aws

Guía para desarrolladores

AWS HealthLake

AWS HealthLake: Guía para desarrolladores

Table of Contents

¿Qué es AWS HealthLake?

AWS HealthLake es un servicio HIPAA apto para la ingesta, el almacenamiento y el análisis de datos clínicos que utiliza la especificación Healthcare Interoperability FHIR (R4).

> **Note**
>
> Después del 20 de febrero de 2023, HealthLake los almacenes de datos no utilizan el procesamiento de lenguaje natural integrado (NLP) de forma predeterminada. Si está interesado en activar esta función en su almacén de datos, consulte el capítulo ¿Cómo activo la función HealthLake de procesamiento de lenguaje natural integrada? de solución de problemas.

Los datos de Salud suelen ser incompletos e inconsistentes. Además, suelen estar desestructurados y la información está contenida en notas clínicas, informes de laboratorio, reclamaciones de seguros, imágenes médicas, conversaciones grabadas y datos de series temporales (por ejemplo, EEG huellas cardíacas ECG o cerebrales).

Los proveedores de servicios de salud pueden utilizarlos HealthLake para almacenar, transformar, consultar y analizar datos en la AWS nube. Al utilizar las capacidades médicas HealthLake integradas de procesamiento del lenguaje natural (NLP), puede analizar textos clínicos no estructurados de diversas fuentes. HealthLake transforma los datos no estructurados mediante modelos de procesamiento de lenguaje natural y proporciona potentes capacidades de consulta y búsqueda. Se puede utilizar HealthLake para organizar, indexar y estructurar la información de los pacientes de forma segura, compatible y auditable.

HealthLake también está integrado con Amazon Athena y AWS Lake Formation. Puede utilizar esta integración para consultar su almacén de datos medianteSQL.

Ventajas de AWS HealthLake

Con AWS HealthLake, usted puede:

- Ingiera datos de salud de forma rápida y sencilla: puede importar de forma masiva archivos locales de Fast Healthcare Interoperability Resources (FHIR), que incluyen notas clínicas, informes de laboratorio, reclamaciones de seguros y más, a un bucket de Amazon Simple Storage

Service (Amazon S3). A continuación, puede utilizar los datos en aplicaciones o flujos de trabajo posteriores.

- Utilice las FHIR REST API operaciones: HealthLake admite el uso de las FHIR REST API operaciones para realizar CRUD (Create/Read/Update/Delete) operaciones en su almacén de datos. FHIRtambién se admite la búsqueda.
- Almacene sus datos en la AWS nube de forma segura y HIPAA apta para que puedan auditarse: puede almacenar los datos en este FHIR formato para poder consultarlos fácilmente. HealthLake crea una vista cronológica completa del historial médico de cada paciente y lo estructura en el formato estándar R4. FHIR
- Integración de Athena: HealthLake la integración con Athena significa que puede crear consultas SQL basadas potentes que puede utilizar para crear y guardar criterios de filtro complejos. Luego, puede usar estos datos en aplicaciones posteriores, como la SageMaker IA para entrenar un modelo de aprendizaje automático o Amazon QuickSight para crear paneles y visualizaciones de datos.
- Transforme datos no estructurados mediante modelos de aprendizaje automático (ML) especializados: HealthLake proporciona procesamiento médico del lenguaje natural integrado (NLP) con Amazon Comprehend Medical. Los datos textuales médicos sin procesar se transforman mediante modelos de aprendizaje automático especializados. Estos modelos se han entrenado para comprender y extraer información significativa a partir de datos de atención médica no estructurados. Con la medicina integradaNLP, puede extraer automáticamente datos de entidades (por ejemplo, procedimientos médicos y medicamentos), relaciones entre entidades (por ejemplo, un medicamento y su dosis) y rasgos de la entidad (por ejemplo, el resultado positivo o negativo de una prueba o la hora del procedimiento) de su texto médico. HealthLake a continuación, crea nuevos recursos en función de las características, el signo, el síntoma y la afección. Se añaden como nuevos tipos de condición, observación y MedicationStatement recurso.

HealthLake casos de uso

Se puede utilizar HealthLake para las siguientes aplicaciones sanitarias:

- Gestión de la salud de la población: HealthLake ayuda a las organizaciones sanitarias a analizar las tendencias, los resultados y los costos de la salud de la población. Esto ayuda a las organizaciones a identificar la intervención más adecuada para una población de pacientes y a elegir mejores opciones de gestión de la atención.

- Mejorar la calidad de la atención: HealthLake ayuda a los hospitales, las compañías de seguros de salud y las organizaciones de ciencias de la vida a cerrar brechas en la atención, mejorar la calidad de la atención y reducir los costos al recopilar una visión completa del historial médico del paciente.
- Optimización de la eficiencia hospitalaria: HealthLake ofrece a los hospitales herramientas clave de análisis y aprendizaje automático para mejorar la eficiencia y reducir los residuos hospitalarios.

Acceder HealthLake

Puede acceder HealthLake a través de AWS Management Console, AWS Command Line Interface (AWS CLI) o AWS SDKs.

1. AWS Management Console — Proporciona una interfaz web a la que puede acceder HealthLake.
2. AWS Command Line Interface (AWS CLI): proporciona comandos para un amplio conjunto de AWS servicios HealthLake, incluidos y es compatible con Windows, macOS y Linux. Para obtener más información sobre la instalación de AWS CLI, consulte AWS Command Line Interface.
3. AWS SDKs— AWS proporciona SDKs (kits de desarrollo de software) que consisten en bibliotecas y código de muestra para varios lenguajes de programación y plataformas (Java, Python, Ruby,. NET, iOS, Android, etc.). SDKsProporcionan una forma cómoda de crear un acceso programático a HealthLake y AWS. Para obtener más información, consulte AWSSDKpara Python.

