Condition/0854e1f3-894d-448e-a8d9-3af5b9902baf"
    }
   }
  ],
  "url": "http://healthlake.amazonaws.com/system-generated-resources/"
 }
```

Linkage

Pour voir les résultats pour un seul type de `Linkage` ressource, faites une `GET` demande dans laquelle le `ID` nom d'une ressource spécifique est fourni.

```
GET https://https://healthlake.your-region.amazonaws.com/
datastore/your-datastore-id/r4/eeb8005725ae22b35b4edbdc68cf2dfd/r4/
Linkage/e366d29f-2c22-4c19-866e-09603937935a
```

En cas de succès, vous obtenez un code de `200` HTTP réponse et la JSON réponse tronquée suivante.

La réponse contient l'`item`élément. Dans ce document, la paire clé-valeur `"type": "source"` indique l'`DocumentReference`entrée spécifique utilisée pour modifier la `Condition` et `Observations` répertoriée sous la paire `"type": "alternate"` clé-valeur.

Vous voyez également l'*`meta`*élément, ainsi que la paire clé-valeur correspondante*`"tag": [{"display": "SYSTEM_GENERATED"}]`*, indiquant que ces ressources ont été créées par. HealthLake

```
{
  "resourceType": "Linkage",
  "id": "e366d29f-2c22-4c19-866e-09603937935a",
  "active": true,
  "item":
  [
   {
    "type": "alternate",
    "resource": {
     "reference": "Observation/c6e0a3ff-7a17-4d8b-bfd0-d02d7da090c5",
     "type": "Observation"
    }
   },
   {
    "type": "alternate",
```

```
    "resource": {
     "reference": "Condition/9d5c1ef6-f822-4faf-b55f-7c70f2a4aa8d",
     "type": "Condition"
    }
   },
   {
    "type": "source",
    "resource": {
     "reference": "DocumentReference/0e938f03-da7f-4178-acd8-eea9586c46ed",
     "type": "DocumentReference"
    }
   }
  ],
  "meta": {
   "lastUpdated": "2022-10-21T19:38:31.327Z",
   "tag": [{
    "display": "SYSTEM_GENERATED"
   }]
  }
 }
```

Resource type: Observation

Pour voir les résultats pour un seul type de `Observation` ressource, faites une `GET` demande dans laquelle le `ID` nom d'une ressource spécifique est fourni.

```
GET https://https://healthlake.your-region.amazonaws.com/
datastore/your-datastore-id/r4/eeb8005725ae22b35b4edbdc68cf2dfd/r4/
Observation/e366d29f-2c22-4c19-866e-09603937935a
```

Les résultats des opérations d'Amazon Comprehend API Medical sont modifiés selon les éléments suivants `code` :`meta`, `modifierExtension` et.

`code`

Un élément de type`CodeableConcept`. Pour en savoir plus, consultez CodeableConceptl'index de la FHIR documentation.

HealthLake ajoute les trois paires clé-valeur suivantes.

- `"system": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/"`: Où le URL fait référence à une opération Amazon Comprehend API Medical spécifique. Dans ce cas, inférez ICD1 0 cm.

- `"code": "A52.06"`: Où se `A52.06` trouve le code ICD -10-CM qui identifie le concept trouvé dans la base de connaissances des Centers for Disease Control.
- `"display": "Other syphilitic heart involvement"`: Où se `"Other syphilitic heart involvement"` trouve la description longue du code ICD -10-CM dans l'ontologie.

La JSON réponse tronquée suivante contient uniquement l'`code`élément.

```
"code": {
  "coding":
  [
    {
   "system": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/",
   "code": "A52.06",
   "display": "Other syphilitic heart involvement"
    }
  ],
  "text": "Other syphilitic heart involvement"
 }
```

Pour comprendre dans quelle mesure le modèle est sûr que le code ICD -10-CM attribué est correct, utilisez l'`modifierExtension`élément.

`meta`

L'`meta`élément contient des métadonnées qui indiquent s'il contient des informations qui ont été ajoutées par les opérations d'Amazon Comprehend API Medical. `code`

La JSON réponse tronquée suivante contient uniquement l'`meta`élément.

```
"meta": {
  "lastUpdated": "2022-10-21T19:38:30.879Z",
  "tag": [{
   "display": "SYSTEM_GENERATED"
  }]
 }
```

`modifierExtension`

L'`modifierExtension`élément contient plus de détails sur le niveau de confiance des codes assignés trouvés dans l'`code`élément. Il comporte également des paires clé-valeur qui fournissent

un lien vers l'original DocumentReference utilisé pour générer les résultats et le type de ressource de liaison associé.

Pour chaque `coding` élément ajouté, vous verrez un `entity-score` et un `entity-Concept-Score` ajoutés aumodifierExtension. Pour chaque valeur de la paire clé-valeur, vous voyez un score. En effet`entity-score`, ce score correspond au niveau de confiance d'Amazon Comprehend Medical quant à la précision de la détection. En effet`entity-Concept-Score`, ce score est le niveau de confiance d'Amazon Comprehend Medical quant à la précision du lien entre l'entité et ICD un concept de -10 CM.

La JSON réponse tronquée suivante contient uniquement l'`modifierExtension`élément.

```
"modifierExtension": [{
   "url": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/aws-cm-icd10-entity-
score",
   "valueDecimal": 0.45005733
  },
  {
   "url": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/aws-cm-icd10-entity-
Concept-Score",
   "valueDecimal": 0.1111792
  },
  {
   "url": "http://healthlake.amazonaws.com/system-generated-linkage",
   "valueReference": {
    "reference": "Linkage/e366d29f-2c22-4c19-866e-09603937935a"
   }
  },
  {
   "url": "http://healthlake.amazonaws.com/source-document-reference",
   "valueReference": {
    "reference": "DocumentReference/0e938f03-da7f-4178-acd8-eea9586c46ed"
   }
  }
 ]
```

JSONRéponse complète

```
{
  "subject": {
   "reference": "Patient/0679b7b7-937d-488a-b48d-6315b8e7003b"
  },
```

```
  "resourceType": "Observation",
  "status": "unknown",
  "code": {
   "coding": [{
    "system": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/",
    "code": "A52.06",
    "display": "Other syphilitic heart involvement"
   }],
   "text": "Other syphilitic heart involvement"
  },
  "meta": {
   "lastUpdated": "2022-10-21T19:38:30.879Z",
   "tag": [{
    "display": "SYSTEM_GENERATED"
   }]
  },
  "modifierExtension": [{
    "url": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/aws-cm-icd10-entity-
score",
    "valueDecimal": 0.45005733
   },
   {
    "url": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/aws-cm-icd10-entity-
Concept-Score",
    "valueDecimal": 0.1111792
   },
   {
    "url": "http://healthlake.amazonaws.com/system-generated-linkage",
    "valueReference": {
     "reference": "Linkage/e366d29f-2c22-4c19-866e-09603937935a"
    }
   },
   {
    "url": "http://healthlake.amazonaws.com/source-document-reference",
    "valueReference": {
     "reference": "DocumentReference/0e938f03-da7f-4178-acd8-eea9586c46ed"
    }
   }
  ],
  "id": "7e88c7c5-21a5-4dd7-8fc2-a02474fba583"
 }
```

Condition

Pour voir les résultats pour un seul type de `Condition` ressource, faites une `GET` demande dans laquelle le `ID` nom d'une ressource spécifique est fourni.

```
GET https://https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/eeb8005725ae22b35b4edbdc68cf2dfd/r4/Condition/b06d343d-ddb8-4f36-82cb-853fcd434dfd
```

Les résultats des opérations d'Amazon Comprehend API Medical sont modifiés selon les éléments suivants `code` :`meta`, `modifierExtension` et.

`code`

Un élément de type`CodeableConcept`. Pour en savoir plus, consultez CodeableConceptl'index de la FHIR documentation.

HealthLake ajoute les trois paires clé-valeur suivantes.

- `"system": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/"`: Où le URL fait référence à une opération Amazon Comprehend API Medical spécifique. Dans ce cas, inférez ICD1 0 cm.
- `"code": "I70.0"`: Où se `A52.06` trouve le code ICD -10-CM qui identifie le concept trouvé dans la base de connaissances des Centers for Disease Control.
- `"display": "Atherosclerosis of aorta"`: Où se `"Other syphilitic heart involvement"` trouve la description longue du code ICD -10-CM dans l'ontologie.

La JSON réponse tronquée suivante contient uniquement l'`code`élément.

```
"code": {
  "coding":
  [
    {
   "system": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/",
   "code": "I70.0",
   "display": "Atherosclerosis of aorta"
    }
  ],
  "text": "Atherosclerosis of aorta"
 }
```

Pour comprendre dans quelle mesure le modèle est sûr que le code ICD -10-CM attribué est correct, utilisez l'`modifierExtension`élément.

`meta`

L'`meta`élément contient des métadonnées qui indiquent s'il contient des informations qui ont été ajoutées par les opérations d'Amazon Comprehend API Medical. `code`

La JSON réponse tronquée suivante contient uniquement l'`meta`élément.

```
"meta": {
  "lastUpdated": "2022-10-21T19:38:30.877Z",
  "tag": [{
   "display": "SYSTEM_GENERATED"
  }]
 }
```

`modifierExtension`

L'`modifierExtension`élément contient plus de détails sur le niveau de confiance des codes assignés trouvés dans l'`code`élément. Il comporte également des paires clé-valeur qui fournissent un lien vers l'original DocumentReference utilisé pour générer les résultats et le type de ressource de liaison associé.

Pour chaque `coding` élément ajouté, vous verrez un `entity-score` et un `entity-Concept-Score` ajoutés aumodifierExtension. Pour chaque valeur de la paire clé-valeur, vous voyez un score. En effet`entity-score`, ce score correspond au niveau de confiance d'Amazon Comprehend Medical quant à la précision de la détection. En effet`entity-Concept-Score`, ce score est le niveau de confiance d'Amazon Comprehend Medical quant à la précision du lien entre l'entité et ICD un concept de -10 CM.

La JSON réponse tronquée suivante contient uniquement l'`modifierExtension`élément.

```
"modifierExtension": [{
   "url": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/aws-cm-icd10-entity-
score",
   "valueDecimal": 0.94417894
  },
  {
   "url": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/aws-cm-icd10-entity-
Concept-Score",
```

```
   "valueDecimal": 0.8458298
  },
  {
   "url": "http://healthlake.amazonaws.com/system-generated-linkage",
   "valueReference": {
    "reference": "Linkage/e366d29f-2c22-4c19-866e-09603937935a"
   }
  },
  {
   "url": "http://healthlake.amazonaws.com/source-document-reference",
   "valueReference": {
    "reference": "DocumentReference/0e938f03-da7f-4178-acd8-eea9586c46ed"
   }
  }
 ]
```

JSONRéponse complète

```
{
  "subject": {
   "reference": "Patient/0679b7b7-937d-488a-b48d-6315b8e7003b"
  },
  "resourceType": "Condition",
  "code": {
   "coding": [{
    "system": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/",
    "code": "I70.0",
    "display": "Atherosclerosis of aorta"
   }],
   "text": "Atherosclerosis of aorta"
  },
  "meta": {
   "lastUpdated": "2022-10-21T19:38:30.877Z",
   "tag": [{
    "display": "SYSTEM_GENERATED"
   }]
  },
  "modifierExtension": [{
    "url": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/aws-cm-icd10-entity-
score",
    "valueDecimal": 0.94417894
   },
   {
```

```
    "url": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/aws-cm-icd10-entity-
Concept-Score",
    "valueDecimal": 0.8458298
   },
   {
    "url": "http://healthlake.amazonaws.com/system-generated-linkage",
    "valueReference": {
     "reference": "Linkage/e366d29f-2c22-4c19-866e-09603937935a"
    }
   },
   {
    "url": "http://healthlake.amazonaws.com/source-document-reference",
    "valueReference": {
     "reference": "DocumentReference/0e938f03-da7f-4178-acd8-eea9586c46ed"
    }
   }
  ],
  "id": "b06d343d-ddb8-4f36-82cb-853fcd434dfd"
 }
```

Exemple 2 : A **`DocumentReference`** contenant le type de MedicationStatement ressource

Voici un exemple de note clinique basée sur la rencontre d'un patient avec un professionnel de santé.

> ⚠ Données synthétiques
>
> Le texte de cet exemple est un contenu synthétique et ne contient pas d'informations médicales personnelles (PHI).

```
Tom is not prescribed Advil
```

Les onglets suivants montrent comment le dossier médical ingéré est enregistré dans votre banque de HealthLake données en fonction du type de ressource.

DocumentReference

Pour voir les résultats pour un seul type de `DocumentReference` ressource, faites une `GET` demande dans laquelle le `ID` nom d'une ressource spécifique est fourni.

```
GET https://https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/eeb8005725ae22b35b4edbdc68cf2dfd/r4/DocumentReference/c549125d-a218-421f-b8bf-23614c5e796c
```

En cas de succès, vous obtenez un code de `200` HTTP réponse et la JSON réponse tronquée suivante.

La paire clé-valeur indique que les `"url": "http://healthlake.amazonaws.com/system-generated-resources/"` types de ressources qu'elle contient `extension` ont été ajoutés par les opérations d'Amazon API Comprehend Medical. Vous pouvez voir le nouveau type de `Linkage` ressource et plusieurs `MedicationStatement` ressources.

```
"extension": [{
   "extension": [{
     "url": "http://healthlake.amazonaws.com/linkage",
     "valueReference": {
      "reference": "Linkage/394bb244-177b-4409-8657-26b20ed56dd7"
     }
    },
    {
     "url": "http://healthlake.amazonaws.com/nlp-entity",
     "valueReference": {
      "reference": "MedicationStatement/cbf6af10-b0b9-451c-bdde-99611e3498a8"
     }
    },
    {
     "url": "http://healthlake.amazonaws.com/nlp-entity",
     "valueReference": {
      "reference": "MedicationStatement/9a89b0d3-6681-45ca-9926-27951edce5c7"
     }
    },
    {
     "url": "http://healthlake.amazonaws.com/nlp-entity",
     "valueReference": {
      "reference": "MedicationStatement/4a01f6c8-5f3a-4122-80ab-405312f96aa2"
     }
    },
    {
     "url": "http://healthlake.amazonaws.com/nlp-entity",
     "valueReference": {
      "reference": "MedicationStatement/fbfb77d8-70cf-4579-b4c0-d6fe3c01656b"
     }
```

```
    },
    {
     "url": "http://healthlake.amazonaws.com/nlp-entity",
     "valueReference": {
      "reference": "MedicationStatement/1340c9ce-9c48-4bf9-9b2f-d0ab027f5e0b"
     }
    }
   ],
   "url": "http://healthlake.amazonaws.com/system-generated-resources/"
  }
```

Linkage

Pour voir les résultats pour un seul type de `Linkage` ressource, faites une `GET` demande dans laquelle le `ID` nom d'une ressource spécifique est fourni.

```
GET https://https://healthlake.your-region.amazonaws.com/
datastore/your-datastore-id/r4/eeb8005725ae22b35b4edbdc68cf2dfd/r4/
Linkage/394bb244-177b-4409-8657-26b20ed56dd7
```

En cas de succès, vous obtenez un code de `200` HTTP réponse et la JSON réponse suivante.

La réponse contient l'`item`élément. La paire clé-valeur y `"type": "source"` indique l'`DocumentReference`entrée spécifique utilisée pour modifier les types de `MedicationStatement` ressources.

Vous pouvez également voir l'`meta`élément et une paire clé-valeur correspondante`"tag": [{"display": "SYSTEM_GENERATED"}]`, indiquant que ces ressources ont été créées par. HealthLake

```
{
  "resourceType": "Linkage",
  "id": "394bb244-177b-4409-8657-26b20ed56dd7",
  "active": true,
  "item": [{
    "type": "alternate",
    "resource": {
     "reference": "MedicationStatement/cbf6af10-b0b9-451c-bdde-99611e3498a8",
     "type": "MedicationStatement"
    }
   },
   {
```

```
    "type": "alternate",
    "resource": {
     "reference": "MedicationStatement/9a89b0d3-6681-45ca-9926-27951edce5c7",
     "type": "MedicationStatement"
    }
   },
   {
    "type": "alternate",
    "resource": {
     "reference": "MedicationStatement/4a01f6c8-5f3a-4122-80ab-405312f96aa2",
     "type": "MedicationStatement"
    }
   },
   {
    "type": "alternate",
    "resource": {
     "reference": "MedicationStatement/fbfb77d8-70cf-4579-b4c0-d6fe3c01656b",
     "type": "MedicationStatement"
    }
   },
   {
    "type": "alternate",
    "resource": {
     "reference": "MedicationStatement/1340c9ce-9c48-4bf9-9b2f-d0ab027f5e0b",
     "type": "MedicationStatement"
    }
   },
   {
    "type": "source",
    "resource": {
     "reference": "DocumentReference/c549125d-a218-421f-b8bf-23614c5e796c",
     "type": "DocumentReference"
    }
   }
  ],
  "meta": {
   "lastUpdated": "2022-10-24T20:05:03.501Z",
   "tag": [{
    "display": "SYSTEM_GENERATED"
   }]
  }
 }
```

MedicationStatement

Pour voir les résultats pour un seul type de `MedicationStatement` ressource, faites une `GET` demande dans laquelle le `ID` nom d'une ressource spécifique est fourni.

```
GET https://https://healthlake.your-region.amazonaws.com/
datastore/your-datastore-id/r4/eeb8005725ae22b35b4edbdc68cf2dfd/r4/
MedicationStatement/9a89b0d3-6681-45ca-9926-27951edce5c7
```

Le type de MedicationStatement ressource correspond à l'endroit où se trouvent les résultats de l'opération Amazon Comprehend InferRxNorm API Medical. Les résultats sont modifiés selon les éléments suivants : `medicationCodeableConceptmeta`, et`modifierExtension`.

medicationCodeableConcept

Un élément de type`CodeableConcept`. Pour en savoir plus, consultez CodeableConceptl'index de la FHIR documentation.

HealthLake ajoute les trois paires clé-valeur suivantes.

- `"system": ""http://healthlake.amazonaws.com/aws-cm/infer-rxnorm/`: Où le URL fait référence à une opération Amazon Comprehend API Medical spécifique. Dans ce cas, InferRxNorm.
- `"code": "731533"`: Où se `731533` trouve un identifiant RxNorm conceptuel, également connu sous le nom de RxCUI.
- `"display": "ibuprofen 200 MG Oral Capsule [Advil]"`: Où se `ibuprofen 200 MG Oral Capsule [Advil]` trouve la description du RxNorm concept.

La JSON réponse tronquée suivante contient uniquement l'`MedicationStatement`élément.

```
"medicationCodeableConcept": {
  "coding": [
   {
    "system": "http://healthlake.amazonaws.com/aws-cm/infer-rxnorm/",
    "code": "731533",
    "display": "ibuprofen 200 MG Oral Capsule [Advil]"
   }
  ]
 }
```

meta

L'`meta`élément contient des métadonnées qui indiquent s'il contient des informations qui ont été ajoutées par les opérations d'Amazon Comprehend API Medical. `code`

La JSON réponse tronquée suivante contient uniquement l'`meta`élément.

```
"meta": {
  "lastUpdated": "2022-10-24T20:05:02.800Z",
  "tag": [
   {
    "display": "SYSTEM_GENERATED"
   }
  ]
 }
```

modifierExtension

L'`modifierExtension`élément contient des paires clé-valeur qui fournissent un lien vers l'original DocumentReference utilisé pour générer les résultats et le type de ressource de liaison associé.

```
"modifierExtension": [
   {
    "url": "http://healthlake.amazonaws.com/system-generated-linkage",
    "valueReference": {
     "reference": "Linkage/394bb244-177b-4409-8657-26b20ed56dd7"
    }
   },
   {
    "url": "http://healthlake.amazonaws.com/source-document-reference",
    "valueReference": {
     "reference": "DocumentReference/c549125d-a218-421f-b8bf-23614c5e796c"
    }
   }
 ]
```

Paramètres de recherche

Le tableau suivant répertorie les attributs consultables pour la médecine NLP intégrée.

Paramètres de recherche

Paramètres de recherche	Trouve des correspondances pour
detectEntities-catégorie-entité	Catégorie d'entité au sein de la DetectEntities sous-extension de l'extension AWS CM
detectEntities-entité-texte	Texte d'entité dans la DetectEntities sous-extension de l'extension AWS CM
detectEntities-type d'entité	Type d'entité dans la DetectEntities sous-extension de l'extension AWS CM
detectEntities-entity-score	Entity Score au sein de la DetectEntities sous-extension de l'extension AWS CM
infer-icd10 cm-entity-text	Texte d'entité dans la sous-extension Infer ICD1 0CM au sein de l'extension CM AWS
infer-icd10 cm-entity-score	Score d'entité dans la sous-extension Infer ICD1 0CM au sein de l'extension CM AWS
infer-icd10 cm-entity-concept-code	Code de concept d'entité dans la sous-extension Infer ICD1 0CM au sein de l'extension CM AWS
infer-icd10 cm-entity-concept-description	Description du concept d'entité dans la sous-extension ICD1 Infer 0CM de l'extension CM AWS
infer-icd10 cm-entity-concept-score	Score du concept d'entité dans la sous-extension Infer ICD1 0CM au sein de l'extension CM AWS
infer-rxnorm-entity-score	Entity Score au sein de la InferRxNorm sous-extension de l'extension AWS CM
infer-rxnorm-entity-text	Texte d'entité dans la InferRxNorm sous-extension de l'extension AWS CM
infer-rxnorm-entity-concept-code	Code de concept d'entité dans la InferRxNorm sous-extension de l'extension AWS CM

Paramètres de recherche	Trouve des correspondances pour
infer-rxnorm-entity-concept-description	Description du concept d'entité dans la InferRxNorm sous-extension de l'extension AWS CM
infer-rxnorm-entity-concept-score	Entity Concept Score au sein de la InferRxNorm sous-extension de l'extension AWS CM

Pour répondre aux critères où `EntityText` et `EntityCategory` font partie de la même entité, HealthLake propose une recherche spéciale. Le tableau suivant décrit les paramètres de recherche spéciaux pris en charge dans HealthLake.

Paramètres de recherche

Paramètres de recherche	Allumettes retournées
detectEntities-entity-text-category	S'il existe au moins une entité dans la DetectEntities sous-extension qui correspond à la fois au entityText etentityCategory.
detectEntities-entity-type-score	S'il existe au moins une entité dans la DetectEntities sous-extension qui correspond à la fois au entityType etentityScore.
detectEntities-entity-text-score	S'il existe au moins une entité dans la DetectEntities sous-extension qui correspond à la fois au entityText etentityScore.
detectEntities-entity-text-type	S'il existe au moins une entité dans la DetectEntities sous-extension qui correspond à la fois au entityText etentityType.
detectEntities-entity-category-score	S'il existe au moins une entité qui correspond à la fois au entityCategory etentityScore.
code infer-icd10 cm-entity-text-concept	S'il existe au moins une entité dans la sous-extension Infer ICD1 0CM qui correspond au code entityText

Paramètres de recherche	Allumettes retournées
	et qu'il y en a au moins une conceptCode pour cette entité qui correspond au code.
score de infer-icd10 cm-entity-text-concept	S'il y a au moins une entité dans la sous-extension Infer ICD1 0CM qui correspond au score entityText et qu'il y en a au moins une conceptScore pour cette entité qui correspond au score.
infer-icd10 - score conceptuel cm-entity-concept-description	S'il existe au moins un concept au sein de l'entité de la sous-extension Infer ICD1 0CM qui correspond à la description du concept et au. conceptScore
infer-rxnorm-entity-text-code conceptuel	S'il existe au moins une entité dans la InferRxNorm sous-extension qui correspond au code entityText et qu'il y en a au moins une conceptCode pour cette entité qui correspond au code.
infer-rxnorm-entity-text-concept score	S'il y a au moins une entité dans la InferRxNorm sous-extension qui correspond au score entityText et qu'il y en a au moins une conceptScore pour cette entité qui correspond au score.
infer-rxnorm-entity-concept-description-concept-score	S'il existe au moins un concept au sein de l'entité de la InferRxNorm sous-extension qui correspond à la description du concept et auconceptScore.

Sécurité dans AWS HealthLake

La sécurité du cloud AWS est la priorité absolue. En tant que AWS client, vous bénéficiez d'un centre de données et d'une architecture réseau conçus pour répondre aux exigences des entreprises les plus sensibles en matière de sécurité.

La sécurité est une responsabilité partagée entre vous AWS et vous. Le modèle de responsabilité partagée décrit cela comme la sécurité du cloud et la sécurité dans le cloud :

- La sécurité du cloud AWS est chargée de protéger l'infrastructure qui exécute AWS les services dans le AWS cloud. AWS vous fournit également des services que vous pouvez utiliser en toute sécurité. Des auditeurs tiers testent et vérifient régulièrement l'efficacité de notre sécurité dans le cadre des programmes de AWS conformité Programmes de de conformité. Pour en savoir plus sur les programmes de conformité qui s'appliquent à HealthLake, voir AWSServices concernés par programme de conformité AWS .
- Sécurité dans le cloud — Votre responsabilité est déterminée par le AWS service que vous utilisez. Vous êtes également responsable d'autres facteurs, y compris de la sensibilité de vos données, des exigences de votre entreprise, ainsi que de la législation et de la réglementation applicables.

Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lors de son utilisation HealthLake. Les rubriques suivantes expliquent comment procéder à la configuration HealthLake pour atteindre vos objectifs de sécurité et de conformité. Vous apprendrez également à utiliser d'autres AWS services qui vous aident à surveiller et à sécuriser vos HealthLake ressources.

Rubriques

Protection des données dans AWS HealthLake

Le modèle de responsabilité AWS partagée de s'applique à la protection des données dans AWS HealthLake. Comme décrit dans ce modèle, AWS est chargé de protéger l'infrastructure mondiale qui gère tous les AWS Cloud. La gestion du contrôle de votre contenu hébergé sur cette infrastructure relève de votre responsabilité. Vous êtes également responsable des tâches de configuration et de gestion de la sécurité des Services AWS que vous utilisez. Pour plus d'informations sur la confidentialité des données, consultez la section Confidentialité des données FAQ. Pour plus d'informations sur la protection des données en Europe, consultez le modèle de responsabilitéAWS partagée et le billet de GDPR blog sur le blog sur la AWS sécurité.

À des fins de protection des données, nous vous recommandons de protéger les Compte AWS informations d'identification et de configurer les utilisateurs individuels avec AWS IAM Identity Center ou AWS Identity and Access Management (IAM). Ainsi, chaque utilisateur se voit attribuer uniquement les autorisations nécessaires pour exécuter ses tâches. Nous vous recommandons également de sécuriser vos données comme indiqué ci-dessous :

- Utilisez l'authentification multifactorielle (MFA) pour chaque compte.
- UtilisezSSL/TLSpour communiquer avec les AWS ressources. Nous avons besoin de la TLS version 1.2 et recommandons la TLS version 1.3.
- Configuration API et journalisation de l'activité des utilisateurs avec AWS CloudTrail. Pour plus d'informations sur l'utilisation des CloudTrail sentiers pour capturer AWS des activités, consultez la section Utilisation des CloudTrail sentiers dans le guide de AWS CloudTrail l'utilisateur.
- Utilisez des solutions de AWS chiffrement, ainsi que tous les contrôles de sécurité par défaut qu'ils contiennent Services AWS.
- Utilisez des services de sécurité gérés avancés tels qu’Amazon Macie, qui contribuent à la découverte et à la sécurisation des données sensibles stockées dans Amazon S3.
- Si vous avez besoin de FIPS 140 à 3 modules cryptographiques validés pour accéder AWS via une interface de ligne de commande ou unAPI, utilisez un point de terminaison. FIPS Pour plus d'informations sur les FIPS points de terminaison disponibles, voir Federal Information Processing Standard (FIPS) 140-3.

Nous vous recommandons fortement de ne jamais placer d’informations confidentielles ou sensibles, telles que les adresses e-mail de vos clients, dans des balises ou des champs de texte libre tels que le champ Nom. Cela inclut lorsque vous travaillez avec HealthLake ou d'autres Services AWS utilisateurs de la consoleAPI, AWS CLI, ou AWS SDKs. Toutes les données que vous entrez dans

des balises ou des champs de texte de forme libre utilisés pour les noms peuvent être utilisées à des fins de facturation ou dans les journaux de diagnostic. Si vous fournissez un URL à un serveur externe, nous vous recommandons vivement de ne pas inclure d'informations d'identification dans le URL afin de valider votre demande auprès de ce serveur.

Chiffrement chez REST for AWS HealthLake

HealthLake fournit un chiffrement par défaut pour protéger les données sensibles des clients au repos en utilisant une AWS clé Key Management Service (AWSKMS) appartenant au service. Les KMS clés gérées par le client sont également prises en charge et sont requises pour l'importation et l'exportation de fichiers depuis un magasin de données. Pour en savoir plus sur les KMS clés gérées par le client, consultez Amazon Key Management Service. Les clients peuvent choisir une KMS clé AWS détenue ou une KMS clé gérée par le client lors de la création d'un magasin de données. La configuration du chiffrement ne peut pas être modifiée une fois qu'un magasin de données a été créé. Si un magasin de données utilise une KMS clé qu'il AWS possède, elle sera désignée comme telle AWS_OWNED_KMS_KEY et vous ne verrez pas la clé spécifique utilisée pour le chiffrement au repos.

AWSKMSclé possédée

HealthLake utilise ces clés par défaut pour chiffrer automatiquement les informations potentiellement sensibles telles que les données personnelles identifiables ou les données de santé privées (PHI) au repos. AWSKMSles clés que vous détenez ne sont pas stockées dans votre compte. Elles font partie d'un ensemble de KMS clés détenues AWS et gérées pour être utilisées dans plusieurs AWS comptes. AWSles services peuvent utiliser KMS des clés AWS détenues pour protéger vos données. Vous ne pouvez pas afficher, gérer, utiliser AWS les KMS clés que vous possédez ou auditer leur utilisation. Cependant, vous n'avez pas besoin de travailler ou de modifier de programme pour protéger les clés qui chiffrent vos données.

Aucuns frais mensuels ni frais d'utilisation ne vous sont facturés si vous utilisez les KMS clés que vous AWS possédez, et elles ne sont pas prises en compte dans les AWS KMS quotas de votre compte. Pour plus d'informations, consultez la section Clés AWS détenues.

KMSClés gérées par le client

HealthLake prend en charge l'utilisation d'une KMS clé symétrique gérée par le client que vous créez, détenez et gérez pour ajouter une deuxième couche de chiffrement au chiffrement que vous AWS

possédez déjà. Étant donné que vous avez le contrôle total de cette couche de chiffrement, vous pouvez effectuer les tâches suivantes :

- Établir et maintenir des politiques, des IAM politiques et des subventions clés
- Rotation des matériaux de chiffrement de clé
- Activation et désactivation des stratégies de clé
- Ajout de balises
- Création d'alias de clé
- Planification des clés pour la suppression

Vous pouvez également l' CloudTrail utiliser pour suivre les demandes HealthLake envoyées AWS KMS en votre nom. Des AWS KMS frais supplémentaires s'appliquent. Pour plus d'informations, consultez la section sur les clés détenues par le client.

Création d’une clé gérée par le client

Vous pouvez créer une clé symétrique gérée par le client à l'aide de la console AWS de gestion ou du AWS KMS APIs.

Suivez les étapes de création d'une clé symétrique gérée par le client dans le guide du développeur du service de gestion des AWS clés.

Les politiques de clés contrôlent l’accès à votre clé gérée par le client. Chaque clé gérée par le client doit avoir exactement une stratégie de clé, qui contient des instructions qui déterminent les personnes pouvant utiliser la clé et comment elles peuvent l’utiliser. Lorsque vous créez votre clé gérée par le client, vous pouvez spécifier une stratégie de clé. Pour plus d'informations, consultez la section Gestion de l'accès aux clés gérées par le client dans le Guide du développeur du service de gestion des AWS clés.

Pour utiliser votre clé gérée par le client avec vos HealthLake ressources, les CreateGrant opérations kms : doivent être autorisées dans la politique des clés. Cela ajoute une autorisation à une clé gérée par le client qui contrôle l'accès à une KMS clé spécifiée, ce qui donne à l'utilisateur l'accès aux opérations kms:grant requises. HealthLake Voir Utilisation des subventions pour plus d'informations.

Pour utiliser votre KMS clé gérée par le client avec vos HealthLake ressources, les API opérations suivantes doivent être autorisées dans la politique des clés :

- kms : CreateGrant ajoute des subventions à une KMS clé spécifique gérée par le client qui permet d'accéder aux opérations de subvention.
- kms : DescribeKey fournit les informations clés gérées par le client nécessaires à la validation de la clé. Cela est obligatoire pour toutes les opérations.
- kms : GenerateDataKey fournit un accès aux ressources de chiffrement au repos pour toutes les opérations d'écriture.
- KMS:Decrypt permet d'accéder aux opérations de lecture ou de recherche de ressources chiffrées.

Voici un exemple de déclaration de politique qui permet à un utilisateur de créer et d'interagir avec un magasin de AWS HealthLake données chiffré par cette clé :

```
"Statement": [
    {
        "Sid": "Allow access to create data stores and do CRUD/search in AWS
 HealthLake",
        "Effect": "Allow",
        "Principal": {
            "AWS": "arn:aws:iam::111122223333:HealthLakeFullAccessRole"
        },
        "Action": [
            "kms:DescribeKey",
            "kms:CreateGrant",
            "kms:GenerateDataKey",
            "kms:Decrypt"
        ],
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "kms:ViaService": "healthlake.amazonaws.com",
                "kms:CallerAccount": "111122223333"
            }
        }
    }
]
```

IAMAutorisations requises pour utiliser une KMS clé gérée par le client

Lors de la création d'un magasin de données dont le AWS KMS chiffrement est activé à l'aide d'une KMS clé gérée par le client, des autorisations sont requises à la fois pour la politique clé et pour la IAM politique de l'utilisateur ou du rôle qui crée le magasin de HealthLake données.

Vous pouvez utiliser la clé de ViaService condition kms : pour limiter l'utilisation de la KMS clé aux seules demandes provenant de HealthLake.

Pour plus d'informations sur les politiques clés, consultez la section Activation IAM des politiques dans le Guide du développeur du service de gestion des AWS clés.

L'IAMutilisateur, le IAM rôle ou le AWS compte qui crée vos référentiels doit disposer des autorisations kms: CreateGrantGenerateDataKey, kms :, et kms: DescribeKey des HealthLake autorisations nécessaires.

Comment HealthLake utilise les subventions dans AWS KMS

HealthLake nécessite une autorisation pour utiliser votre KMS clé gérée par le client. Lorsque vous créez un magasin de données chiffré à l'aide d'une KMS clé gérée par le client, vous HealthLake créez une subvention en votre nom en envoyant une CreateGrantdemande à AWSKMS. Les subventions AWS KMS sont utilisées pour donner HealthLake accès à une KMS clé dans un compte client.

Les subventions HealthLake créées en votre nom ne doivent pas être révoquées ou retirées. Si vous révoquez ou retirez l' HealthLake autorisation d'utiliser les AWS KMS clés de votre compte, vous HealthLake ne pouvez pas accéder à ces données, chiffrer les nouvelles FHIR ressources envoyées au magasin de données ou les déchiffrer lorsqu'elles sont extraites. Lorsque vous révoquez ou retirez une subvention pour HealthLake, le changement intervient immédiatement. Pour révoquer les droits d'accès, vous devez supprimer le magasin de données plutôt que de révoquer l'autorisation. Lorsqu'un magasin de données est supprimé, les HealthLake subventions sont annulées en votre nom.

Surveillance de vos clés de chiffrement pour HealthLake

Vous pouvez l'utiliser CloudTrail pour suivre les demandes HealthLake envoyées en votre AWS KMS nom lorsque vous utilisez une KMS clé gérée par le client. Les entrées du CloudTrail journal indiquent healthlake.amazonaws.com dans le userAgent champ afin de distinguer clairement les demandes effectuées par. HealthLake

Les exemples suivants sont CloudTrail des événements pour CreateGrant GenerateDataKey, déchiffrer et surveiller les AWS KMS opérations appelées DescribeKey pour accéder HealthLake aux données chiffrées par votre clé gérée par le client.

Ce qui suit montre comment utiliser CreateGrant pour autoriser l'accès HealthLake à une KMS clé fournie par le client, HealthLake permettant d'utiliser cette KMS clé pour chiffrer toutes les données client au repos.

Les utilisateurs ne sont pas tenus de créer leurs propres subventions. HealthLake crée une subvention en votre nom en envoyant une CreateGrant demande à AWSKMS. Les subventions AWS KMS sont utilisées pour donner HealthLake accès à une AWS KMS clé dans un compte client.