HIPAAaptitud y seguridad de los datos

Este es un servicio HIPAA elegible. Para obtener más información sobre AWS la Ley de Portabilidad y Responsabilidad de los Seguros de Salud de los Estados Unidos de 1996 (HIPAA) y el uso de AWS los servicios para procesar, almacenar y transmitir información de salud protegida (PHI), consulte HIPAADescripción general.

Las conexiones que HealthLake contienen información de identificación personal (PII) deben estar cifradas. De forma predeterminada, se deben HealthLake utilizar todas las HTTPS conexionesTLS. HealthLake almacena el contenido cifrado de los clientes y funciona según el principio de responsabilidad AWS compartida.

Precios

Para obtener información sobre HealthLake los precios, consulta la página AWS HealthLake de precios. Para estimar mejor los posibles costes asociados a HealthLake ellos, puedes utilizar la calculadora de HealthLake precios.

Cómo AWS HealthLake funciona

AWS HealthLake crea un almacén de datos que almacena los registros de salud utilizando la especificación Healthcare Interoperability FHIR (R4). Con HealthLake, puede realizar las siguientes tareas.

> **Note**
>
> Después del 20 de febrero de 2023, HealthLake los almacenes de datos no utilizarán el procesamiento de lenguaje natural integrado (NLP) de forma predeterminada. Si está interesado en activar esta función en su almacén de datos, consulte el capítulo ¿Cómo activo la función HealthLake de procesamiento de lenguaje natural integrada? de solución de problemas.

- Cree, supervise y elimine un almacén de datos.
- Se utiliza `StartFHIRImportJob` para importar datos de salud de forma masiva desde un depósito de Amazon Simple Storage Service (Amazon S3) a un almacén de datos.
- Utilice las operaciones de creación, lectura, actualización y eliminación (CRUD) para gestionar los datos almacenados en su almacén de datos.
- Úselo SQL en Amazon Athena para consultar su almacén de datos.
- Utilice un HTTP cliente en las FHIR REST API operaciones para buscar en su almacén de datos.
- Permita que las operaciones de Amazon Comprehend API Medical busquen información médica en sus datos mediante el procesamiento de lenguaje natural NLP ().

Creación y supervisión de almacenes de datos

Con él HealthLake, puede crear y monitorear almacenes de datos que puedan almacenar datos de Fast Healthcare Interoperability Resources (FHIR).

Para crear un nuevo banco de datos, puede usar la consola C reateFHIRDatastore o la HealthLake consola. Para ver el estado de un banco de datos, utilice escribeFHIRDatastoreD. Para ver el estado de varios almacenes de datos activos, utilice istFHIRDatastoresL. Para eliminar un banco de datos, utilice eleteFHIRDatastoreD.

FHIRRESTAPIoperaciones

Puede utilizar las FHIR REST API operaciones para realizar las operaciones de creación, lectura, actualización y eliminación (CRUD) en su almacén de HealthLake datos. Para obtener más información sobre cómo HealthLake apoya las FHIR REST API operaciones, consulteUso de FHIR REST API interacciones con un almacén HealthLake de datos.

Generación automatizada de recursos a partir de extensiones FHIR DocumentReference de recursos

> **Note**
>
> Cuando cree un almacén de HealthLake datos y añada los datos que lo contenganDocumentReference, incurrirá en cargos en su AWS cuenta. Para obtener más información, consulta los AWS HealthLake precios.

HealthLake proporciona información NLP sobre los documentos que se encuentran en el tipo de DocumentReference recurso. Para analizar el texto, HealthLake utiliza las siguientes operaciones de Amazon Comprehend API Medical.

- DetectEntitiesV2: inspecciona el texto clínico en busca de diversas entidades médicas y devuelve información específica sobre ellas, como la categoría de la entidad, la ubicación y el puntaje de confianza.
- InferICD10CM: Inspecciona el texto clínico para detectar afecciones médicas como entidades incluidas en la historia clínica de un paciente y las vincula a los identificadores conceptuales normalizados de la base de conocimientos de los Centros para el Control y la ICD Prevención de Enfermedades (-10-CM).
- InferRxNorm: Inspecciona el texto clínico para detectar los medicamentos como entidades incluidas en la historia clínica de un paciente y enlaza con los identificadores conceptuales normalizados de la base de datos de la RxNorm Biblioteca Nacional de Medicina.

HealthLake analiza automáticamente los datos encontrados en el tipo DocumentReference de recurso cuando se agrega al almacén de datos. Los archivos de DocumentReference recursos originales permanecen inalterados. La información médica extraída se adjunta automáticamente

como extensiones FHIR compatibles. Para obtener más información sobre cómo NLP funciona HealthLake, consulte. Uso de la generación automática de recursos basada en el procesamiento del lenguaje natural (NLP) del tipo de FHIR DocumentReference recurso en AWS HealthLake

Busque con consultas SQL basadas

> **Note**
>
> En el caso de los almacenes de datos creados antes del 14 de noviembre de 2022, la búsqueda se limita a las FHIR REST API operaciones. Para utilizar consultas SQL basadas en los datos de su HealthLake banco de datos, consulteConsulte AWS HealthLake los almacenes de datos con SQL Amazon Athena.

Amazon Athena es un servicio de consultas sin servidorSQL. HealthLake Los almacenes de datos se incorporan a Athena como tablas de Apache Iceberg. Estas tablas están diseñadas para admitir conjuntos de datos analíticos de gran tamaño. En Athena, cada tipo de FHIR recurso se representa como una tabla. Con Athena, solo puede realizar `READ` solicitudes en su almacén de datos. Para obtener más información sobre la búsqueda SQL basada, consulteConsulte su almacén HealthLake de datos mediante SQL.

Busque con FHIR REST API operaciones

Puede buscar en los registros médicos almacenados en su almacén de datos especificando un tipo de recurso con parámetros de búsqueda compatibles o utilizando un identificador de recurso que se encuentra en el servidor, sin especificar el tipo de recurso. Para obtener más información sobre la búsqueda mediante las FHIR REST API operaciones, consulteUso de FHIR REST API interacciones con un almacén HealthLake de datos.