```
         {
"eventVersion": "1.08",
"userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLEROLE:Sampleuser01",
    "arn": "arn:aws:sts::111122223333:assumed-role/Sampleuser01,
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLEKEYID",
    "sessionContext": {
        "sessionIssuer": {
            "type": "Role",
            "principalId": "EXAMPLEROLE",
            "arn": "arn:aws:iam::111122223333:role/Sampleuser01",
            "accountId": "111122223333",
            "userName": "Sampleuser01"
        },
        "webIdFederationData": {},
        "attributes": {
            "creationDate": "2021-06-30T19:33:37Z",
            "mfaAuthenticated": "false"
        }
    },
    "invokedBy": "healthlake.amazonaws.com"
},
"eventTime": "2021-06-30T20:31:15Z",
"eventSource": "kms.amazonaws.com",
"eventName": "CreateGrant",
"awsRegion": "us-east-1",
"sourceIPAddress": "healthlake.amazonaws.com",
"userAgent": "healthlake.amazonaws.com",
```

```
    "requestParameters": {
        "operations": [
            "CreateGrant",
            "Decrypt",
            "DescribeKey",
            "Encrypt",
            "GenerateDataKey",
            "GenerateDataKeyWithoutPlaintext",
            "ReEncryptFrom",
            "ReEncryptTo",
            "RetireGrant"
        ],
        "granteePrincipal": "healthlake.us-east-1.amazonaws.com",
        "keyId": "arn:aws:kms:us-east-1:111122223333:key/EXAMPLE_KEY_ARN",
        "retiringPrincipal": "healthlake.us-east-1.amazonaws.com"
    },
    "responseElements": {
        "grantId": "EXAMPLE_ID_01"
    },
    "requestID": "EXAMPLE_ID_02",
    "eventID": "EXAMPLE_ID_03",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-east-1:111122223333:key/EXAMPLE_KEY_ARN"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

Les exemples suivants montrent comment s' GenerateDataKey assurer que l'utilisateur dispose des autorisations nécessaires pour chiffrer les données avant de les stocker.

```
            {
    "eventVersion": "1.08",
    "userIdentity": {
```

```
        "type": "AssumedRole",
        "principalId": "EXAMPLEUSER",
        "arn": "arn:aws:sts::111122223333:assumed-role/Sampleuser01",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLEKEYID",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "EXAMPLEROLE",
                "arn": "arn:aws:iam::111122223333:role/Sampleuser01",
                "accountId": "111122223333",
                "userName": "Sampleuser01"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2021-06-30T21:17:06Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "healthlake.amazonaws.com"
    },
    "eventTime": "2021-06-30T21:17:37Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "healthlake.amazonaws.com",
    "userAgent": "healthlake.amazonaws.com",
    "requestParameters": {
        "keySpec": "AES_256",
        "keyId": "arn:aws:kms:us-east-1:111122223333:key/EXAMPLE_KEY_ARN"
    },
    "responseElements": null,
    "requestID": "EXAMPLE_ID_01",
    "eventID": "EXAMPLE_ID_02",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-east-1:111122223333:key/EXAMPLE_KEY_ARN"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
```

```
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

L'exemple suivant montre comment HealthLake appelle l'opération Decrypt pour utiliser la clé de données cryptée stockée afin d'accéder aux données cryptées.

```
         {
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "EXAMPLEUSER",
        "arn": "arn:aws:sts::111122223333:assumed-role/Sampleuser01",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLEKEYID",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "EXAMPLEROLE",
                "arn": "arn:aws:iam::111122223333:role/Sampleuser01",
                "accountId": "111122223333",
                "userName": "Sampleuser01"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2021-06-30T21:17:06Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "healthlake.amazonaws.com"
    },
    "eventTime": "2021-06-30T21:21:59Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "healthlake.amazonaws.com",
    "userAgent": "healthlake.amazonaws.com",
    "requestParameters": {
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "keyId": "arn:aws:kms:us-east-1:111122223333:key/EXAMPLE_KEY_ARN"
    },
```

```
    "responseElements": null,
    "requestID": "EXAMPLE_ID_01",
    "eventID": "EXAMPLE_ID_02",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-east-1:111122223333:key/EXAMPLE_KEY_ARN"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