Acciones para la importación de datos

Úselo AWS HealthLake para importar sus archivos de forma masiva desde un bucket de Amazon S3. Utilice la consola o S tartFHIRImport Job para iniciar un trabajo de importación. Tras importar los archivos, puede utilizar D escribeFHIRImport Job para supervisar el estado del trabajo. Una vez finalizado el trabajo de importación, los datos pueden añadirse a Athena, transformarse o analizarse y utilizarse en aplicaciones posteriores.

Acciones para la exportación de datos

Úselo HealthLake para exportar sus archivos de forma masiva a un bucket de Amazon S3. Utilice la consola o S tartFHIRExport Job para iniciar un trabajo de exportación. Tras exportar los archivos, puede utilizar D escribeFHIRExport Job para supervisar el estado del trabajo y ver sus propiedades. Una vez finalizado el trabajo de exportación, puedes visualizar los datos mediante Amazon QuickSight o puedes acceder a ellos mediante otros AWS servicios.

AWS HealthLake FHIRvalidaciones de perfil compatibles

HealthLake es compatible con la especificación FHIR R4 básica. En la especificación R4 se incluyen los FHIR perfiles. Los perfiles se utilizan en un tipo de FHIR recurso para definir una definición de tipo de recurso más específica mediante restricciones y/o extensiones del tipo de recurso base. Por ejemplo, un FHIR perfil puede identificar campos obligatorios, como extensiones y conjuntos de valores. Un recurso puede admitir varios perfiles. Todos los almacenes HealthLake de datos admiten el uso FHIR de perfiles.

No es necesario añadir un FHIR perfil al añadir datos a un banco HealthLake de datos. Si no se especifica ningún FHIR perfil al agregar o actualizar un recurso, el recurso solo se valida con el esquema FHIR R4 base.

FHIRLos perfiles a los que se ajusta un recurso se incluyen en el recurso antes de que se ingiera. HealthLake HealthLake valida los FHIR perfiles especificados cuando se agregan al banco de datos HealthLake .

FHIRLos perfiles se especifican en una guía de implementación. HealthLake valida los FHIR perfiles definidos en las siguientes guías de implementación.

FHIRPerfiles compatibles con HealthLake

Nombre	Versi	Guía de implementación	Funcionalidad
Núcleo estadouni dense	3.1.1	http://hl7.org/fhir/us/core/STU3.1.1/	Predeterminado
Núcleo estadouni dense	4.0.0	https://hl7.org/fhir/us/core/STU4/index.html	Compatible
CARINBotón azul	1.1.0	http://hl7.org/fhir/us/carin-bb/STU1.1/	Predeterminado
CARINBotón azul	1.0.0	https://hl7.org/fhir/us/carin-bb/STU1/	Compatible
Da Vinci Payer Data Exchange	1.0.0	https://hl7.org/fhir/us/davinci-pdex/	Predeterminado

Nombre	Versi	Guía de implementación	Funcionalidad
Intercambio de registros de salud Da Vinci (HRex)	0.2.0	https://hl7.org/fhir/us/davinci-hrex/2020Sep/	Predeterminado
DaVinci PDEXPlan Net	1.1.0	https://hl7.org/fhir/us/davinci-pdex-plan-net/STU1.1/	Predeterminado
DaVinci PDEXPlanear Net	1.0.0	https://hl7.org/fhir/us/davinci-pdex-plan-net/STU1/	Compatible
DaVinci Formulario de medicamentos de EE. UU. Payer Data Exchange (PDex)	1.1.0	https://hl7.org/fhir/us/davinci-drug-formulary/STU1.1/	Predeterminado
DaVinci Formulario de medicamentos de EE. UU. Payer Data Exchange (PDex)	1.0.1	https://hl7.org/fhir/us/davinci-drug-formulary/STU1.0.1/	Compatible
Misión digital Ayushman Bharat de la Autoridad Nacional de Salud () ABDM	2.0	https://www.nrces.in/ndhm/fhir/r4/index.html	Predeterminado

Validar FHIR los perfiles especificados en un recurso

Para validar un FHIR perfil, agréguelo al `profile` elemento de recursos individuales utilizando el perfil URL designado en la guía de implementación.

FHIRLos perfiles se validan al agregar un nuevo recurso a su banco de datos. Para agregar un nuevo recurso, puede usar la API operación S tartFHIRImport Job, realizar una `POST` solicitud para agregar un nuevo recurso o `PUT` actualizar un recurso existente.

Example — Para ver a qué FHIR perfil se hace referencia en un recurso

El perfil URL se añade al `profile` elemento del par `"meta" : "profile"` clave-valor. Este recurso se ha truncado para mayor claridad.

```
            {
             "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 — Cómo hacer referencia a un perfil compatible no predeterminado FHIR

Para realizar la validación con un perfil no predeterminado compatible (p. ej. CarinBB 1.0.0): añada el perfil URL con la versión (separada por '|') y el perfil URL base en el elemento. `meta.profile` Este recurso de ejemplo se ha truncado para mayor claridad.

```
            {
             "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“
              ]
             }
            }
```

Tipos de datos precargados

HealthLake solo se admite SYNTHEA como tipo de datos precargado. Synthea es un generador de pacientes sintético que modela la historia clínica de los pacientes generados por modelos. Es un repositorio de Git de código abierto que permite HealthLake generar paquetes de recursos FHIR compatibles con R4 para que los usuarios puedan probar modelos sin utilizar datos reales de pacientes.

Los siguientes tipos de recursos están disponibles en los almacenes de datos precargados.

Tipos de recursos de Synthea compatibles

AllergyIntolerance	Ubicación
CarePlan	MedicationAdministration
CareTeam	MedicationRequest
Reclamación	Observación
Condición	Organización
Dispositivo	Paciente
DiagnosticReport	Práctico
Encuentro	PractitionerRole
ExplanationofBenefit	Procedimiento
ImagingStudy	Procedencia
Inmunización	

Configurar los permisos para empezar a usar AWS HealthLake

En este capítulo, utilizará el AWS Management Console para configurar los permisos necesarios para empezar a usar AWS HealthLake y crear un almacén de datos. Para configurar los permisos para crear un banco de datos, debe crear un IAM usuario o rol que sea administrador y HealthLake administrador del lago de datos. Convierte a este usuario en administrador de lagos de datos en AWS Lake Formation. El administrador del lago de datos concede a Lake Formation acceso a los recursos necesarios para utilizar Amazon Athena para consultar un almacén de datos.

Después de crear un almacén de datos HealthLake, puede configurar los permisos para importar archivos al almacén de datos o exportarlos. Para obtener información sobre cómo configurar los permisos para importar archivos, consulteConfiguración de permisos para trabajos de importación. Para obtener información sobre cómo configurar los permisos para exportar archivos, consulteConfiguración de permisos para trabajos de exportación.

Temas

- Inscríbase en una Cuenta de AWS
- Creación de un usuario con acceso administrativo
- Configure IAM el usuario o el rol que desee utilizar HealthLake (IAMadministrador)
- Agregue un usuario o rol como administrador de Data Lake en Lake Formation (IAMadministrador)

Inscríbase en una Cuenta de AWS

Si no tiene uno Cuenta de AWS, complete los siguientes pasos para crearlo.

Para suscribirte a una Cuenta de AWS

1. Abrir https://portal.aws.amazon.com/billing/registro.
2. Siga las instrucciones que se le indiquen.

   Parte del procedimiento de registro consiste en recibir una llamada telefónica e indicar un código de verificación en el teclado del teléfono.

   Cuando te registras en un Cuenta de AWS, Usuario raíz de la cuenta de AWSse crea un. El usuario raíz tendrá acceso a todos los Servicios de AWS y recursos de esa cuenta. Como

práctica recomendada de seguridad, asigne acceso administrativo a un usuario y utilice únicamente el usuario raíz para realizar tareas que requieren acceso de usuario raíz.

AWS te envía un correo electrónico de confirmación una vez finalizado el proceso de registro. En cualquier momento, puede ver la actividad de su cuenta actual y administrarla accediendo a https://aws.amazon.com/y seleccionando Mi cuenta.

Creación de un usuario con acceso administrativo

Después de crear un usuario administrativo Cuenta de AWS, asegúrelo Usuario raíz de la cuenta de AWS AWS IAM Identity Center, habilite y cree un usuario administrativo para no usar el usuario root en las tareas diarias.

Proteja su Usuario raíz de la cuenta de AWS

1. Inicie sesión AWS Management Consolecomo propietario de la cuenta seleccionando el usuario root e introduciendo su dirección de Cuenta de AWS correo electrónico. En la siguiente página, escriba su contraseña.

   Para obtener ayuda para iniciar sesión con el usuario raíz, consulte Iniciar sesión como usuario raíz en la Guía del usuario de AWS Sign-In .
2. Activa la autenticación multifactorial (MFA) para tu usuario root.

   Para obtener instrucciones, consulte Habilitar un MFA dispositivo virtual para el usuario Cuenta de AWS root (consola) en la Guía del IAM usuario.

Creación de un usuario con acceso administrativo

1. Habilite IAM Identity Center.

   Consulte las instrucciones en Activar AWS IAM Identity Center en la Guía del usuario de AWS IAM Identity Center .
2. En IAM Identity Center, conceda acceso administrativo a un usuario.

   Para ver un tutorial sobre cómo usar el Directorio de IAM Identity Center como fuente de identidad, consulte Configurar el acceso de los usuarios con la configuración predeterminada Directorio de IAM Identity Center en la Guía del AWS IAM Identity Center usuario.

Inicio de sesión como usuario con acceso de gestionador

- Para iniciar sesión con su usuario de IAM Identity Center, utilice el inicio de sesión URL que se envió a su dirección de correo electrónico cuando creó el usuario de IAM Identity Center.

  Para obtener ayuda para iniciar sesión con un usuario de IAM Identity Center, consulte Iniciar sesión en el portal de AWS acceso en la Guía del AWS Sign-In usuario.

Concesión de acceso a usuarios adicionales

1. En IAM Identity Center, cree un conjunto de permisos que siga la práctica recomendada de aplicar permisos con privilegios mínimos.

   Para conocer las instrucciones, consulte Create a permission set en la Guía del usuario de AWS IAM Identity Center .
2. Asigne usuarios a un grupo y, a continuación, asigne el acceso de inicio de sesión único al grupo.

   Para conocer las instrucciones, consulte Add groups en la Guía del usuario de AWS IAM Identity Center .

Configure IAM el usuario o el rol que desee utilizar HealthLake (IAMadministrador)

> Persona: IAM administrador
>
> Un usuario que puede crear IAM usuarios y roles, y puede añadir administradores de lagos de datos.

Los pasos de este tema los debe llevar a cabo un IAM administrador.

Para conectar su almacén de HealthLake datos a Athena, debe crear un IAM usuario o rol que sea administrador y administrador del lago de datos. HealthLake Este nuevo usuario o rol otorga acceso a los recursos que se encuentran en un almacén de datos a través de AWS Lake Formation y tiene la política `AmazonHealthLakeFullAccess` AWS administrada agregada a su usuario o rol.

> **Important**
>
> Un IAM usuario o rol que sea administrador de un lago de datos no puede crear nuevos administradores de lagos de datos. Para agregar un administrador de lago de datos adicional, debe usar un IAM usuario o rol al que se le haya otorgado `AdministratorAccess` acceso.

Para crear un administrador

1. Agregue la política **`AmazonHealthlakeFullAccess`** IAM AWS administrada a un usuario o rol de su organización.

   Si no está familiarizado con la creación de un IAM usuario, consulte Creación de un IAM usuario y descripción general de AWS IAM las políticas en la Guía del IAM usuario.
2. Conceda al IAM usuario o rol acceso a AWS Lake Formation.

   - Agregue la siguiente política IAM AWS administrada a un usuario o rol de su organización: **`AWSLakeFormationDataAdmin`**

     > **Note**
     >
     > La `AWSLakeFormationDataAdmin` política otorga acceso a todos los recursos de AWS Lake Formation. Le recomendamos que utilice siempre los permisos mínimos necesarios para realizar la tarea. Para obtener más información, consulte las prácticas IAM recomendadas en la Guía del IAM usuario.
3. Agregue la siguiente política en línea al usuario o rol. Para obtener más información, consulte las políticas integradas en la Guía del IAMusuario.