L'exemple suivant montre comment HealthLake utiliser l' DescribeKey opération pour vérifier si la AWS KMS clé détenue par le AWS KMS client est dans un état utilisable et pour aider l'utilisateur à résoudre les problèmes si elle n'est pas fonctionnelle.

```
         {
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "EXAMPLEUSER",
        "arn": "arn:aws:sts::111122223333:assumed-role/Sampleuser01",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLEKEYID",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "EXAMPLEROLE",
                "arn": "arn:aws:iam::111122223333:role/Sampleuser01",
                "accountId": "111122223333",
                "userName": "Sampleuser01"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2021-07-01T18:36:14Z",
                "mfaAuthenticated": "false"
```

```
            }
        },
        "invokedBy": "healthlake.amazonaws.com"
    },
    "eventTime": "2021-07-01T18:36:36Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DescribeKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "healthlake.amazonaws.com",
    "userAgent": "healthlake.amazonaws.com",
    "requestParameters": {
        "keyId": "arn:aws:kms:us-east-1:111122223333:key/EXAMPLE_KEY_ARN"
    },
    "responseElements": null,
    "requestID": "EXAMPLE_ID_01",
    "eventID": "EXAMPLE_ID_02",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-east-1:111122223333:key/EXAMPLE_KEY_ARN"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```


En savoir plus

Les ressources suivantes fournissent des informations supplémentaires sur le chiffrement des données au repos.

Pour plus d'informations sur les concepts de base du service de gestion des AWS clés, consultez la AWS KMS documentation.

Pour plus d'informations sur les meilleures pratiques en matière de sécurité, AWS KMS consultez la documentation.

Chiffrement en transit pour AWS HealthLake

AWS HealthLake utilise la TLS version 1.2 pour chiffrer les données en transit via le point de terminaison public et via les services principaux.

Gestion des identités et des accès pour AWS HealthLake

AWS Identity and Access Management (IAM) est un outil Service AWS qui permet à un administrateur de contrôler en toute sécurité l'accès aux AWS ressources. IAMles administrateurs contrôlent qui peut être authentifié (connecté) et autorisé (autorisé) à utiliser les HealthLake ressources. IAMest un Service AWS outil que vous pouvez utiliser sans frais supplémentaires.

Rubriques

Public ciblé

La façon dont vous utilisez AWS Identity and Access Management (IAM) varie en fonction du travail que vous effectuez HealthLake.

Utilisateur du service : si vous utilisez le HealthLake service pour effectuer votre travail, votre administrateur vous fournit les informations d'identification et les autorisations dont vous avez besoin. Au fur et à mesure que vous utilisez de nouvelles HealthLake fonctionnalités pour effectuer votre travail, vous aurez peut-être besoin d'autorisations supplémentaires. En comprenant bien la gestion des accès, vous saurez demander les autorisations appropriées à votre administrateur. Si vous ne pouvez pas accéder à une fonctionnalité dans HealthLake, consultez Résolution des problèmes AWS HealthLake d'identité et d'accès.

Administrateur du service — Si vous êtes responsable des HealthLake ressources de votre entreprise, vous avez probablement un accès complet à HealthLake. C'est à vous de déterminer

les HealthLake fonctionnalités et les ressources auxquelles les utilisateurs de votre service doivent accéder. Vous devez ensuite soumettre les demandes à votre administrateur IAM pour modifier les autorisations des utilisateurs de votre service. Consultez les informations sur cette page pour comprendre les concepts de base d'IAM. Pour en savoir plus sur la façon dont votre entreprise peut utiliser IAM avec HealthLake, voirComment AWS HealthLake fonctionne avec IAM.

IAMadministrateur — Si vous êtes IAM administrateur, vous souhaiterez peut-être en savoir plus sur la manière dont vous pouvez rédiger des politiques pour gérer l'accès à HealthLake. Pour consulter des exemples de politiques HealthLake basées sur l'identité que vous pouvez utiliser dansIAM, consultez. Exemples de politiques basées sur l'identité pour AWS HealthLake

Authentification par des identités

L'authentification est la façon dont vous vous connectez à AWS l'aide de vos informations d'identification. Vous devez être authentifié (connecté à AWS) en tant que Utilisateur racine d'un compte AWS, en tant qu'IAMutilisateur ou en assumant un IAM rôle.

Vous pouvez vous connecter en AWS tant qu'identité fédérée en utilisant les informations d'identification fournies par le biais d'une source d'identité. AWS IAM Identity Center Les utilisateurs (IAMIdentity Center), l'authentification unique de votre entreprise et vos informations d'identification Google ou Facebook sont des exemples d'identités fédérées. Lorsque vous vous connectez en tant qu'identité fédérée, votre administrateur aura précédemment configuré une fédération d'identités avec des rôles IAM. Lorsque vous accédez à AWS l'aide de la fédération, vous assumez indirectement un rôle.

Selon le type d'utilisateur que vous êtes, vous pouvez vous connecter au portail AWS Management Console ou au portail AWS d'accès. Pour plus d'informations sur la connexion à AWS, consultez la section Comment vous connecter à votre compte Compte AWS dans le guide de Connexion à AWS l'utilisateur.

Si vous y accédez AWS par programmation, AWS fournit un kit de développement logiciel (SDK) et une interface de ligne de commande (CLI) pour signer cryptographiquement vos demandes à l'aide de vos informations d'identification. Si vous n'utilisez pas d' AWS outils, vous devez signer vous-même les demandes. Pour plus d'informations sur l'utilisation de la méthode recommandée pour signer vous-même les demandes, consultez la version 4 de AWS Signature pour les API demandes dans le guide de IAM l'utilisateur.

Quelle que soit la méthode d'authentification que vous utilisez, vous devrez peut-être fournir des informations de sécurité supplémentaires. Par exemple, il vous AWS recommande d'utiliser

l'authentification multifactorielle (MFA) pour renforcer la sécurité de votre compte. Pour en savoir plus, voir Authentification multifactorielle dans le guide de l'AWS IAM Identity Center utilisateur et Authentification AWS multifactorielle IAM dans le guide de l'IAMutilisateur.

Compte AWS utilisateur root

Lorsque vous créez un Compte AWS, vous commencez par une identité de connexion unique qui donne un accès complet à toutes Services AWS les ressources du compte. Cette identité est appelée utilisateur Compte AWS root et est accessible en vous connectant avec l'adresse e-mail et le mot de passe que vous avez utilisés pour créer le compte. Il est vivement recommandé de ne pas utiliser l'utilisateur racine pour vos tâches quotidiennes. Protégez vos informations d'identification d'utilisateur racine et utilisez-les pour effectuer les tâches que seul l'utilisateur racine peut effectuer. Pour obtenir la liste complète des tâches qui nécessitent que vous vous connectiez en tant qu'utilisateur root, consultez la section Tâches nécessitant des informations d'identification utilisateur root dans le Guide de IAM l'utilisateur.

Identité fédérée

La meilleure pratique consiste à obliger les utilisateurs humains, y compris ceux qui ont besoin d'un accès administrateur, à utiliser la fédération avec un fournisseur d'identité pour accéder à l'aide Services AWS d'informations d'identification temporaires.

Une identité fédérée est un utilisateur de l'annuaire des utilisateurs de votre entreprise, d'un fournisseur d'identité Web AWS Directory Service, du répertoire Identity Center ou de tout utilisateur qui y accède à l'aide des informations d'identification fournies Services AWS par le biais d'une source d'identité. Lorsque des identités fédérées y accèdent Comptes AWS, elles assument des rôles, qui fournissent des informations d'identification temporaires.

Pour une gestion des accès centralisée, nous vous recommandons d'utiliser AWS IAM Identity Center. Vous pouvez créer des utilisateurs et des groupes dans IAM Identity Center, ou vous pouvez vous connecter et synchroniser avec un ensemble d'utilisateurs et de groupes dans votre propre source d'identité afin de les utiliser dans toutes vos applications Comptes AWS et applications. Pour plus d'informations sur IAM Identity Center, consultez Qu'est-ce qu'IAMIdentity Center ? dans le guide de AWS IAM Identity Center l'utilisateur.

Utilisateurs et groupes IAM

Un IAMutilisateur est une identité au sein de vous Compte AWS qui possède des autorisations spécifiques pour une seule personne ou une seule application. Dans la mesure du possible, nous vous recommandons de vous appuyer sur des informations d'identification temporaires plutôt que de

créer des IAM utilisateurs dotés d'informations d'identification à long terme, telles que des mots de passe et des clés d'accès. Toutefois, si vous avez des cas d'utilisation spécifiques qui nécessitent des informations d'identification à long terme auprès des IAM utilisateurs, nous vous recommandons de faire pivoter les clés d'accès. Pour plus d'informations, voir Rotation régulière des clés d'accès pour les cas d'utilisation nécessitant des informations d'identification à long terme dans le Guide de IAM l'utilisateur.

Un groupe IAM est une identité qui spécifie un ensemble d'utilisateurs IAM. Vous ne pouvez pas vous connecter en tant que groupe. Vous pouvez utiliser les groupes pour spécifier des autorisations pour plusieurs utilisateurs à la fois. Les groupes permettent de gérer plus facilement les autorisations pour de grands ensembles d'utilisateurs. Par exemple, vous pouvez nommer un groupe IAMAdminset lui donner les autorisations nécessaires pour administrer IAM des ressources.

Les utilisateurs sont différents des rôles. Un utilisateur est associé de manière unique à une personne ou une application, alors qu'un rôle est conçu pour être endossé par tout utilisateur qui en a besoin. Les utilisateurs disposent d'informations d'identification permanentes, mais les rôles fournissent des informations d'identification temporaires. Pour en savoir plus, consultez la section Cas d'utilisation pour IAM les utilisateurs dans le Guide de IAM l'utilisateur.

Rôles IAM

Un IAMrôle est une identité au sein de Compte AWS vous dotée d'autorisations spécifiques. Le concept ressemble à celui d'utilisateur IAM, mais un rôle n'est pas associé à une personne en particulier. Pour assumer temporairement un IAM rôle dans le AWS Management Console, vous pouvez passer d'un rôle d'utilisateur à un IAM rôle (console). Vous pouvez assumer un rôle en appelant une AWS API opération AWS CLI or ou en utilisant une option personnaliséeURL. Pour plus d'informations sur les méthodes d'utilisation des rôles, consultez la section Méthodes pour assumer un rôle dans le Guide de IAM l'utilisateur.

Les rôles IAM avec des informations d'identification temporaires sont utiles dans les cas suivants :

- Accès utilisateur fédéré : pour attribuer des autorisations à une identité fédérée, vous créez un rôle et définissez des autorisations pour le rôle. Quand une identité externe s'authentifie, l'identité est associée au rôle et reçoit les autorisations qui sont définies par celui-ci. Pour plus d'informations sur les rôles pour la fédération, voir Créer un rôle pour un fournisseur d'identité tiers (fédération) dans le guide de IAM l'utilisateur. Si vous utilisez IAM Identity Center, vous configurez un ensemble d'autorisations. Pour contrôler les accès auxquels vos identités peuvent accéder après leur authentification, IAM Identity Center met en corrélation l'ensemble d'autorisations avec un rôle

dans. IAM Pour plus d'informations sur les jeux d'autorisations, consultez Jeux d'autorisations dans le Guide de l'utilisateur AWS IAM Identity Center .

- Autorisations IAM utilisateur temporaires : un IAM utilisateur ou un rôle peut assumer un IAM rôle afin d'obtenir temporairement différentes autorisations pour une tâche spécifique.
- Accès intercompte : vous pouvez utiliser un rôle IAM pour permettre à un utilisateur (un principal de confiance) d'un compte différent d'accéder aux ressources de votre compte. Les rôles constituent le principal moyen d'accorder l'accès intercompte. Toutefois, dans certains Services AWS cas, vous pouvez associer une politique directement à une ressource (au lieu d'utiliser un rôle comme proxy). Pour connaître la différence entre les rôles et les politiques basées sur les ressources pour l'accès entre comptes, voir Accès aux ressources entre comptes IAM dans le guide de l'IAMutilisateur.
- Accès multiservices — Certains Services AWS utilisent des fonctionnalités dans d'autres Services AWS. Par exemple, lorsque vous effectuez un appel dans un service, il est courant que ce service exécute des applications dans Amazon EC2 ou stocke des objets dans Amazon S3. Un service peut le faire en utilisant les autorisations d'appel du principal, un rôle de service ou un rôle lié au service.
  - Sessions d'accès transmises (FAS) — Lorsque vous utilisez un IAM utilisateur ou un rôle pour effectuer des actions AWS, vous êtes considéré comme un mandant. Lorsque vous utilisez certains services, vous pouvez effectuer une action qui déclenche ensuite une autre action dans un autre service. FASutilise les autorisations du principal appelant an Service AWS, combinées à la demande Service AWS pour adresser des demandes aux services en aval. FASles demandes ne sont effectuées que lorsqu'un service reçoit une demande qui nécessite des interactions avec d'autres personnes Services AWS ou des ressources pour être traitée. Dans ce cas, vous devez disposer d'autorisations nécessaires pour effectuer les deux actions. Pour plus de détails sur les politiques relatives FAS aux demandes, consultez la section Transférer les sessions d'accès.
  - Rôle de service — Un rôle de service est un IAMrôle qu'un service assume pour effectuer des actions en votre nom. Un administrateur IAM peut créer, modifier et supprimer une fonction du service à partir de IAM. Pour plus d'informations, consultez la section Créer un rôle pour déléguer des autorisations à un Service AWS dans le guide de IAM l'utilisateur.
  - Rôle lié à un service — Un rôle lié à un service est un type de rôle de service lié à un. Service AWS Le service peut endosser le rôle afin d'effectuer une action en votre nom. Les rôles liés à un service apparaissent dans votre Compte AWS répertoire et appartiennent au service. Un administrateur IAM peut consulter, mais ne peut pas modifier les autorisations concernant les rôles liés à un service.

- Applications exécutées sur Amazon EC2 : vous pouvez utiliser un IAM rôle pour gérer les informations d'identification temporaires pour les applications qui s'exécutent sur une EC2 instance et qui font AWS CLI des AWS API demandes. Cette solution est préférable au stockage des clés d'accès au sein de l'instance EC2. Pour attribuer un AWS rôle à une EC2 instance et le rendre disponible pour toutes ses applications, vous devez créer un profil d'instance attaché à l'instance. Un profil d'instance contient le rôle et permet aux programmes qui s'exécutent sur l'instance EC2 d'obtenir des informations d'identification temporaires. Pour plus d'informations, consultez Utiliser un IAM rôle pour accorder des autorisations aux applications exécutées sur des EC2 instances Amazon dans le Guide de IAM l'utilisateur.

Gestion des accès à l’aide de politiques

Vous contrôlez l'accès en AWS créant des politiques et en les associant à AWS des identités ou à des ressources. Une politique est un objet AWS qui, lorsqu'il est associé à une identité ou à une ressource, définit leurs autorisations. AWS évalue ces politiques lorsqu'un principal (utilisateur, utilisateur root ou session de rôle) fait une demande. Les autorisations dans les politiques déterminent si la demande est autorisée ou refusée. La plupart des politiques sont stockées AWS sous forme de JSON documents. Pour plus d'informations sur la structure et le contenu des documents de JSON politique, voir Présentation des JSON politiques dans le guide de IAM l'utilisateur.

Les administrateurs peuvent utiliser AWS JSON des politiques pour spécifier qui a accès à quoi. C’est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

Par défaut, les utilisateurs et les rôles ne disposent d’aucune autorisation. Pour autoriser les utilisateurs à effectuer des actions sur les ressources dont ils ont besoin, un IAM administrateur peut créer des IAM politiques. L'administrateur peut ensuite ajouter les IAM politiques aux rôles, et les utilisateurs peuvent assumer les rôles.

Les stratégies IAM définissent les autorisations d'une action quelle que soit la méthode que vous utilisez pour exécuter l'opération. Par exemple, supposons que vous disposiez d’une politique qui autorise l’action `iam:GetRole`. Un utilisateur appliquant cette politique peut obtenir des informations sur le rôle auprès du AWS Management Console AWS CLI, ou du AWS API.

Politiques basées sur l’identité

Les politiques basées sur l'identité sont JSON des documents de politique d'autorisation que vous pouvez joindre à une identité, telle qu'un IAM utilisateur, un groupe d'utilisateurs ou un rôle. Ces politiques contrôlent quel type d’actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour savoir comment créer une politique basée sur l'identité, voir Définir des IAM autorisations personnalisées avec des politiques gérées par le client dans le Guide de l'IAMutilisateur.

Les politiques basées sur l’identité peuvent être classées comme des politiques en ligne ou des politiques gérées. Les politiques en ligne sont intégrées directement à un utilisateur, groupe ou rôle. Les politiques gérées sont des politiques autonomes que vous pouvez associer à plusieurs utilisateurs, groupes et rôles au sein de votre Compte AWS. Les politiques gérées incluent les politiques AWS gérées et les politiques gérées par le client. Pour savoir comment choisir entre une politique gérée ou une politique intégrée, voir Choisir entre les politiques gérées et les politiques intégrées dans le Guide de l'IAMutilisateur.

Politiques basées sur les ressources

Les politiques basées sur les ressources sont des documents JSON de stratégie que vous attachez à une ressource. Les politiques de confiance dans les IAM rôles et les politiques relatives aux compartiments Amazon S3 sont des exemples de politiques basées sur les ressources. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l’accès à une ressource spécifique. Pour la ressource dans laquelle se trouve la politique, cette dernière définit quel type d’actions un principal spécifié peut effectuer sur cette ressource et dans quelles conditions. Vous devez spécifier un principal dans une politique basée sur les ressources. Les principaux peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ou. Services AWS

Les politiques basées sur les ressources sont des politiques en ligne situées dans ce service. Vous ne pouvez pas utiliser de politiques AWS gérées depuis une IAM stratégie basée sur les ressources.

Listes de contrôle d'accès (ACLs)

Les listes de contrôle d'accès (ACLs) contrôlent les principaux (membres du compte, utilisateurs ou rôles) autorisés à accéder à une ressource. ACLssont similaires aux politiques basées sur les ressources, bien qu'elles n'utilisent pas le format du document JSON de stratégie.

Amazon S3 et Amazon VPC sont des exemples de services compatiblesACLs. AWS WAF Pour en savoir plusACLs, consultez la présentation de la liste de contrôle d'accès (ACL) dans le guide du développeur Amazon Simple Storage Service.

Autres types de politique

AWS prend en charge d'autres types de politiques moins courants. Ces types de politiques peuvent définir le nombre maximum d'autorisations qui vous sont accordées par des types de politiques plus courants.

- Limites d'autorisations — Une limite d'autorisations est une fonctionnalité avancée dans laquelle vous définissez le maximum d'autorisations qu'une politique basée sur l'identité peut accorder à une IAM entité (IAMutilisateur ou rôle). Vous pouvez définir une limite d'autorisations pour une entité. Les autorisations en résultant représentent la combinaison des politiques basées sur l'identité d'une entité et de ses limites d'autorisation. Les politiques basées sur les ressources qui spécifient l'utilisateur ou le rôle dans le champ `Principal` ne sont pas limitées par les limites d'autorisations. Un refus explicite dans l'une de ces politiques annule l'autorisation. Pour plus d'informations sur les limites d'autorisations, consultez la section Limites d'autorisations pour les IAM entités dans le Guide de IAM l'utilisateur.
- Politiques de contrôle des services (SCPs) : SCPs JSON politiques qui spécifient les autorisations maximales pour une organisation ou une unité organisationnelle (UO) dans AWS Organizations. AWS Organizations est un service permettant de regrouper et de gérer de manière centralisée Comptes AWS les multiples propriétés de votre entreprise. Si vous activez toutes les fonctionnalités d'une organisation, vous pouvez appliquer des politiques de contrôle des services (SCPs) à l'un ou à l'ensemble de vos comptes. Les SCP limites d'autorisations pour les entités présentes dans les comptes des membres, y compris chacune d'entre elles Utilisateur racine d'un compte AWS. Pour plus d'informations sur les Organizations et consultez SCPs les politiques de contrôle des services dans le Guide de AWS Organizations l'utilisateur.
- Politiques de contrôle des ressources (RCPs) : RCPs JSON politiques que vous pouvez utiliser pour définir le maximum d'autorisations disponibles pour les ressources de vos comptes sans mettre à jour les IAM politiques associées à chaque ressource que vous possédez. Cela RCP limite les autorisations pour les ressources dans les comptes des membres et peut avoir un impact sur les autorisations effectives pour les identités Utilisateur racine d'un compte AWS, y compris, qu'elles appartiennent ou non à votre organisation. Pour plus d'informations sur les OrganizationsRCPs, y compris une liste de ces Services AWS supportsRCPs, consultez la section Resource control policies (RCPs) dans le guide de AWS Organizations l'utilisateur.

- Politiques de séance : les politiques de séance sont des politiques avancées que vous utilisez en tant que paramètre lorsque vous créez par programmation une séance temporaire pour un rôle ou un utilisateur fédéré. Les autorisations de séance en résultant sont une combinaison des politiques basées sur l'identité de l'utilisateur ou du rôle et des politiques de séance. Les autorisations peuvent également provenir d'une politique basée sur les ressources. Un refus explicite dans l'une de ces politiques annule l'autorisation. Pour en savoir plus, consultez Politiques de session dans le Guide de l'utilisateur IAM.

Plusieurs types de politique

Lorsque plusieurs types de politiques s'appliquent à la requête, les autorisations en résultant sont plus compliquées à comprendre. Pour savoir comment AWS déterminer s'il faut autoriser une demande lorsque plusieurs types de politiques sont impliqués, consultez la section Logique d'évaluation des politiques dans le guide de IAM l'utilisateur.

Comment AWS HealthLake fonctionne avec IAM

Avant IAM de gérer l'accès à HealthLake, découvrez quelles IAM fonctionnalités sont disponibles HealthLake.

IAMfonctionnalités que vous pouvez utiliser avec AWS HealthLake

Fonctionnalité IAM	HealthLake soutien
Politiques basées sur l'identité	Oui
Politiques basées sur les ressources	Non
Actions de politique	Oui
Ressources de politique	Oui
Clés de condition de politique	Oui
ACLs	Non
ABAC(balises dans les politiques)	Oui
Informations d'identification temporaires	Oui

Fonctionnalité IAM	HealthLake soutien
Autorisations de principal	Oui
Rôles de service	Oui
Rôles liés à un service	Non

Pour obtenir une vue d'ensemble du fonctionnement de la plupart des IAM fonctionnalités HealthLake et des autres AWS services, reportez-vous à la section AWS Services compatibles IAM dans le Guide de IAM l'utilisateur.

Politiques basées sur l'identité pour AWS HealthLake

Prend en charge les politiques basées sur l’identité : oui

Les politiques basées sur l'identité sont JSON des documents de politique d'autorisation que vous pouvez joindre à une identité, telle qu'un IAM utilisateur, un groupe d'utilisateurs ou un rôle. Ces politiques contrôlent quel type d’actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour savoir comment créer une politique basée sur l'identité, voir Définir des IAM autorisations personnalisées avec des politiques gérées par le client dans le Guide de l'IAMutilisateur.

Avec les stratégies IAM basées sur l’identité, vous pouvez spécifier des actions et ressources autorisées ou refusées, ainsi que les conditions dans lesquelles les actions sont autorisées ou refusées. Vous ne pouvez pas spécifier le principal dans une politique basée sur une identité, car celle-ci s’applique à l’utilisateur ou au rôle auquel elle est attachée. Pour en savoir plus sur tous les éléments que vous pouvez utiliser dans une JSON politique, consultez la référence aux éléments de IAM JSON politique dans le Guide de IAM l'utilisateur.

Exemples de politiques basées sur l'identité pour AWS HealthLake

Pour consulter des exemples de politiques HealthLake basées sur l'identité, consultez. Exemples de politiques basées sur l'identité pour AWS HealthLake

Politiques basées sur les ressources au sein de AWS HealthLake

Prend en charge les politiques basées sur les ressources : non

Les politiques basées sur les ressources sont des documents JSON de stratégie que vous attachez à une ressource. Les politiques de confiance dans les IAM rôles et les politiques relatives aux compartiments Amazon S3 sont des exemples de politiques basées sur les ressources. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Pour la ressource dans laquelle se trouve la politique, cette dernière définit quel type d'actions un principal spécifié peut effectuer sur cette ressource et dans quelles conditions. Vous devez spécifier un principal dans une politique basée sur les ressources. Les principaux peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ou. Services AWS

Pour permettre un accès entre comptes, vous pouvez spécifier un compte entier ou des entités IAM dans un autre compte en tant que mandataire dans une stratégie basée sur les ressources. L'ajout d'un principal entre comptes à une politique basée sur les ressources ne représente qu'une partie de l'instauration de la relation d'approbation. Lorsque le principal et la ressource sont différents Comptes AWS, un IAM administrateur du compte de confiance doit également accorder à l'entité principale (utilisateur ou rôle) l'autorisation d'accéder à la ressource. Pour ce faire, il attache une politique basée sur une identité à l'entité. Toutefois, si une politique basée sur des ressources accorde l'accès à un principal dans le même compte, aucune autre politique basée sur l'identité n'est requise. Pour plus d'informations, voir Accès aux ressources entre comptes IAM dans le Guide de IAM l'utilisateur.

Actions politiques pour AWS HealthLake

Prend en charge les actions de politique : oui

Les administrateurs peuvent utiliser AWS JSON des politiques pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'`Action`élément d'une JSON politique décrit les actions que vous pouvez utiliser pour autoriser ou refuser l'accès dans une politique. Les actions de stratégie portent généralement le même nom que l' AWS APIopération associée. Il existe certaines exceptions, telles que les actions avec autorisation uniquement qui n'ont pas d'opération correspondante. API Certaines opérations nécessitent également plusieurs actions dans une politique. Ces actions supplémentaires sont nommées actions dépendantes.

Intégration d'actions dans une politique afin d'accorder l'autorisation d'exécuter les opérations associées.

Pour consulter la liste des HealthLake actions, reportez-vous à la section Actions définies par AWS HealthLake dans la référence d'autorisation de service.

Les actions de politique en HealthLake cours utilisent le préfixe suivant avant l'action :

```
healthlake
```

Pour spécifier plusieurs actions dans une seule instruction, séparez-les par une virgule.

```
"Action": [
      "healthlake:action1",
      "healthlake:action2"
         ]
```

Pour consulter des exemples de politiques HealthLake basées sur l'identité, consultez. Exemples de politiques basées sur l'identité pour AWS HealthLake

Ressources politiques pour AWS HealthLake

Prend en charge les ressources de politique : oui

Les administrateurs peuvent utiliser AWS JSON des politiques pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Resource` JSON de stratégie indique le ou les objets auxquels s'applique l'action. Les instructions doivent inclure un élément `Resource` ou `NotResource`. Il est recommandé de spécifier une ressource en utilisant son Amazon Resource Name (ARN). Vous pouvez le faire pour des actions qui prennent en charge un type de ressource spécifique, connu sous la dénomination autorisations de niveau ressource.

Pour les actions qui ne sont pas compatibles avec les autorisations de niveau ressource, telles que les opérations de liste, utilisez un caractère générique (*) afin d'indiquer que l'instruction s'applique à toutes les ressources.