```
{
    "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": "*"
        }
    ]
}
```

Para obtener más información sobre la `AWSLakeFormationDataAdmin` política, consulte la referencia sobre personas y IAM permisos de Lake Formation en la Guía para desarrolladores de AWS Lake Formation.

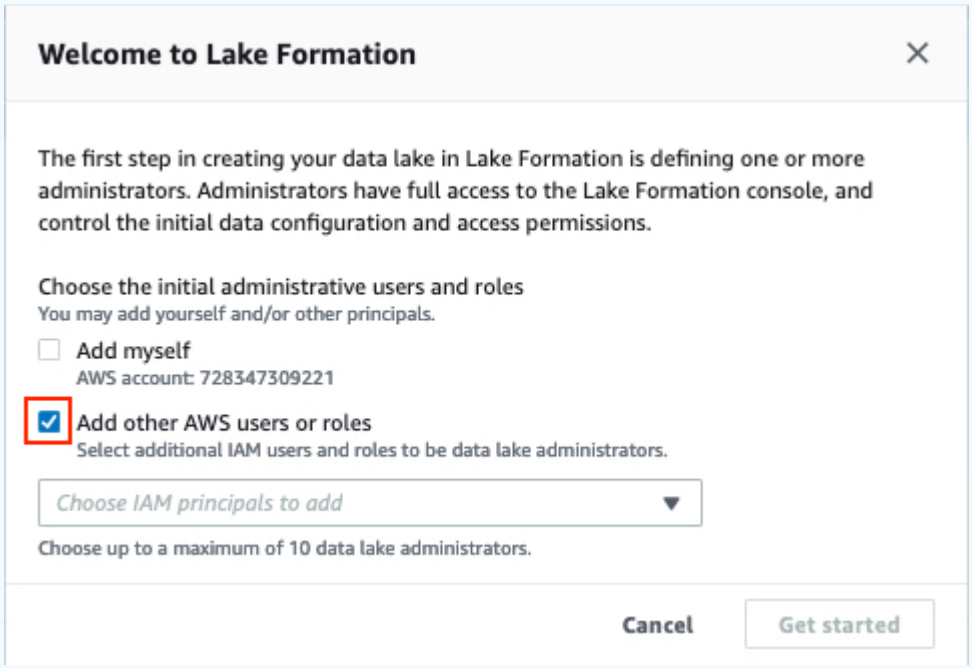
Agregue un usuario o rol como administrador de Data Lake en Lake Formation (IAMadministrador)

A continuación, el IAM administrador debe añadir el usuario o rol creado en el paso 1 como administrador del lago de datos en Lake Formation.

Para agregar un IAM usuario o un rol como administrador de un lago de datos

1. Abre la consola de AWS Lake Formation: https://console.aws.amazon.com/lakeformation/

> **Note**
>
> Si es la primera vez que visita Lake Formation, aparecerá el cuadro de diálogo Bienvenido a Lake Formation pidiéndole que defina un administrador de Lake Formation.



2. Asigne al nuevo usuario o rol como administrador AWS del lago de datos de Lake Formation.
   - Opción 1: Si ha recibido el cuadro de diálogo Welcome to Lake Formation.
     1. Seleccione Añadir otros AWS usuarios o roles.
     2. Seleccione la flecha hacia abajo (▼).
     3. Elija el HealthLake administrador que le gustaría que también fuera administrador de Lake Formation.
     4. Elija Comenzar.
   - Opción 2: utilice el panel de navegación (☰).
     1. Elija el panel de navegación (☰).
     2. En Permisos, elija Funciones y tareas administrativas.
     3. En la sección Administradores de Data Lake, selecciona Elegir administradores.
     4. En el cuadro de diálogo Administrar administradores de lagos de datos, seleccione la flecha hacia abajo (▼).
     5. A continuación, seleccione o busque los HealthLake administradores, usuarios o roles que también desee que sean administradores de Lake Formation.
     6. Seleccione Guardar.
3. Cambie la configuración de seguridad predeterminada para que la administre Lake Formation. Los recursos del almacén de HealthLake datos deben ser gestionados por Lake Formation,

noIAM. Para actualizar, consulte Cambiar el modelo de permisos predeterminado en la Guía para desarrolladores de AWS Lake Formation.

Crear un almacén de datos en AWS HealthLake

Una vez que haya terminadoConfigurar los permisos para empezar a usar AWS HealthLake, estará listo para crear un banco de datos. En AWS HealthLake, utiliza un banco de datos para almacenar datos en formato HL7 FHIR (R4). En los temas de este capítulo se describe cómo crear un banco de datos.

Tanto para crear almacenes de datos con capacidad de análisis como para conceder acceso a ellos en Athena, añada la política `AWSLakeFormationDataAdmin` gestionada a su IAM usuario, grupo o función. La `AWSLakeFormationDataAdmin` política le permite crear administradores de lagos de datos y conceder acceso a los almacenes de datos en Athena. Para obtener información sobre la configuración de permisos, consulteConfigurar los permisos para empezar a usar AWS HealthLake.

HealthLake también está integrado con AWS CloudTrail. Se puede utilizar CloudTrail para proporcionar un registro de las acciones realizadas por un usuario, un rol o un AWS servicio en HealthLake. CloudTrail captura todas las API llamadas y acciones de la consola HealthLake como eventos. Para obtener más información, consulte Registro de llamadas a la API de AWS HealthLake API con AWS CloudTrail.

Para obtener más información sobre los tipos de recursos de Fast Healthcare Interoperability Resources (FHIR) compatibles HealthLake, consulteTipos FHIR de recursos compatibles en AWS HealthLake.

> ⚠ Compatibilidad con Amazon Athena
>
> HealthLake los almacenes de fechas creados antes del 14 de noviembre de 2022 no pueden realizar SQL consultas con Athena. Para utilizar las funciones de búsqueda de Athena en su banco de datos preexistente, primero migre los datos a un nuevo banco de datos. Para obtener más información sobre la migración de almacenes de datos preexistentes, consulte. Migración de un almacén de datos existente para usar Amazon Athena

Tras crear un banco de datos, puede obtener sus propiedades, incluido su estado, con las operaciones API_DescribeFHIRDatastoreAPI_ListFHIRDatastoreso.html. API O bien, puede encontrar los estados de los almacenes de datos y otros detalles en la página de almacenes de datos de la HealthLake consola.

Un HealthLake banco de datos puede tener los siguientes estados:

- Creación: se está creando su banco de datos.
- Activo: su almacén de datos está activo. Puede importar y exportar datos desde él. También puede gestionar y buscar los FHIR recursos que ha almacenado en el almacén de datos.
- Eliminar: se está eliminando su almacén de datos.
- Eliminado: se ha eliminado su almacén de datos.

Temas

Crear un almacén de datos (AWS Management Console)

> ⚠ HealthLake diferencias entre consolas
>
> La HealthLake consola no admite la creación de un SMART almacén de datos FHIR activado. Para crear un SMART banco de datos FHIR activado, debe usar uno de los compatibles AWS CLI o uno de los AWS compatiblesSDKS. Para obtener más información, consulte SMARTIntegrándose FHIR con AWS HealthLake. Además, la consola no diferencia entre los dos tipos de almacenes de datos compatibles HealthLake cuando se consulta la página de detalles de un banco de datos individual.

Para crear un banco HealthLake de datos

1. Abre la HealthLake consola en https://console.aws.amazon.com//healthlake/casa.
2. Abre el panel de navegación (≡).
3. A continuación, elija Almacenes de datos.
4. A continuación, selecciona Crear almacén de datos.
5. En la sección de configuración del almacén de datos, en Nombre del almacén de datos, especifique un nombre.
6. (Opcional) En la sección de configuración del almacén de datos, en Precargar datos de muestra, active la casilla de verificación para precargar los datos de Synthea.
   - Los datos de Synthea son un conjunto de datos de muestra precargado. Para obtener más información, consulte Tipos de datos precargados.

7. En la sección de cifrado del almacén de datos, elija Usar una AWS clave propia (predeterminada) o Elegir una AWS KMS clave diferente (avanzada).
8. En la sección Etiquetas (opcional), puede añadir etiquetas a su almacén de datos.
   - Para obtener más información sobre cómo etiquetar el banco de datos, consulteAñadir una etiqueta a un almacén de datos.
9. A continuación, elija Crear almacén de datos. El estado de los almacenes de datos está disponible en la página Almacenes de datos.

Crear un almacén de datos (AWS CLI y AWS SDKs)

Puede usar los siguientes ejemplos de código para crear un banco HealthLake de datos.

AWS CLI

El siguiente ejemplo muestra el uso de la operación `CreateFHIRDatastore` en la AWS CLI. Para ejecutar el ejemplo, debe instalar AWS CLI. Al crear el almacén de datos, el cifrado en reposo se establece de forma predeterminada en una AWS KMS clave propia, a menos que se especifique lo contrario. Para obtener más información sobre el cifrado, REST HealthLake consulte. Cifrado en REST forma AWS HealthLake

El ejemplo está formateado para Unix, Linux y macOS. En Windows, sustituya la barra invertida (\) que continúa en Unix al final de cada línea por un signo de intercalación (). ^

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

Si tiene éxito, obtendrá la siguiente respuesta. JSON Cuando el almacén de datos esté listo para ingerir datos, el estado cambiará a`ACTIVE`. Para obtener más información sobre la importación de datos a su banco HealthLake de datos, consulteImportación de archivos a un almacén HealthLake de datos.

```
{
    "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/"
}
```

Para ver una lista de todos los almacenes de datos/almacenes de datos, puede utilizar la `ListFHIRDataStore` operación. También puede ver una lista de los almacenes de datos activos en la HealthLake consola.

Python (boto3)

En el siguiente ejemplo, se muestra cómo crear un HealthLake banco de datos mediante la `create_fhir_datastore` operación. Al crear el almacén de datos, el cifrado en reposo se establece de forma predeterminada en una AWS AWS KMS clave propia, a menos que se especifique lo contrario. Para obtener más información sobre el cifrado, REST HealthLake consulte. Cifrado en REST forma 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 almacén de datos puede tener uno de cuatro estados. Se utiliza `list_fhir_datastores` para ver una lista de sus almacenes de HealthLake datos, independientemente del estado. En este ejemplo se muestra cómo se puede filtrar en función del estado de un banco de datos.

```
import boto3

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

Para obtener más información, consulte `list_fhir_datastore`la documentación de Boto3.

Importación de archivos a un almacén HealthLake de datos

Una vez que haya terminadoCrear un almacén de datos en AWS HealthLake, podrá importar archivos al almacén de datos desde un bucket de Amazon Simple Storage Service (Amazon S3). Para importar archivos, inicie un trabajo de importación con la HealthLake consola o la `StartFHIRImportJob` API operación.

Cuando crea un trabajo de importación, especifica la ubicación de los datos de entrada en Amazon S3, una ubicación de bucket de Amazon S3 para los archivos de registro de salida, un IAM rol que otorga HealthLake acceso a sus buckets y una AWS Key Management Service clave propiedad o AWS propiedad del cliente. HealthLake utiliza esta clave para cifrar los datos en la ubicación de origen y se utilizará para descifrarlos y HealthLake permitir su importación. Para obtener información sobre cómo configurar los permisos para los trabajos de importación, consulte. Configuración de permisos para trabajos de importación Para obtener más información sobre la creación y el uso de AWS KMS claves, consulte Creación de claves en la Guía AWS para desarrolladores del Servicio de administración de claves.