```
"Resource": "*"
```

Pour consulter la liste des types de HealthLake ressources et leurs caractéristiquesARNs, voir Ressources définies par AWS HealthLake dans la référence d'autorisation de service. Pour connaître

les actions permettant ARN de définir chaque ressource, consultez la section Actions définies par AWS HealthLake.

Pour consulter des exemples de politiques HealthLake basées sur l'identité, consultez. Exemples de politiques basées sur l'identité pour AWS HealthLake

Clés de conditions de politique pour AWS HealthLake

Prend en charge les clés de condition de politique spécifiques au service : oui

Les administrateurs peuvent utiliser AWS JSON des politiques pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Condition` (ou le bloc `Condition`) vous permet de spécifier des conditions lorsqu'une instruction est appliquée. L'élément `Condition` est facultatif. Vous pouvez créer des expressions conditionnelles qui utilisent des opérateurs de condition, tels que les signes égal ou inférieur à, pour faire correspondre la condition de la politique aux valeurs de la demande.

Si vous spécifiez plusieurs éléments `Condition` dans une instruction, ou plusieurs clés dans un seul élément `Condition`, AWS les évalue à l'aide d'une opération `AND` logique. Si vous spécifiez plusieurs valeurs pour une seule clé de condition, AWS évalue la condition à l'aide d'une `OR` opération logique. Toutes les conditions doivent être remplies avant que les autorisations associées à l'instruction ne soient accordées.

Vous pouvez aussi utiliser des variables d'espace réservé quand vous spécifiez des conditions. Par exemple, vous pouvez accorder à un utilisateur IAM l'autorisation d'accéder à une ressource uniquement si elle est balisée avec son nom d'utilisateur IAM . Pour plus d'informations, consultez IAMla section Éléments de politique : variables et balises dans le Guide de IAM l'utilisateur.

AWS prend en charge les clés de condition globales et les clés de condition spécifiques au service. Pour voir toutes les clés de condition AWS globales, voir les clés contextuelles de condition AWS globales dans le guide de IAM l'utilisateur.

Pour consulter la liste des clés de HealthLake condition, reportez-vous à la section Clés de condition pour AWS HealthLake la référence d'autorisation de service. Pour connaître les actions et les ressources avec lesquelles vous pouvez utiliser une clé de condition, consultez la section Actions définies par AWS HealthLake.

Pour consulter des exemples de politiques HealthLake basées sur l'identité, consultez. Exemples de politiques basées sur l'identité pour AWS HealthLake

Listes de contrôle d'accès (ACLs) dans AWS HealthLake

Supports ACLs : Non

Les listes de contrôle d'accès (ACLs) contrôlent les principaux (membres du compte, utilisateurs ou rôles) autorisés à accéder à une ressource. ACLssont similaires aux politiques basées sur les ressources, bien qu'elles n'utilisent pas le format du document JSON de stratégie.

Contrôle d'accès basé sur les attributs () ABAC avec AWS HealthLake

Supports ABAC (balises dans les politiques) : Oui

Le contrôle d'accès basé sur les attributs (ABAC) est une stratégie d'autorisation qui définit les autorisations en fonction des attributs. Dans AWS, ces attributs sont appelés balises. Vous pouvez associer des balises à IAM des entités (utilisateurs ou rôles) et à de nombreuses AWS ressources. Le balisage des entités et des ressources est la première étape deABAC. Vous concevez ensuite des ABAC politiques pour autoriser les opérations lorsque le tag du principal correspond à celui de la ressource à laquelle il essaie d'accéder.

ABACest utile dans les environnements qui se développent rapidement et aide dans les situations où la gestion des politiques devient fastidieuse.

Pour contrôler l'accès basé sur des étiquettes, vous devez fournir les informations d'étiquette dans l'élément de condition d'une politique utilisant les clés de condition `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou `aws:TagKeys`.

Si un service prend en charge les trois clés de condition pour tous les types de ressources, alors la valeur pour ce service est Oui. Si un service prend en charge les trois clés de condition pour certains types de ressources uniquement, la valeur est Partielle.

Pour plus d'informationsABAC, voir Définir des autorisations avec ABAC autorisation dans le Guide de IAM l'utilisateur. Pour consulter un didacticiel présentant les étapes de configurationABAC, voir Utiliser le contrôle d'accès basé sur les attributs (ABAC) dans le guide de l'IAMutilisateur.

Utilisation d'informations d'identification temporaires avec AWS HealthLake

Prend en charge les informations d'identification temporaires : oui

Certains Services AWS ne fonctionnent pas lorsque vous vous connectez à l'aide d'informations d'identification temporaires. Pour plus d'informations, y compris celles qui Services AWS fonctionnent avec des informations d'identification temporaires, consultez Services AWS la section relative à l'utilisation IAM dans le Guide de IAM l'utilisateur.

Vous utilisez des informations d'identification temporaires si vous vous connectez à l' AWS Management Console aide d'une méthode autre qu'un nom d'utilisateur et un mot de passe. Par exemple, lorsque vous accédez à AWS l'aide du lien d'authentification unique (SSO) de votre entreprise, ce processus crée automatiquement des informations d'identification temporaires. Vous créez également automatiquement des informations d'identification temporaires lorsque vous vous connectez à la console en tant qu'utilisateur, puis changez de rôle. Pour plus d'informations sur le changement de rôle, voir Passer d'un utilisateur à un IAM rôle (console) dans le guide de IAM l'utilisateur.

Vous pouvez créer manuellement des informations d'identification temporaires à l'aide du AWS CLI ou AWS API. Vous pouvez ensuite utiliser ces informations d'identification temporaires pour y accéder AWS. AWS recommande de générer dynamiquement des informations d'identification temporaires au lieu d'utiliser des clés d'accès à long terme. Pour plus d'informations, consultez la section Informations d'identification de sécurité temporaires dans IAM.

Autorisations principales interservices pour AWS HealthLake

Prend en charge les sessions d'accès transféré (FAS) : Oui

Lorsque vous utilisez un IAM utilisateur ou un rôle pour effectuer des actions AWS, vous êtes considéré comme un mandant. Lorsque vous utilisez certains services, vous pouvez effectuer une action qui déclenche ensuite une autre action dans un autre service. FASutilise les autorisations du principal appelant an Service AWS, combinées à la demande Service AWS pour adresser des demandes aux services en aval. FASles demandes ne sont effectuées que lorsqu'un service reçoit une demande qui nécessite des interactions avec d'autres personnes Services AWS ou des ressources pour être traitée. Dans ce cas, vous devez disposer d'autorisations nécessaires pour effectuer les deux actions. Pour plus de détails sur les politiques relatives FAS aux demandes, consultez la section Transférer les sessions d'accès.

Rôles de service pour AWS HealthLake

Prend en charge les rôles de service : oui

Un rôle de service est un IAMrôle qu'un service assume pour effectuer des actions en votre nom. Un administrateur IAM peut créer, modifier et supprimer une fonction du service à partir de IAM. Pour

plus d'informations, consultez la section Créer un rôle pour déléguer des autorisations à un Service AWS dans le guide de IAM l'utilisateur.

Pour plus d'informations sur les rôles de service et la politique en ligne requise pour un accès complet à AWS HealthLake, consultezConfiguration des autorisations pour commencer à utiliser AWS HealthLake.

> **Warning**
>
> La modification des autorisations associées à un rôle de service peut perturber HealthLake les fonctionnalités. Modifiez les rôles de service uniquement lorsque HealthLake vous recevez des instructions à cet effet.

Rôles liés à un service pour AWS HealthLake

Prend en charge les rôles liés à un service : non

Un rôle lié à un service est un type de rôle de service lié à un. Service AWS Le service peut endosser le rôle afin d'effectuer une action en votre nom. Les rôles liés à un service apparaissent dans votre Compte AWS répertoire et appartiennent au service. Un administrateur IAM peut consulter, mais ne peut pas modifier les autorisations concernant les rôles liés à un service.

Pour plus de détails sur la création ou la gestion des rôles liés à un service, consultez la section AWS Services compatibles avec. IAM Recherchez un service dans le tableau qui inclut un `Yes` dans la colonne Rôle lié à un service. Choisissez le lien Oui pour consulter la documentation du rôle lié à ce service.

Exemples de politiques basées sur l'identité pour AWS HealthLake

Par défaut, les utilisateurs et les rôles ne sont pas autorisés à créer ou modifier les ressources HealthLake. Ils ne peuvent pas non plus effectuer de tâches en utilisant le AWS Management Console, AWS Command Line Interface (AWS CLI) ou AWS API. Pour autoriser les utilisateurs à effectuer des actions sur les ressources dont ils ont besoin, un IAM administrateur peut créer des IAM politiques. L'administrateur peut ensuite ajouter les IAM politiques aux rôles, et les utilisateurs peuvent assumer les rôles.

Pour savoir comment créer une politique IAM basée sur l'identité à l'aide de ces exemples de documents de JSON stratégie, voir Créer des IAM politiques (console) dans le guide de l'IAMutilisateur.

Pour plus de détails sur les actions et les types de ressources définis par HealthLake, y compris le format de ARNs pour chacun des types de ressources, voir Actions, ressources et clés de condition AWS HealthLake dans la référence d'autorisation de service.

Rubriques

Bonnes pratiques en matière de politiques

Les politiques basées sur l'identité déterminent si quelqu'un peut créer, accéder ou supprimer HealthLake des ressources dans votre compte. Ces actions peuvent entraîner des frais pour votre Compte AWS. Lorsque vous créez ou modifiez des politiques basées sur l'identité, suivez ces instructions et recommandations :

- Commencez AWS par les politiques gérées et passez aux autorisations du moindre privilège : pour commencer à accorder des autorisations à vos utilisateurs et à vos charges de travail, utilisez les politiques AWS gérées qui accordent des autorisations pour de nombreux cas d'utilisation courants. Ils sont disponibles dans votre Compte AWS. Nous vous recommandons de réduire davantage les autorisations en définissant des politiques gérées par les AWS clients spécifiques à vos cas d'utilisation. Pour plus d'informations, consultez les politiques AWS gérées ou les politiques AWS gérées pour les fonctions professionnelles dans le Guide de IAM l'utilisateur.
- Appliquer les autorisations du moindre privilège : lorsque vous définissez des autorisations à IAM l'aide de politiques, accordez uniquement les autorisations nécessaires à l'exécution d'une tâche. Pour ce faire, vous définissez les actions qui peuvent être entreprises sur des ressources spécifiques dans des conditions spécifiques, également appelées autorisations de moindre privilège. Pour plus d'informations sur l'utilisation IAM pour appliquer des autorisations, consultez la section Politiques et autorisations du Guide de IAM l'utilisateur. IAM
- Utilisez des conditions dans IAM les politiques pour restreindre davantage l'accès : vous pouvez ajouter une condition à vos politiques pour limiter l'accès aux actions et aux ressources. Par exemple, vous pouvez rédiger une condition de politique pour spécifier que toutes les demandes doivent être envoyées en utilisantSSL. Vous pouvez également utiliser des conditions pour accorder l'accès aux actions de service si elles sont utilisées par le biais d'un service spécifique

Service AWS, tel que AWS CloudFormation. Pour plus d'informations, voir Éléments IAM JSON de politique : Condition dans le guide de IAM l'utilisateur.

- Utilisez IAM Access Analyzer pour valider vos IAM politiques afin de garantir des autorisations sécurisées et fonctionnelles. IAM Access Analyzer valide les politiques nouvelles et existantes afin qu'elles soient conformes au langage des IAM politiques (JSON) et IAM aux meilleures pratiques. IAM Access Analyzer fournit plus de 100 vérifications des politiques et des recommandations exploitables pour vous aider à créer des politiques sécurisées et fonctionnelles. Pour plus d'informations, consultez la section Valider les politiques avec IAM Access Analyzer dans le guide de l'IAMutilisateur.
- Exiger l'authentification multifactorielle (MFA) : si vous avez un scénario qui nécessite des IAM utilisateurs ou un utilisateur root Compte AWS, activez-le MFA pour une sécurité supplémentaire. Pour exiger le MFA moment où les API opérations sont appelées, ajoutez MFA des conditions à vos politiques. Pour plus d'informations, consultez la section APIAccès sécurisé avec MFA dans le guide de IAM l'utilisateur.

Pour plus d'informations sur les meilleures pratiques en matière de sécuritéIAM, consultez la section Bonnes pratiques en matière de sécurité IAM dans le Guide de IAM l'utilisateur.

Utilisation de la console AWS HealthLake

Pour accéder à la AWS HealthLake console, vous devez disposer d'un ensemble minimal d'autorisations. Ces autorisations doivent vous permettre de répertorier et d'afficher les détails HealthLake des ressources de votre Compte AWS. Si vous créez une politique basée sur l'identité qui est plus restrictive que l'ensemble minimum d'autorisations requis, la console ne fonctionnera pas comme prévu pour les entités (utilisateurs ou rôles) tributaires de cette politique.

Il n'est pas nécessaire d'accorder des autorisations de console minimales aux utilisateurs qui appellent uniquement le AWS CLI ou le AWS API. Au lieu de cela, autorisez uniquement l'accès aux actions correspondant à l'APIopération qu'ils tentent d'effectuer.

Pour un accès complet à HealthLake, associez les politiques suivantes à un IAM utilisateur ou à un rôle : `AmazonHealthLakeFullAccess` et`AWSLakeFormationDataAdmin`. Vous devez également associer la politique HealthLake intégrée qui est un rôle de service. Un rôle de service est un IAMrôle qu'un service assume pour effectuer des actions en votre nom. Un administrateur IAM peut créer, modifier et supprimer une fonction du service à partir de IAM. Pour plus d'informations, consultez la section Créer un rôle pour déléguer des autorisations à un Service AWS dans le guide de IAM l'utilisateur. Pour plus d'informations sur la politique intégrée qui crée le rôle de service requis,

consultezConfiguration des autorisations pour commencer à utiliser AWS HealthLake. Vous devez également utiliser la AWS Lake Formation console ou CLI désigner votre HealthLake administrateur comme administrateur de AWS Lake Formation Data Lake. Pour de plus amples informations, veuillez consulter Configuration des autorisations pour commencer à utiliser AWS HealthLake.

Accès à un magasin AWS HealthLake de données dans Amazon Athena

Si vous souhaitez permettre aux utilisateurs et aux rôles d'accéder aux magasins de HealthLake données Amazon Athena, associez les IAM politiques suivantes au rôle ou à l'utilisateur : `AmazonAthenaFullAccess` et`AmazonS3FullAccess`. `Select`et `Describe` des autorisations sont également requises sur les tables gérées par AWS Lake Formation. AWS Lake Formation les autorisations de table sont accordées par un AWS Lake Formation administrateur dans la AWS Lake Formation console ou via leCLI. Pour plus d'informations, consultez Configuration des autorisations pour commencer à utiliser AWS HealthLake.

Autoriser des utilisateurs à afficher leurs propres autorisations

Cet exemple montre comment créer une stratégie qui permet aux utilisateurs IAM d'afficher les stratégies en ligne et gérées attachées à leur identité d'utilisateur. Cette politique inclut les autorisations permettant d'effectuer cette action sur la console ou par programmation à l'aide du AWS CLI ou. AWS API

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
```

```
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS politiques gérées pour AWS HealthLake

Une politique AWS gérée est une politique autonome créée et administrée par AWS. AWS les politiques gérées sont conçues pour fournir des autorisations pour de nombreux cas d'utilisation courants afin que vous puissiez commencer à attribuer des autorisations aux utilisateurs, aux groupes et aux rôles.

N'oubliez pas que les politiques AWS gérées peuvent ne pas accorder d'autorisations de moindre privilège pour vos cas d'utilisation spécifiques, car elles sont accessibles à tous les AWS clients. Nous vous recommandons de réduire encore les autorisations en définissant des politiques gérées par le client qui sont propres à vos cas d’utilisation.

Vous ne pouvez pas modifier les autorisations définies dans les politiques AWS gérées. Si les autorisations définies dans une politique AWS gérée sont AWS mises à jour, la mise à jour affecte toutes les identités principales (utilisateurs, groupes et rôles) auxquelles la politique est attachée. AWS est le plus susceptible de mettre à jour une politique AWS gérée lorsqu'une nouvelle Service AWS est lancée ou que de nouvelles API opérations sont disponibles pour les services existants.

Pour plus d’informations, consultez la rubrique AWS Politiques gérées dans le IAMGuide de l’utilisateur.

AWS politique gérée : AmazonHealthLakeFullAccess

La `AmazonHealthLakeFullAccess` politique fournit un accès complet à HealthLake. Cette politique étant attachée à leur utilisateur ou à leur rôle, les utilisateurs peuvent l'utiliser HealthLake pour accéder aux données, les interroger, les importer et les exporter HealthLake. Pour effectuer de nombreuses actions courantes dans HealthLake, vous devez ajouter des politiques supplémentaires à l'utilisateur ou au rôle. Pour plus d'informations, consultez la section HealthLake Opérations Configuration des autorisations pour commencer à utiliser AWS HealthLake et autorisations.

Vous pouvez attacher la politique `AmazonHealthLakeFullAccess` à vos identités IAM.

Cette politique accorde des *`administrative and contributor`* autorisations qui permettent aux utilisateurs et aux rôles d'effectuer des requêtes, de rechercher HealthLake, d'importer et d'exporter, et elle permet également d' HealthLake effectuer des actions au nom des utilisateurs et des rôles dotés de ces autorisations.

Détails de l’autorisation

Cette politique inclut la déclaration suivante.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Action": [
    "healthlake:*",
    "s3:ListAllMyBuckets",
    "s3:ListBucket",
    "s3:GetBucketLocation",
    "iam:ListRoles"
   ],
   "Resource": "*",
   "Effect": "Allow"
  },
  {
   "Effect": "Allow",
   "Action": "iam:PassRole",
   "Resource": "*",
```

```
   "Condition": {
    "StringEquals": {
     "iam:PassedToService": "healthlake.amazonaws.com"
    }
   }
  }
 ]
}
```

AWS politique gérée : AmazonHealthLakeReadOnlyAccess

`AmazonHealthLakeReadOnlyAccess`la politique accorde un accès et des autorisations en lecture seule HealthLake aux ressources associées dans d'autres AWS services. Appliquez cette politique aux utilisateurs auxquels vous souhaitez accorder la possibilité d'interroger et de consulter HealthLake des banques de données, mais pas la possibilité de les créer ou d'y apporter des modifications.

Vous pouvez attacher la politique `AmazonHealthLakeReadOnlyAccess` à vos identités IAM.

Cette politique accorde des *`read-only`* autorisations qui permettent aux utilisateurs et aux rôles d'effectuer des requêtes HealthLake.

Détails de l’autorisation

Cette politique inclut la déclaration suivante.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "healthlake:ListFHIRDatastores",
                "healthlake:DescribeFHIRDatastore",
                "healthlake:DescribeFHIRImportJob",
                "healthlake:DescribeFHIRExportJob",
                "healthlake:GetCapabilities",
                "healthlake:ReadResource",
```

```
                "healthlake:SearchWithGet",
                "healthlake:SearchWithPost",
                "healthlake:SearchEverything"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

HealthLake opérations et autorisations

Le tableau suivant répertorie les opérations typiques HealthLake et les autorisations nécessaires pour les exécuter.