HealthLake acepta archivos de entrada en formato newline delimited JSON (`.ndjson`), donde cada línea consta de un recurso válidoFHIR. Puede utilizar las API operaciones `ListFHIRImportJobs` para describir `DescribeFHIRImportJob` y enumerar los trabajos de importación en curso.

Para cada trabajo de importación, HealthLake genera un `manifest.json` archivo. Este registro describe los éxitos y los fracasos de un trabajo de importación. HealthLake envía el archivo al bucket de Amazon S3 que especifique al crear un trabajo de importación. Para obtener más información, consulte JSONArchivo de manifiesto.

Puede poner en cola los trabajos de importación o exportación. Estos trabajos de importación o exportación asíncronos se procesan según el método de procesamiento FIFO (primero en entrar, primero en salir). Puede crear, leer, actualizar o eliminar FHIR recursos mientras se está realizando un trabajo de importación o exportación.

Después de rellenar un banco de datos con datos precargados o importados, puede empezar a consultarlo en Amazon SQL Athena. Para obtener más información, consulte Consulte AWS HealthLake los almacenes de datos con SQL Amazon Athena.

Temas

- Configuración de permisos para trabajos de importación
- Iniciar un trabajo de importación en HealthLake

- JSONArchivo de manifiesto
- Ejemplo: iniciar y supervisar los trabajos de importación con AWS CLI

Configuración de permisos para trabajos de importación

Antes de importar archivos a un almacén de datos, debe conceder HealthLake permiso para acceder a los depósitos de entrada y salida en Amazon S3. Para conceder el HealthLake acceso, debe crear un rol de IAM servicio HealthLake, añadir una política de confianza al rol para conceder permisos de HealthLake asumir el rol y adjuntar una política de permisos al rol que le conceda acceso a sus buckets de Amazon S3.

Al crear un trabajo de importación, debe especificar el nombre de recurso de Amazon (ARN) de este rol para el`DataAccessRoleArn`. Para obtener más información sobre las IAM funciones y las políticas de confianza, consulte IAMFunciones.

Tras configurar el permiso, estará listo para importar archivos a su almacén de datos con un trabajo de importación. Para obtener más información, consulte Iniciar un trabajo de importación en HealthLake.

Para configurar los permisos de importación

1. Si aún no lo ha hecho, cree un bucket de Amazon S3 de destino para los archivos de registro de salida. El bucket de Amazon S3 debe estar en la misma AWS región que el servicio y bloquear el acceso público debe estar activado en todas las opciones. Para obtener más información, consulte Uso de Amazon S3 para bloquear el acceso público. También se debe usar una KMS clave propiedad de Amazon o propiedad del cliente para el cifrado. Para obtener más información sobre el uso de KMS claves, consulta Amazon Key Management Service.
2. Cree una función de servicio de acceso a datos HealthLake y dé permiso al HealthLake servicio para que la asuma con la siguiente política de confianza. HealthLake usa esto para escribir el bucket Amazon S3 de salida.

```
{
    "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. Añada una política de permisos a la función de acceso a los datos que le permita acceder al bucket de Amazon S3. `amzn-s3-demo-bucket`Sustitúyalo por el nombre de tu bucket.

```
{
    "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"
    }]
}
```

Iniciar un trabajo de importación en HealthLake

Tras crear un banco de datos y configurar los permisos para los trabajos de importación (Configuración de permisos para trabajos de importación), puede empezar a importar archivos con un trabajo de importación. Puede iniciar un trabajo de importación mediante la AWS HealthLake consola o la función de AWS HealthLake importaciónAPI, start-fhir-import-jobAPI.

Temas

- Importación de archivos con API operaciones
- Iniciar un trabajo de importación (consola)

Importación de archivos con API operaciones

Requisitos previos

Al utilizar las AWS HealthLake API operaciones, primero debe crear una política AWS Identity and Access Management (IAM) y asociarla a un IAM rol. Para obtener más información sobre IAM los roles y las políticas de confianza, consulta IAMPolíticas y permisos. Los clientes también deben usar una KMS clave para el cifrado. Para obtener más información sobre el uso de KMS Keys, consulta Amazon Key Management Service.

Para importar archivos (API), sigue los siguientes pasos.

1. Cargue sus datos en un bucket de Amazon S3.
2. Utilice la `start-fhir-import-job API` API operación. Al iniciar el trabajo, especifique el nombre del depósito de Amazon S3 que contiene los archivos de entrada, la KMS clave que quiere usar para el cifrado y la configuración de los datos de salida.
3. Para obtener más información sobre un trabajo de FHIR importación, utilice la describe-fhir-import-joboperación para obtener el identificador, el nombreARN, la hora de inicio, la hora de

finalización y el estado actual del trabajo. Se utiliza list-fhir-import-jobpara mostrar todos los trabajos de importación y sus estados.

Iniciar un trabajo de importación (consola)

Para importar archivos con la consola, debe cargar los datos en un bucket de Amazon S3,

Para importar archivos, siga los siguientes pasos.

1. Cargue sus datos en un bucket de Amazon S3.
2. Abre la HealthLake consola en https://console.aws.amazon.com//healthlake/casa.
3. Ve a la página de detalles del almacén de datos de tu banco de datos y selecciona Importar.
4. Especifique su bucket de Amazon S3 y cree o identifique el IAM rol y la KMS clave que quiere usar.
5. Elija Importar datos.

JSONArchivo de manifiesto

Para cada trabajo de importación, HealthLake genera un `manifest.json` archivo. HealthLake envía el archivo al bucket de Amazon S3 que especifique al crear un trabajo de importación.

El `manifest.json` archivo describe los éxitos y los fracasos de un trabajo de importación. Los archivos de registro se organizan en dos carpetas, denominadas `SUCCESS` y`FAILURE`. Un archivo de salida puede contener información confidencial, por lo que, al crear un trabajo de importación, debe proporcionar tanto un bucket de Amazon S3 de salida como una AWS KMS clave de cifrado.

El siguiente es un ejemplo del `manifest.json` archivo de salida. Se recomienda utilizar este archivo como primer paso para solucionar un error en un trabajo de importación. Proporciona detalles sobre cada archivo y los motivos por los que se produjo un error en el trabajo de importación.