HealthLake opérations	Autorisations nécessaires
Créez un magasin de données dans HealthLake	`AmazonHealthLakeFullAccess` `AmazonLakeFormationDataAdmin` , politique intégrée et autorisations d' AWS Lake Formation administrateur gérées par AWS Lake Formation
Supprimer un magasin de données dans HealthLake	`AmazonHealthLakeFullAccess` `AmazonLakeFormationDataAdmin` , politique intégrée et autorisations d' AWS Lake Formation administrateur gérées par AWS Lake Formation
Répertorier, rechercher ou interroger un magasin de données dans HealthLake	`AmazonHealthLakeReadOnlyAccess`
Interrogez un magasin de données à l'aide de Amazon Athena	`AmazonAthenaFullAccess` `AmazonS3FullAccess` , AWS Lake Formation `Select` et les `Describe` autorisations sur les tables gérées par AWS Lake Formation
Importer des données depuis HealthLake	Consultez Configuration des autorisations pour les tâches d'importation.

HealthLake opérations	Autorisations nécessaires
Exporter des données depuis HealthLake	Consultez Exportation de fichiers depuis votre banque de données (AWS SDKs).

HealthLake mises à jour des politiques AWS gérées

Consultez les informations relatives aux mises à jour des politiques AWS gérées HealthLake depuis le moment où ce service a commencé à suivre ces modifications. Pour recevoir des alertes automatiques concernant les modifications apportées à cette page, abonnez-vous au RSS fil sur la page Historique du HealthLake document.

Modification	Description	Date
AmazonHealthLakeFullAccess	`AmazonHealthLakeFullAccess` politique requise pour permettre un accès complet à HealthLake.	14 novembre 2022
AmazonHealthLakeReadOnlyAccess	`AmazonHealthLakeReadOnlyAccess` politique requise pour l'accès en lecture seule à. HealthLake	14 novembre 2022
HealthLake a commencé à suivre les modifications	HealthLake a commencé à suivre les modifications apportées AWS à ses politiques gérées.	14 novembre 2022

Résolution des problèmes AWS HealthLake d'identité et d'accès

Utilisez les informations suivantes pour vous aider à diagnostiquer et à résoudre les problèmes courants que vous pouvez rencontrer lorsque vous travaillez avec HealthLake etIAM.

Rubriques

- Je ne suis pas autorisé à effectuer une action dans AWS HealthLake

- Je ne suis pas autorisé à effectuer iam : PassRole
- Je souhaite autoriser des personnes extérieures à mon AWS compte à accéder à mes AWS HealthLake ressources

Je ne suis pas autorisé à effectuer une action dans AWS HealthLake

S'il vous AWS Management Console indique que vous n'êtes pas autorisé à effectuer une action, vous devez contacter votre administrateur pour obtenir de l'aide. Votre administrateur est la personne qui vous a fourni votre nom d'utilisateur et votre mot de passe.

L'exemple d'erreur suivant se produit lorsque l'`mateojackson`IAMutilisateur essaie d'utiliser la console pour afficher les détails d'une *`my-example-widget`* ressource fictive mais ne dispose pas des `healthlake:`*`GetWidget`* autorisations fictives.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 healthlake:GetWidget on resource: my-example-widget
```

Dans ce cas, Mateo demande à son administrateur de mettre à jour ses politiques pour lui permettre d'accéder à la ressource *`my-example-widget`* à l'aide de l'action `healthlake:`*`GetWidget`*.

Je ne suis pas autorisé à effectuer iam : PassRole

Si vous recevez une erreur selon laquelle vous n'êtes pas autorisé à exécuter `iam:PassRole` l'action, vos stratégies doivent être mises à jour afin de vous permettre de transmettre un rôle à HealthLake.

Certains vous Services AWS permettent de transmettre un rôle existant à ce service au lieu de créer un nouveau rôle de service ou un rôle lié à un service. Pour ce faire, un utilisateur doit disposer des autorisations nécessaires pour transmettre le rôle au service.

L'exemple d'erreur suivant se produit lorsqu'un utilisateur IAM nommé `marymajor` essaie d'utiliser la console pour exécuter une action dans HealthLake. Toutefois, l'action nécessite que le service ait des autorisations accordées par un rôle de service. Mary ne dispose pas des autorisations nécessaires pour transférer le rôle au service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Dans ce cas, les politiques de Mary doivent être mises à jour pour lui permettre d'exécuter l'action `iam:PassRole`.

Si vous avez besoin d'aide, contactez votre AWS administrateur. Votre administrateur vous a fourni vos informations d'identification de connexion.

Je souhaite autoriser des personnes extérieures à mon AWS compte à accéder à mes AWS HealthLake ressources

Vous pouvez créer un rôle que les utilisateurs provenant d'autres comptes ou les personnes extérieures à votre organisation pourront utiliser pour accéder à vos ressources. Vous pouvez spécifier qui est autorisé à assumer le rôle. Pour les services qui prennent en charge les politiques basées sur les ressources ou les listes de contrôle d'accès (ACLs), vous pouvez utiliser ces politiques pour autoriser les utilisateurs à accéder à vos ressources.

Pour en savoir plus, consultez les éléments suivants :

- Pour savoir si ces fonctionnalités sont prises HealthLake en charge, consultezComment AWS HealthLake fonctionne avec IAM.
- Pour savoir comment donner accès à vos ressources sur un site Comptes AWS qui vous appartient, consultez la section Fournir l'accès à un IAM utilisateur dans un autre site Compte AWS que vous possédez dans le Guide de IAM l'utilisateur.
- Pour savoir comment fournir l'accès à vos ressources à des tiers Comptes AWS, consultez la section Fournir un accès à des ressources Comptes AWS détenues par des tiers dans le Guide de IAM l'utilisateur.
- Pour savoir comment fournir un accès via la fédération d'identité, consultez la section Fournir un accès aux utilisateurs authentifiés de manière externe (fédération d'identité) dans le guide de l'IAMutilisateur.
- Pour connaître la différence entre l'utilisation de rôles et l'utilisation de politiques basées sur les ressources pour l'accès entre comptes, voir Accès aux ressources entre comptes IAM dans le guide de l'IAMutilisateur.

Journalisation des appels AWS HealthLake API avec AWS CloudTrail

AWS HealthLake est intégré à AWS CloudTrail un service qui fournit un enregistrement des actions entreprises par un utilisateur, un rôle ou un AWS service dans HealthLake. CloudTrail capture tous

les API appels HealthLake sous forme d'événements. Les appels capturés incluent des appels provenant de la HealthLake console et des appels de code vers les HealthLake API opérations. Si vous créez un suivi, vous pouvez activer la diffusion continue d' CloudTrailévénements vers un compartiment Amazon S3, y compris les événements pour HealthLake. Si vous ne configurez pas de suivi, vous pouvez toujours consulter les événements les plus récents dans la CloudTrail console dans Historique des événements. À l'aide des informations collectées par CloudTrail, vous pouvez déterminer la demande qui a été faite HealthLake, l'adresse IP à partir de laquelle la demande a été faite, qui a fait la demande, quand elle a été faite et des détails supplémentaires.

Pour en savoir plus CloudTrail, consultez le guide de AWS CloudTrail l'utilisateur.

AWS HealthLake Informations dans CloudTrail

CloudTrail est activé sur votre AWS compte lorsque vous le créez. Lorsqu'une activité se produit dans HealthLake, cette activité est enregistrée dans un CloudTrail événement avec d'autres événements de AWS service dans l'historique des événements. Vous pouvez afficher, rechercher et télécharger les événements récents dans votre compte AWS . Pour plus d'informations, consultez la section Affichage des événements avec l'historique des CloudTrail événements.

Pour un enregistrement continu des événements de votre AWS compte, y compris des événements pour HealthLake, créez un parcours. Un suivi permet CloudTrail de fournir des fichiers journaux à un compartiment Amazon S3. Par défaut, lorsque vous créez un journal de suivi dans la console, il s’applique à toutes les régions AWS. Le journal enregistre les événements de toutes les régions de la AWS partition et transmet les fichiers journaux au compartiment Amazon S3 que vous spécifiez. En outre, vous pouvez configurer d'autres AWS services pour analyser plus en détail les données d'événements collectées dans les CloudTrail journaux et agir en conséquence. Pour plus d’informations, consultez les ressources suivantes :

- Vue d’ensemble de la création d’un journal d’activité
- CloudTrail Services et intégrations pris en charge
- Configuration des SNS notifications Amazon pour CloudTrail
- Réception de fichiers CloudTrail journaux de plusieurs régions et réception de fichiers CloudTrail journaux de plusieurs comptes

Toutes les HealthLake actions sont enregistrées CloudTrail et documentées dans la HealthLake APIréférence et dans ce guide du développeur pour les actions effectuées à l'aide du FHIR

RESTAPI. Par exemple, les appels aux actions suivantes génèrent des entrées dans les fichiers CloudTrail journaux :

- `DescribeFHIRImportJob`
- `DescribeFHIRExportJob`
- `StartFHIRImportJob`
- `ListFHIRImportJobs`
- `StartFHIRExportJob`
- `ListFHIRExportJobs`
- `CreateFHIRDatastore`
- `ListFHIRDatastores`
- `DeleteFHIRDatastore`
- `DescribeFHIRDatastore`
- `UpdateResource`
- `CreateResource`
- `DeleteResource`
- `ReadResource`
- `GetCapabilities`
- `SearchWithGet`
- `SearchWithPost`

Chaque événement ou entrée de journal contient des informations sur la personne ayant initié la demande. Les informations relatives à l'identité permettent de déterminer les éléments suivants :

- Si la demande a été faite avec les informations d'identification de l'utilisateur root ou AWS Identity and Access Management (IAM).
- Si la demande a été effectuée avec les informations d'identification de sécurité temporaires d'un rôle ou d'un utilisateur fédéré.
- Si la demande a été faite par un autre AWS service.

Pour plus d'informations, consultez l'CloudTrail userIdentityélément.

Comprendre les entrées du fichier AWS HealthLake journal

Un suivi est une configuration qui permet de transmettre des événements sous forme de fichiers journaux à un compartiment Amazon S3 que vous spécifiez. CloudTrail les fichiers journaux contiennent une ou plusieurs entrées de journal. Un événement représente une demande unique provenant de n'importe quelle source et inclut des informations sur l'action demandée, la date et l'heure de l'action, les paramètres de la demande, etc. CloudTrail les fichiers journaux ne constituent pas une trace ordonnée des API appels publics, ils n'apparaissent donc pas dans un ordre spécifique.

L'exemple suivant montre une entrée de CloudTrail journal illustrant l'`CreateFHIRDatastore`action.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROA2B3ZHOADD2OJ4AHJX:git
 full_access_iam_role580074395690222150",
        "arn": "arn:aws:sts::691207106566:assumed-role/
colossusfrontend_full_access_iam_role/_iam_role580074395690222150",
        "accountId": "AccountID",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROA2B3ZHOADD2OJ4AHJX",
                "arn": "arn:aws:iam::691207106566:role/full_access_iam_role",
                "accountId": "AccountID",
                "userName": "full_access_iam_role"
            },
            "webIdFederationData": {

            },
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2020-11-20T00:08:15Z"
            }
        }
    },
    "eventTime": "2020-11-20T00:08:16Z",
    "eventSource": "healthlake.amazonaws.com",
```

```
    "eventName": "CreateFHIRDatastore",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "3.213.247.1",
    "userAgent": "Coral/Netty4",
    "requestParameters": {
        "datastoreName":
 "testCreateFHIRDatastore_GBYAZFCLLBLSUTOYYFQZRLBLQJNFOYQVRPZBOJAIIUAHICAEAGIWLNVQEYAMSXVWMBLXC
        "datastoreTypeVersion": "R4",
        "clientToken": "d737ffe0-14dd-44cc-9f0a-fdf59b26c66b"
    },
    "responseElements": {
        "datastoreId": "datastoreID",
        "datastoreArn": "arn:aws:healthlake:us-
east-1:691207106566:datastore/55576c487ff4975262b10d1d65eb4509",
        "datastoreStatus": "CREATING",
        "datastoreEndpoint": "datastore_endpoint/"
    },
    "requestID": "68e62bdd-d2d4-44c1-af69-e6f055a69f99",
    "eventID": "7ef483dc-5dca-469e-823a-7d9e3a7fe924",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "691207106566"
}
```


Validation de conformité pour AWS HealthLake

Des auditeurs tiers évaluent la sécurité et AWS HealthLake la conformité de plusieurs programmes de AWS conformité. Car HealthLake cela inclutHIPAA.

Pour savoir si un programme Services AWS de conformité Service AWS s'inscrit dans le champ d'application de programmes de conformité spécifiques, consultez Services AWS la section de conformité et sélectionnez le programme de conformité qui vous intéresse. Pour des informations générales, voir Programmes de AWS conformité Programmes AWS de .

Vous pouvez télécharger des rapports d'audit tiers à l'aide de AWS Artifact. Pour plus d'informations, voir Téléchargement de rapports dans AWS Artifact .

Votre responsabilité en matière de conformité lors de l'utilisation Services AWS est déterminée par la sensibilité de vos données, les objectifs de conformité de votre entreprise et les lois et

réglementations applicables. AWS fournit les ressources suivantes pour faciliter la mise en conformité :

- Conformité et gouvernance de la sécurité : ces guides de mise en œuvre de solutions traitent des considérations architecturales et fournissent les étapes à suivre afin de déployer des fonctionnalités de sécurité et de conformité.
- Architecture axée sur la HIPAA sécurité et la conformité sur Amazon Web Services : ce livre blanc décrit comment les entreprises peuvent AWS créer HIPAA des applications éligibles.

> **Note**
>
> Tous ne Services AWS sont pas HIPAA éligibles. Pour plus d'informations, consultez la référence des services HIPAA éligibles.

- AWS Ressources de https://aws.amazon.com/compliance/resources/ de conformité — Cette collection de classeurs et de guides peut s'appliquer à votre secteur d'activité et à votre région.
- AWS Guides de conformité destinés aux clients — Comprenez le modèle de responsabilité partagée sous l'angle de la conformité. Les guides résument les meilleures pratiques en matière de sécurisation Services AWS et reprennent les directives relatives aux contrôles de sécurité dans de nombreux cadres (notamment le National Institute of Standards and Technology (NIST), le Payment Card Industry Security Standards Council (PCI) et l'Organisation internationale de normalisation (ISO)).
- Évaluation des ressources à l'aide des règles du guide du AWS Config développeur : le AWS Config service évalue dans quelle mesure les configurations de vos ressources sont conformes aux pratiques internes, aux directives du secteur et aux réglementations.
- AWS Security Hub— Cela Service AWS fournit une vue complète de votre état de sécurité interne AWS. Security Hub utilise des contrôles de sécurité pour évaluer vos ressources AWS et vérifier votre conformité par rapport aux normes et aux bonnes pratiques du secteur de la sécurité. Pour obtenir la liste des services et des contrôles pris en charge, consultez Référence des contrôles Security Hub.
- Amazon GuardDuty — Cela Service AWS détecte les menaces potentielles qui pèsent sur vos charges de travail Comptes AWS, vos conteneurs et vos données en surveillant votre environnement pour détecter toute activité suspecte et malveillante. GuardDuty peut vous aider à répondre à diverses exigences de conformité PCIDSS, par exemple en répondant aux exigences de détection des intrusions imposées par certains cadres de conformité.

- AWS Audit Manager— Cela vous Service AWS permet d'auditer en permanence votre AWS utilisation afin de simplifier la gestion des risques et la conformité aux réglementations et aux normes du secteur.

Résilience dans AWS HealthLake

L'infrastructure AWS mondiale est construite autour des AWS régions et des zones de disponibilité. AWS Les régions fournissent plusieurs zones de disponibilité physiquement séparées et isolées, connectées par un réseau à faible latence, à haut débit et hautement redondant. Avec les zones de disponibilité, vous pouvez concevoir et exploiter des applications et des bases de données qui basculent automatiquement d'une zone à l'autre sans interruption. Les zones de disponibilité sont davantage disponibles, tolérantes aux pannes et ont une plus grande capacité de mise à l'échelle que les infrastructures traditionnelles à un ou plusieurs centres de données.

Pour plus d'informations sur AWS les régions et les zones de disponibilité, consultez la section Infrastructure AWS mondiale.

Outre l'infrastructure AWS mondiale, HealthLake propose plusieurs fonctionnalités pour répondre à vos besoins en matière de résilience et de sauvegarde des données.

Sécurité de l'infrastructure dans AWS HealthLake

En tant que service géré, AWS HealthLake il est protégé par les procédures de sécurité du réseau AWS mondial décrites dans le livre blanc Amazon Web Services : présentation des processus de sécurité.

Vous utilisez API les appels AWS publiés pour accéder HealthLake via le réseau. Les clients doivent prendre en charge Transport Layer Security (TLS) 1.0 ou version ultérieure. Nous recommandons la version TLS 1.2 ou une version ultérieure. Les clients doivent également prendre en charge les suites de chiffrement parfaitement confidentielles (), telles que Ephemeral Diffie-Hellman (PFS) ou Elliptic Curve Ephemeral Diffie-Hellman (DHE). ECDHE La plupart des systèmes modernes tels que Java 7 et les versions ultérieures prennent en charge ces modes.

En outre, les demandes doivent être signées à l'aide d'un ID de clé d'accès et d'une clé d'accès secrète associée à un principal IAM. Vous pouvez également utiliser AWS Security Token Service (AWS STS) pour générer des informations d'identification de sécurité temporaires et signer les demandes.

Bonnes pratiques de sécurité dans AWS HealthLake

AWS HealthLake fournit un certain nombre de fonctionnalités de sécurité à prendre en compte lors de l'élaboration et de la mise en œuvre de vos propres politiques de sécurité. Les bonnes pratiques suivantes doivent être considérées comme des instructions générales et ne représentent pas une solution de sécurité complète. Étant donné que ces bonnes pratiques peuvent ne pas être appropriées ou suffisantes pour votre environnement, considérez-les comme des remarques utiles plutôt que comme des recommandations.

- Mettez en œuvre l'accès avec le moindre privilège.
- Dans la mesure du possible, utilisez Customer-Managed-Keys (CMKs) pour chiffrer vos données. Pour en savoir plusCMKs, consultez Amazon Key Management Service.
- Utilisez Rechercher avecPOST, et non Rechercher avec, GET lorsque vous recherchez PHI ou parcourez PII votre banque de données.
- Limitez l'accès aux fonctions d'audit sensibles et importantes.
- Lorsque vous créez des ressources par le biais de la mise à jour ou de l'importation en bloc APIsPII, n'utilisez pas PHI ou, y compris les noms des banques de données et des tâches, dans les champs visibles ou dans l'FHIRID logique (LID).
- Lorsque vous envoyez des demandes de création, de lecture, de mise à jour, de suppression ou de recherche, ne les utilisez pas PHI dans l'HTTPen-tête.
- Activez cette option AWS CloudTrail pour auditer AWS HealthLake l'utilisation et vous assurer qu'il n'y a aucune activité inattendue.
- Passez en revue les meilleures pratiques pour utiliser les compartiments Amazon S3 en toute sécurité. Pour en savoir plus, consultez les meilleures pratiques en matière de sécurité dans le guide de l'utilisateur d'Amazon S3.

AWS HealthLake points de terminaison et quotas

Les sections suivantes contiennent des informations sur les AWS HealthLake quotas et les points de terminaison. Pour les quotas ajustables, vous pouvez demander une augmentation de quota à l'aide de la console Service Quotas. Pour plus d'informations, consultez Demande d'augmentation de quota dans le Guide de l'utilisateur Service Quotas.

Points de terminaison de service

Le tableau indique les points de terminaison HealthLake de service disponibles dans une région donnée.

Nom de la région	Région	Point de terminaison	Protocole
US East (Ohio)	us-east-2	healthlake.us-east-2.amazonaws.com	HTTPS
		healthlake-fips.us-east-2.amazonaws.com	HTTPS
US East (N. Virginia)	us-east-1	healthlake.us-east-1.amazonaws.com	HTTPS
		healthlake-fips.us-east-1.amazonaws.com	HTTPS
USA Ouest (Oregon)	us-west-2	healthlake.us-west-2.amazonaws.com	HTTPS
		healthlake-fips.us-west-2.amazonaws.com	HTTPS
Asia Pacific (Mumbai)	ap-south-1	healthlake.ap-south-1.amazonaws.com	HTTPS
Asie-Pacifique (Sydney)	ap-southeast-2	healthlake.ap-southeast-2.amazonaws.com	HTTPS
Europe (Londres)	eu-west-2	healthlake.eu-west-2.amazonaws.com	HTTPS

Quotas de service pour HealthLake

Voici les quotas par défaut pour HealthLake.

Nom	Par défaut	Ajusta	Description
Nombre de caractères d'une note médicale	Chaque région prise en charge : 10 000	Non	Le nombre maximum de caractères d'une note médicale individuelle au sein du type de DocumentReference ressource (POST/PUT demandes).
Nombre de jobs S tartFHIRImport Job simultanés	Par région prise en charge : 1	Non	Nombre maximal de tâches S tartFHIRImport Job simultanées.
Nombre d'concurrentStartFHIRExport Jobemplois	Par région prise en charge : 1	Non	Nombre maximal de tâches S tartFHIRExport Job simultanées.
Nombre de magasins de données par compte	Par région prise en charge : 10	Oui	Le nombre maximum par défaut de magasins de données actifs par compte.
Nombre de fichiers dans un S tartFHIRImport Job	Chaque région prise en charge : 10 000	Non	Le nombre maximum de fichiers dans un S tartFHIRImport Job.
Nombre de ressources par bundle	Chaque région prise en charge : 160	Non	Le nombre maximum de ressources autorisées dans une demande de bundle.

Nom	Par défaut	Ajusta	Description
Taux de demandes groupées par compte	Chaque région prise en charge : 20	Oui	Le nombre maximum de demandes de POST bundle que vous pouvez effectuer par seconde et par compte.
Taux de demandes de bundle par magasin de données	Par région prise en charge : 10	Oui	Nombre maximal de demandes de POST bundle que vous pouvez effectuer par seconde et par magasin de données. Les banques de données créées avant le 21/08/2023 seront limitées à une demande par seconde.
Taux de demandes de C ancelFHIR Export Job utilisées DELETE par compte	Par région prise en charge : 1	Non	Le nombre maximum de demandes C ancelFHIR Export Job DELETE que vous pouvez utiliser par minute et par compte.
Taux de reateFHIRDatastore demandes C par compte	Par région prise en charge : 1	Non	Le nombre maximum de reateFHIRDatastore requêtes C que vous pouvez effectuer par minute et par compte.
Taux de DELETE demandes par compte	Chaque Région prise en charge : 2 000	Oui	Le nombre maximum de DELETE demandes que vous pouvez effectuer par seconde et par compte.

Nom	Par défaut	Ajusta	Description
Taux de DELETE demandes par magasin de données	Chaque Région prise en charge : 1 000	Oui	Nombre maximal de DELETE demandes que vous pouvez effectuer par seconde et par magasin de données. Les banques de données créées avant le 21/08/2023 seront limitées à 100 demandes par seconde.
Taux de eleteFHIRDatastore demandes D par compte	Par région prise en charge : 1	Non	Le nombre maximum de eleteFHIRDatastore demandes D que vous pouvez effectuer par minute et par compte.
Taux de escribeFHIRDatastore demandes D par compte	Chaque Région prise en charge : 10	Non	Le nombre maximum de escribeFHIRDatastore demandes D que vous pouvez effectuer par seconde et par compte.
Taux de demandes escribeFHIRExport d'emploi D par compte	Chaque Région prise en charge : 10	Non	Le nombre maximum de demandes D escribeFH IRExport Job que vous pouvez effectuer par seconde et par compte.
Taux de demandes escribeFHIRExport d'emploi D utilisées GET par compte	Chaque Région prise en charge : 10	Non	Le nombre maximum de demandes D escribeFH IRExport Job GET que vous pouvez utiliser par seconde et par compte.

Nom	Par défaut	Ajusta	Description
Taux de demandes escribeFHIRImport d'emploi D par compte	Chaque Région prise en charge : 10	Non	Le nombre maximum de demandes D escribeFH IRImport Job que vous pouvez effectuer par seconde et par compte.
Taux de demandes de découverte par compte	Chaque Région prise en charge : 10	Non	Le nombre maximum de demandes Discovery que vous pouvez effectuer par minute et par compte.
Taux de GET demandes par compte	Chaque région prise en charge : 6 000	Oui	Le nombre maximum de GET demandes que vous pouvez effectuer par seconde et par compte.
Taux de GET demandes par magasin de données	Chaque région prise en charge : 3 000	Oui	Nombre maximal de GET demandes que vous pouvez effectuer par seconde et par magasin de données. Les banques de données créées avant le 21/08/2023 seront limitées à 100 demandes par seconde.
Taux de GetCapabilities demandes par compte	Chaque Région prise en charge : 10	Non	Le nombre maximum de GetCapabilities demandes que vous pouvez effectuer par seconde et par compte.

Nom	Par défaut	Ajusta	Description
Taux de istFHIRDatastores demandes L par compte	Chaque Région prise en charge : 10	Non	Le nombre maximum de istFHIRDatastores demandes L que vous pouvez effectuer par seconde et par compte.
Taux de demandes L istFHIRExport Jobs par compte	Chaque Région prise en charge : 10	Non	Le nombre maximum de demandes L istFHIREx port Jobs que vous pouvez effectuer par seconde et par compte.
Taux de demandes L istFHIRImport Jobs par compte	Chaque Région prise en charge : 10	Non	Le nombre maximum de demandes L istFHIRIm port Jobs que vous pouvez effectuer par seconde et par compte.
Taux de ListTagsforResource demandes par compte	Chaque Région prise en charge : 10	Non	Le nombre maximum de ListTagsforResourc e demandes que vous pouvez effectuer par seconde et par compte.
Taux de POST demandes par compte	Chaque Région prise en charge : 2 000	Oui	Le nombre maximum de POST demandes que vous pouvez effectuer par seconde et par compte.

Nom	Par défaut	Ajusta	Description
Taux de POST demandes par magasin de données	Chaque Région prise en charge : 1 000	Oui	Nombre maximal de POST demandes que vous pouvez effectuer par seconde et par magasin de données. Les banques de données créées avant le 21/08/2023 seront limitées à 100 demandes par seconde.
Taux de PUT demandes par compte	Chaque Région prise en charge : 2 000	Oui	Le nombre maximum de PUT demandes que vous pouvez effectuer par seconde et par compte.
Taux de PUT demandes par magasin de données	Chaque Région prise en charge : 1 000	Oui	Nombre maximal de PUT demandes que vous pouvez effectuer par seconde et par magasin de données. Les banques de données créées avant le 21/08/2023 seront limitées à 100 demandes par seconde.
Taux de S demandes tartFHIRExport d'emploi par compte	Par région prise en charge : 1	Non	Le nombre maximum de demandes S tartFHIRE xport Job que vous pouvez effectuer par minute et par compte.

Nom	Par défaut	Ajusta	Description
Taux de demandes tartFHIRExport d'emploi utilisées POST par compte	Par région prise en charge : 1	Non	Le nombre maximum de demandes S tartFHIRE xport Job POST que vous pouvez utiliser par minute et par compte.
Taux de S demandes tartFHIRImport d'emploi par compte	Par région prise en charge : 1	Non	Le nombre maximum de demandes S tartFHIRI mport Job que vous pouvez effectuer par minute et par compte.
Taux de TagResource demandes par compte	Chaque Région prise en charge : 10	Non	Le nombre maximum de TagResource demandes que vous pouvez effectuer par seconde.
Taux de UntagResource demandes par compte	Chaque Région prise en charge : 10	Non	Le nombre maximum de UntagResource demandes que vous pouvez effectuer par seconde et par compte.
Taux de demandes de recherche utilisées GET par compte	Chaque région prise en charge : 200	Oui	Le nombre maximum de demandes de recherche GET que vous pouvez utiliser par seconde et par compte.
Taux de demandes de recherche utilisées GET par magasin de données	Chaque Région prise en charge : 100	Oui	Nombre maximal de demandes de recherche GET que vous pouvez effectuer par seconde et par magasin de données.

Nom	Par défaut	Ajusta	Description
Taux de demandes de recherche utilisées POST par compte	Chaque région prise en charge : 200	Oui	Le nombre maximal de demandes de recherche POST que vous pouvez utiliser par seconde.
Taux de demandes de recherche utilisées POST par magasin de données	Chaque Région prise en charge : 100	Oui	Nombre maximal de demandes de recherche POST que vous pouvez effectuer par seconde et par magasin de données.
Taille de chaque fichier importé	Chaque Région prise en charge : 5 giga-octets	Non	Taille maximale (en Go) d'un fichier individue l inclus dans un S tartFHIRImport Job.
Taille totale de la tâche d'importation	Chaque région prise en charge : 500 gigaoctets	Non	Taille maximale (en Go) de tous les fichiers inclus dans la tâche d'importa tion.

Résolution des problèmes

La documentation suivante peut vous aider à résoudre les problèmes que vous pourriez rencontrer lors de l'utilisation AWS HealthLake de.

Rubriques

Pourquoi ne puis-je pas créer un magasin HealthLake de données ?

Le 14 novembre 2022, les IAM autorisations requises pour créer un nouveau magasin de données ont été HealthLake mises à jour. Si vous n'avez pas mis à jour les politiques associées à l'utilisateur ou au rôle qui y accède HealthLake , le message d'erreur suivant s'affiche.

```
AccessDeniedException: Insufficient Lake Formation permission(s): Required Database on
 Catalog
```

Pour consulter les exigences IAM de politique mises à jour pour la création d'un magasin de données, voir Politique AWS gérée : AmazonHealthLakeFullAccess. Pour step-by-step savoir comment ajouter ces politiques à votre IAM utilisateur ou à votre rôle, consultezConfiguration des autorisations pour commencer à utiliser AWS HealthLake.

Pour créer un magasin de données, vous devez également utiliser une clé symétrique appartenant au client ou à AmazonKMS. Assurez-vous que vous disposez des autorisations appropriées dans votre IAM politique. Pour en savoir plus AWS KMS, consultez AWS Key Management Servicele guide du AWS Key Management Service développeur.

Nombre de magasins de données autorisés par compte dépassé

HealthLake dispose d'un quota de 10 magasins de données par compte. Pour savoir comment demander une augmentation de quota, consultez le Centre de AWS support.

Comment créer une autorisation pour le FHIR RESTful APIs ?

Les utilisateurs doivent utiliser un processus de signature Signature version 4 pour ajouter l'authentification aux HealthLake API demandes envoyées via un HTTP client. Pour en savoir plus, consultez la section Processus de signature de la version 4 de Signature.

Pour créer une autorisation sigv4 à l'aide de AWS SDK for Python, créez un script similaire à l'exemple suivant.

```
import boto3
import requests
import json
from requests_auth_aws_sigv4 import AWSSigV4

# Set the input arguments
data_store_endpoint = 'https://healthlake.us-east-1.amazonaws.com/datastore/<datastore
 id>/r4//'
resource_path = "Patient"
requestBody = {"resourceType": "Patient", "active": True, "name": [{"use":
 "official","family": "Dow","given": ["Jen"]},{"use": "usual","given":
 ["Jen"]}],"gender": "female","birthDate": "1966-09-01"}
```

```
region = 'us-east-1'

#Frame the resource endpoint
resource_endpoint = data_store_endpoint+resource_path
session = boto3.session.Session(region_name=region)
client = session.client("healthlake")

# Frame authorization
auth = AWSSigV4("healthlake", session=session)

# Calling data store FHIR endpoint using SigV4 auth

r = requests.post(resource_endpoint, json=requestBody, auth=auth, )
print(r.json())
```

Des informations supplémentaires sur l'utilisation de l'autorisation sigv4 AWS SDK pour Python sont disponibles dans la rubrique sur les informations d'identification Boto3.

Mes données ne sont pas au format FHIR R4, puis-je toujours les utiliser ? HealthLake

Seules les données au format FHIR R4 peuvent être importées dans un magasin de HealthLake données. Pour consulter la liste des partenaires proposant des produits destinés à aider les utilisateurs à transformer leurs données, consultez la section AWS HealthLake Partenaires.

Pourquoi est-ce que je reçois AccessDenied des erreurs lors de l'utilisation FHIR RESTful APIs d'un magasin de données crypté avec une KMS clé gérée par le client ?

Les autorisations relatives à la clé gérée par le client et aux IAM politiques sont requises pour qu'un utilisateur ou un rôle puisse accéder à un magasin de données. Un utilisateur doit disposer des IAM autorisations requises pour utiliser une clé gérée par le client. Si un utilisateur a révoqué ou retiré une HealthLake autorisation autorisant l'utilisation de la KMS clé gérée par le client, un AccessDenied message d'erreur HealthLake sera renvoyé.

HealthLake doit avoir l'autorisation d'accéder aux données des clients, de chiffrer les nouvelles FHIR ressources importées dans un magasin de données et de déchiffrer les FHIR ressources lorsqu'elles sont demandées.

Pour en savoir plus, consultez la section Résolution des problèmes d'accès par clé.

Pourquoi mon importation a-t-elle échoué ?

Une tâche d'importation réussie générera un dossier contenant les fichiers inputFileName .ndjson en sortie, mais l'importation d'enregistrements individuels peut échouer. Dans ce cas, un deuxième FAILURE dossier sera généré avec un manifeste des enregistrements qui n'ont pas pu être importés. L'emplacement de sortie de la tâche pour accéder au fichier manifeste est JobProperties. JobOutputDataConfigConfiguration .S3. Uri S3.

Ce fichier manifeste contient des détails sur le résultat de la tâche, tels que l'emplacement de toutes les réponses réussies (successOutput. successOutputS3Uri), l'emplacement de toutes les réponses ayant échoué (. failureOutput failureOutputS3Uri) et des indicateurs de travail supplémentaires.
Le contenu du fichier manifeste est analysable par programmation. L'exemple de fichier manifeste suivant répertorie les compartiments Amazon S3 en entrée et en sortie, ainsi que des informations sur le nombre de ressources analysées et le nombre de ressources importées avec succès.

```
            {
            "inputDataConfig": {
            "s3Uri": "s3://amzn-s3-demo-source-bucket/healthlake-input/invalidInput/"
            },
            "outputDataConfig": {
            "s3Uri": "s3://amzn-s3-demo-logging-
bucket/32839038a2f47f17c2fe0f53f0c3a0ba-FHIR_IMPORT-19dd7bb7bcc8ee12a09bf6d322744a3d/",
            "encryptionKeyID": "arn:aws:kms:us-west-2:123456789012:key/
fbbbfee3-20b3-42a5-a99d-c48c655ed545"
            },
            "successOutput": {
            "successOutputS3Uri": "s3://amzn-s3-demo-logging-
bucket/32839038a2f47f17c2fe0f53f0c3a0ba-FHIR_IMPORT-19dd7bb7bcc8ee12a09bf6d322744a3d/
SUCCESS/"
            },
            "failureOutput": {
            "failureOutputS3Uri": "s3://amzn-s3-demo-logging-
bucket/32839038a2f47f17c2fe0f53f0c3a0ba-FHIR_IMPORT-19dd7bb7bcc8ee12a09bf6d322744a3d/
FAILURE/"
            },
            "numberOfScannedFiles": 1,
            "numberOfFilesImported": 1,
            "sizeOfScannedFilesInMB": 0.023627,
```

```
            "sizeOfDataImportedSuccessfullyInMB": 0.011232,
            "numberOfResourcesScanned": 9,
            "numberOfResourcesImportedSuccessfully": 4,
            "numberOfResourcesWithCustomerError": 5,
            "numberOfResourcesWithServerError": 0
            }
```

Pour analyser pourquoi une tâche d'importation a échoué, utilisez le D escribeFHIRImport Job API pour analyser le JobProperties. Il est recommandé de procéder comme suit :

- Si le statut est FAILED et qu'un message est présent, les échecs sont liés à des paramètres de tâche tels que la taille des données d'entrée ou le nombre de fichiers d'entrée dépassant les HealthLake quotas.
- Si le statut de la tâche d'importation est COMPLETED WITH _ _ERRORS, consultez le fichier manifeste, Manifest.json, pour savoir quels fichiers n'ont pas été importés correctement.
- Si le statut de la tâche d'importation est le même FAILED et qu'aucun message n'est présent, rendez-vous à l'emplacement de sortie de la tâche pour accéder au fichier manifeste, Manifest.json.

Pour chaque fichier d'entrée, il existe un fichier de sortie d'échec avec le nom du fichier d'entrée pour toute ressource dont l'importation échoue. Les réponses contiennent un numéro de ligne (lineId) correspondant à l'emplacement des données d'entrée, à l'objet de FHIR réponse (UpdateResourceResponse) et au code d'état (statusCode) de la réponse.

Un exemple de fichier de sortie ressemblerait à ce qui suit :

```
{"lineId":3, UpdateResourceResponse:{"jsonBlob":
{"resourceType":"OperationOutcome","issue":
[{"severity":"error","code":"processing","diagnostics":"1 validation error detected:
 Value 'Patient123' at 'resourceType' failed to satisfy constraint: Member must satisfy
 regular expression pattern: [A-Za-z]{1,256}"}]}, "statusCode":400}
{"lineId":5, UpdateResourceResponse:{"jsonBlob":
{"resourceType":"OperationOutcome","issue":
[{"severity":"error","code":"processing","diagnostics":"This property must be an
 simple value, not a com.google.gson.JsonArray","location":["/EffectEvidenceSynthesis/
name"]},{"severity":"error","code":"processing","diagnostics":"Unrecognised
 property '@telecom'","location":["/EffectEvidenceSynthesis"]},
{"severity":"error","code":"processing","diagnostics":"Unrecognised
```

```
 property '@gender'","location":["/EffectEvidenceSynthesis"]},
{"severity":"error","code":"processing","diagnostics":"Unrecognised
 property '@birthDate'","location":["/EffectEvidenceSynthesis"]},
{"severity":"error","code":"processing","diagnostics":"Unrecognised
 property '@address'","location":["/EffectEvidenceSynthesis"]},
{"severity":"error","code":"processing","diagnostics":"Unrecognised
 property '@maritalStatus'","location":["/EffectEvidenceSynthesis"]},
{"severity":"error","code":"processing","diagnostics":"Unrecognised
 property '@multipleBirthBoolean'","location":["/EffectEvidenceSynthesis"]},
{"severity":"error","code":"processing","diagnostics":"Unrecognised
 property '@communication'","location":["/EffectEvidenceSynthesis"]},
{"severity":"warning","code":"processing","diagnostics":"Name should be usable as an
 identifier for the module by machine processing applications such as code generation
 [name.matches('[A-Z]([A-Za-z0-9_]){0,254}')]","location":["EffectEvidenceSynthesis"]},
{"severity":"error","code":"processing","diagnostics":"Profile http://hl7.org/fhir/
StructureDefinition/EffectEvidenceSynthesis, Element 'EffectEvidenceSynthesis.status':
 minimum required = 1, but only found 0","location":["EffectEvidenceSynthesis"]},
{"severity":"error","code":"processing","diagnostics":"Profile
 http://hl7.org/fhir/StructureDefinition/EffectEvidenceSynthesis,
 Element 'EffectEvidenceSynthesis.population': minimum required
 = 1, but only found 0","location":["EffectEvidenceSynthesis"]},
{"severity":"error","code":"processing","diagnostics":"Profile
 http://hl7.org/fhir/StructureDefinition/EffectEvidenceSynthesis,
 Element 'EffectEvidenceSynthesis.exposure': minimum required =
 1, but only found 0","location":["EffectEvidenceSynthesis"]},
{"severity":"error","code":"processing","diagnostics":"Profile http://
hl7.org/fhir/StructureDefinition/EffectEvidenceSynthesis, Element
 'EffectEvidenceSynthesis.exposureAlternative': minimum required
 = 1, but only found 0","location":["EffectEvidenceSynthesis"]},
{"severity":"error","code":"processing","diagnostics":"Profile http://hl7.org/fhir/
StructureDefinition/EffectEvidenceSynthesis, Element 'EffectEvidenceSynthesis.outcome':
 minimum required = 1, but only found 0","location":["EffectEvidenceSynthesis"]},
{"severity":"information","code":"processing","diagnostics":"Unknown
 extension http://synthetichealth.github.io/synthea/disability-adjusted-
life-years","location":["EffectEvidenceSynthesis.extension[3]"]},
{"severity":"information","code":"processing","diagnostics":"Unknown extension
 http://synthetichealth.github.io/synthea/quality-adjusted-life-years","location":
["EffectEvidenceSynthesis.extension[4]"]}]}, "statusCode":400}
{"lineId":7, UpdateResourceResponse:{"jsonBlob":
{"resourceType":"OperationOutcome","issue":
[{"severity":"error","code":"processing","diagnostics":"2 validation errors detected:
 Value at 'resourceId' failed to satisfy constraint: Member must satisfy regular
 expression pattern: [A-Za-z0-9-.]{1,64}; Value at 'resourceId' failed to satisfy
 constraint: Member must have length greater than or equal to 1"}]}, "statusCode":400}
```

```
{"lineId":9, UpdateResourceResponse:{"jsonBlob":
{"resourceType":"OperationOutcome","issue":
[{"severity":"error","code":"processing","diagnostics":"Missing required id field in
 resource json"}]}, "statusCode":400}
{"lineId":15, UpdateResourceResponse:{"jsonBlob":
{"resourceType":"OperationOutcome","issue":
[{"severity":"error","code":"processing","diagnostics":"Invalid JSON found in input
 file"}]}, "statusCode":400}
```

L'exemple montre qu'il y a eu des défaillances sur les lignes 3, 4, 7, 9, 15 à partir des lignes d'entrée correspondantes du fichier d'entrée. Pour chacune de ces lignes, les explications sont les suivantes :

- Sur la ligne 3, la réponse explique que le contenu resourceType fourni dans la ligne 3 du fichier d'entrée n'est pas valide.
- Sur la ligne 5, la réponse explique qu'il y a une erreur de FHIR validation dans la ligne 5 du fichier d'entrée.
- À la ligne 7, la réponse explique qu'il existe un problème de validation avec la resourceId valeur fournie en entrée.
- Sur la ligne 9, la réponse explique que le fichier d'entrée doit contenir un identifiant de ressource valide.
- À la ligne 15, la réponse du fichier d'entrée est que le JSON format du fichier n'est pas valide.

Comment puis-je trouver les DocumentReference ressources qui n'ont pas pu être traitées ?

Si une DocumentReference ressource n'est pas valide, HealthLake fournira une extension indiquant une erreur de validation au lieu de la NLP sortie médicale intégrée. Afin de trouver les DocumentReference ressources ayant entraîné une erreur de validation lors du NLP traitement, les clients peuvent utiliser HealthLake la fonction de recherche avec la clé de recherche cm-decoration-statuset la valeur de recherche VALIDATION_ ERROR. Cette recherche répertorie toutes les DocumentReference ressources ayant entraîné des erreurs de validation, ainsi qu'un message d'erreur décrivant la nature de l'erreur. La structure du champ d'extension dans les DocumentReference ressources présentant des erreurs de validation ressemblera à l'exemple suivant.

```
"extension": [
          {
              "extension": [
                  {
                      "url": "http://healthlake.amazonaws.com/aws-cm/status/",
                      "valueString": "VALIDATION_ERROR"
                  },
                  {
                      "url": "http://healthlake.amazonaws.com/aws-cm/message/",
                      "valueString": "Resource led to too many nested objects after NLP
 operation processed the document. 10937 nested objects exceeds the limit of 10000."
                  }
              ],
              "url": "http://healthlake.amazonaws.com/aws-cm/"
          }
    ]
```

Un VALIDATION_ ERROR peut également apparaître si NLP la décoration crée plus de 10 000 objets imbriqués. Dans ce cas, le document doit être scindé en documents plus petits avant d'être traité.

Migration d'un magasin de données existant pour utiliser Amazon Athena

les magasins de données créés avant le 14 novembre 2022 sont fonctionnels, mais ne peuvent pas être interrogés dans Athena à l'aide de. SQL Pour interroger un magasin de données préexistant avec Athena, vous devez d'abord le migrer vers un nouveau magasin de données.

Pour migrer des données vers un nouveau magasin de données

1. Créez un nouveau magasin de données.
2. Exportez les données du compartiment préexistant vers un compartiment Amazon S3.
3. Importez les données dans le nouveau magasin de données depuis le compartiment Amazon S3.

L'exportation de données vers un compartiment Amazon S3 entraîne des frais supplémentaires. Les frais supplémentaires dépendent de la taille des données que vous exportez.

Connecter les résultats de recherche d'Athena à d'autres services AWS

Vous pouvez rencontrer des problèmes lorsque vous partagez les résultats de recherche d'Athena avec d'autres AWS services.

Un problème peut survenir lorsque vous l'utilisez dans `json_extract[1]` le cadre d'une requête SQL de recherche.

Pour résoudre ce problème, vous devez effectuer une mise à jour vers`CATVAR`.

Vous pouvez rencontrer ce problème lorsque vous essayez de créer des résultats de sauvegarde, une table (statique) ou une vue (dynamique).

La console Athena ne fonctionne pas après l'importation de données dans un nouveau magasin de données

Une fois que vous avez importé des données dans un nouveau magasin de données, il est possible que celles-ci ne soient pas immédiatement disponibles pour utilisation. Cela permet de laisser le temps aux données d'être ingérées dans les tables des icebergs. Réessayez ultérieurement.

Pourquoi est-ce que je reçois un message d'erreur relatif aux autorisations de Lake Formation : lakeformation : PutDataLakeSettings lors de l'ajout d'un nouvel administrateur de lac de données ?

Si votre IAM utilisateur ou votre rôle contient la politique `AWSLakeFormationDataAdmin` AWS gérée, vous ne pouvez pas ajouter de nouveaux administrateurs de data lake. Vous recevrez un message d'erreur contenant les informations suivantes :

```
User arn:aws:sts::111122223333:assumed-role/lakeformation-admin-user is not authorized
 to perform: lakeformation:PutDataLakeSettings on resource: arn:aws:lakeformation:us-
east-2:111122223333:catalog:111122223333 with an explicit deny in an identity-based
 policy
```

La politique AWS gérée `AdministratorAccess` est requise pour ajouter un IAM utilisateur ou un rôle en tant qu'administrateur AWS du lac de données de Lake Formation. Si votre IAM utilisateur ou votre rôle `AWSLakeFormationDataAdmin` le contient également, l'action échouera. La politique `AWSLakeFormationDataAdmin` AWS gérée contient un refus explicite de l'APIopération AWS Lake Formation,`PutDataLakeSetting`.

Même les administrateurs disposant d'un accès complet à l'AWSutilisation de la politique `AdministratorAccess` AWS gérée peuvent être limités par cette `AWSLakeFormationDataAdmin` dernière.

Comment activer la fonction intégrée HealthLake de traitement du langage naturel ?

Le 20 février 2023, le comportement par défaut des magasins de HealthLake données a changé.

Magasins de données actuels : tous les magasins de HealthLake données actuels cesseront d'utiliser le traitement du langage naturel (NLP) sur les ressources codées en base64`DocumentReference`. Cela signifie que les nouvelles `DocumentReference` ressources ne seront pas analysées à l'aide NLP du texte du type de ressource et qu'aucune nouvelle ressource ne sera générée à partir du texte contenu dans le type de `DocumentReference` ressource. Pour les `DocumentReference` ressources existantes, les données et les ressources générées via le système sont NLP conservées, mais elles ne seront pas mises à jour après le 20 février 2023.

Nouveaux magasins de données : les magasins de HealthLake données créés après le 20 février 2023 n'effectueront pas de traitement du langage naturel (NLP) sur les ressources codées en base64`DocumentReference`.

Pour activer cette fonctionnalité, vous devez créer un boîtier à l'aide de AWS Support Center Console. Pour créer votre dossier, connectez-vous à votre dossier Compte AWS, puis choisissez Créer un dossier. Pour en savoir plus sur la création d'un dossier et sur la gestion des dossiers, voir Création de dossiers d'assistance et gestion de cas dans le Guide de Support l'utilisateur.

Le statut de mon magasin de données ne change pas depuis Création

Si vous essayez de créer un nouveau magasin de HealthLake données et que le statut de votre magasin de données ne change pas depuis Création, vous devez mettre Athéna à jour pour qu'elle utilise le. AWS Glue Data Catalog

Pour en savoir plus, consultez la section Mise à niveau vers le catalogue de données AWS Glue step-by-step dans le guide de l'utilisateur Amazon Athena.

Une fois la mise à niveau réussie de AWS Glue Data Catalog, vous pouvez désormais créer un magasin de données.

Pour supprimer l'ancienne banque de données, commencez par créer un dossier à l'aide de AWS Support Center Console. Pour créer votre dossier, connectez-vous à votre dossier Compte AWS, puis choisissez Créer un dossier. Pour en savoir plus, consultez les sections Création de demandes d'assistance et gestion de dossiers dans le Guide de Support l'utilisateur.

Le statut de création de mon magasin de SDK données renvoie une exception ou un statut inconnu

Veuillez mettre à jour votre version SDK vers la dernière version si vos API appels de répertoire de données de liste ou de description de magasin de données renvoient une exception ou un statut de magasin de données inconnu.

Mon FHIR POST API opération avec un document de 10 Mo génère une erreur 413Request Entity Too Large. HealthLake

AWS HealthLake dispose d'une API limite de création et de mise à jour synchrones de 5 Mo afin d'éviter des latences et des délais d'attente accrus.

Vous pouvez ingérer des documents volumineux, jusqu'à 164 Mo, en utilisant le binaire à l'aide de l'importation ResourceType en bloc. API

Historique du document pour le guide du AWS HealthLake développeur

Le tableau suivant décrit les modifications apportées à la documentation pour les AWS HealthLake versions.

- APIversion : dernière
- Dernière mise à jour de la documentation : 25/10/2024

Modification	Description	Date
HealthLake désormais des supports FHIR `history` et des `vread` interactions	HealthLake prend désormais en charge l'FHIR`history`interaction pour récupérer l'historique d'une ressource particulière et l'`vread`interaction pour effectuer une lecture d'une ressource spécifique à la version.	25 octobre 2024
HealthLake prend désormais en charge les nouveaux paramètres FHIR de recherche, extensions et types de ressources.	HealthLake prend désormais en charge les nouveaux paramètres FHIR de recherche, extensions et types de ressources.	9 décembre 2023
HealthLake prend désormais en charge le FHIR framework SMART on	HealthLake prend désormais en charge la création SMART sur HealthLake des magasins de données FHIR activés.	31 mai 2023
HealthLake prend désormais en charge la validation des profils	HealthLake prend désormais en charge la validation des FHIR profils.	31 mai 2023

HealthLake prend désormais en charge `export`	HealthLake prend désormais en charge l'exportation de fichiers à l'aide de FHIR REST API l'opération`export`.	31 mai 2023
Région Asie-Pacifique (Mumbai)	AWS HealthLake est désormais disponible dans la région Asie-Pacifique (Mumbai).	4 avril 2023
Traitement du langage naturel intégré désactivé	HealthLake a désactivé le traitement intégré du langage naturel (NLP) sur tous les magasins de données à compter du 20 février 2023.	20 février 2023
HealthLake s'intègre à Amazon Athena	Vous pouvez désormais utiliser Athena pour interroge r les banques de données créées après le 14 novembre 2022.	14 novembre 2022
Augmentation de la taille totale des tâches d'importation	La taille totale maximale de tous les fichiers d'une demande S tartFHIRImport Job est désormais de 500 Go.	3 octobre 2022
Support des offres groupées	HealthLake prend désormais en charge le type de ressource Bundle pour l'ingestion de plusieurs ressources.	5 août 2022
Quotas mis à jour pour CRUD les opérations dans HealthLak e	HealthLake prend désormais en charge des limites plus élevées pour les CRUD demandes.	14 juillet 2022

Inclure le support	HealthLake prend désormais en charge `_include` les requêtes dans le magasin de données.	14 juillet 2022
AWS HealthLake est désormais disponible pour tous	HealthLake est désormais disponible pour tous.	30 juillet 2020

AWS Glossaire

Pour la AWS terminologie la plus récente, consultez le AWS glossaire dans la Glossaire AWS référence.

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.