```
  {
  "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
}
```

Ejemplo: iniciar y supervisar los trabajos de importación con AWS CLI

El siguiente ejemplo muestra cómo utilizar el AWS Command Line Interface para iniciar y supervisar un trabajo de importación. También puede utilizar la 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
```

Cuando comience el trabajo de importación, recibirá la siguiente confirmación.

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

Para supervisar el estado de un trabajo de importación o para conocer sus propiedades de configuración, utilice el comando describe-fhir-import-jobAPIo el AWS CLI comando, como se muestra en el siguiente ejemplo.

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

En respuesta, recibirá la siguiente información.

```
{
    "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)"
    }
}
```

Para ver una lista de todos los trabajos de importación, utilice el comando list-fhir-import-jobsAPIo el AWS CLI comando, como se muestra en el siguiente ejemplo. Puede añadir uno o más filtros para limitar los resultados.

```
    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)
```

En respuesta, recibirá la siguiente información.

```
{
    "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
```

Exportación de archivos de un HealthLake banco de datos

Tras crear un almacén de datos e importar los datos (o si utiliza datos de muestra precargados), puede exportar los datos a un bucket de Amazon S3. Para exportar los datos de su almacén de HealthLake datos, utilice las siguientes operaciones.

- Realice una solicitud de exportación mediante la `StartFHIRExportJob` API operación con las teclas AWS SDKs y HealthLake.
  - Esta operación solo permite realizar una solicitud de exportación en todo el sistema.
- Realice una solicitud de exportación utilizando la `export` sintaxis que utiliza. HealthLake FHIR REST API
  - Esta operación permite realizar solicitudes de exportación a todo el sistema, a pacientes y a grupos. También puede aplicar parámetros para filtrar aún más los datos de la solicitud de exportación.

> **Important**
>
> HealthLake SDKLas solicitudes de exportación mediante `StartFHIRExportJob` API la operación y las solicitudes de FHIR REST API exportación mediante `StartFHIRExportJobWithPost` API la operación tienen IAM acciones distintas. Los permisos de SDK autorización/denegación se pueden gestionar por `StartFHIRExportJobWithPost` separado para cada IAM acción (FHIRRESTAPIexportar con o exportar con). `StartFHIRExportJob` Si quieres restringir SDK tanto FHIR REST API las exportaciones como las exportaciones, asegúrate de denegar los permisos para cada IAM acción.

Ambas operaciones solo permiten exportar los archivos a un bucket de Amazon S3 (S3). Todos los archivos del almacén de HealthLake datos se exportan como archivos delimitados por nuevas líneas JSON (`.ndjson`), donde cada línea consta de un recurso válidoFHIR.

Ambas operaciones requieren una función de servicio. En él, HealthLake debe definirse como el servicio principal y usted debe definir un depósito de Amazon Simple Storage Service (S3) al que desea exportar sus archivos. Para obtener más información, consulte Configuración de permisos para trabajos de exportación.

Puede poner en cola los trabajos de importación o exportación. Estos trabajos de importación o exportación asíncronos se procesan según el método de procesamiento FIFO (primero en entrar, primero en salir). Puede crear, leer, actualizar o eliminar FHIR recursos mientras se está realizando un trabajo de importación o exportación.

Para exportar archivos del almacén de HealthLake datos, consulte las siguientes secciones.

- Configuración de permisos para trabajos de exportación
- Exportación de archivos desde su almacén de datos con la HealthLake consola o AWS SDKs
- Exportación de datos de su almacén de HealthLake datos con FHIR REST API operaciones

Configuración de permisos para trabajos de exportación

Antes de exportar archivos desde un almacén de datos, debe conceder HealthLake permiso para acceder a su bucket de salida en Amazon S3. Para conceder el HealthLake acceso, debe crear un rol de IAM servicio HealthLake, añadir una política de confianza al rol para conceder permisos de HealthLake asumir el rol y adjuntar una política de permisos al rol que le concede acceso a su bucket de Amazon S3.

Si ya has creado un rol para HealthLake inConfiguración de permisos para trabajos de importación, puedes reutilizarlo y concederle los permisos adicionales para tu bucket de exportación de Amazon S3 que se indican en este tema. Para obtener más información sobre IAM los roles y las políticas de confianza, consulta IAMPolíticas y permisos.

> **Important**
>
> HealthLake SDKlas solicitudes de exportación mediante `StartFHIRExportJob` API la operación y las solicitudes de FHIR REST API exportación mediante `StartFHIRExportJobWithPost` API la operación tienen IAM acciones distintas. Los permisos de SDK autorización/denegación se pueden gestionar por `StartFHIRExportJobWithPost` separado para cada IAM acción (FHIRRESTAPIexportar con o exportar con). `StartFHIRExportJob` Si quieres restringir SDK tanto FHIR REST API las exportaciones como las exportaciones, asegúrate de denegar los permisos para cada IAM acción. Si concedes a los usuarios acceso total HealthLake, no será necesario cambiar los permisos de los IAM usuarios.

El usuario o rol que configura los permisos debe tener permiso para crear roles, crear políticas y adjuntar políticas a los roles. La siguiente IAM política concede estos permisos.

```
{
  "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"
      }
    }
  }]
}
```

Para configurar los permisos de exportación

1. Si aún no lo ha hecho, cree un bucket de Amazon S3 de destino para los datos que va a exportar desde su almacén de datos. El bucket de Amazon S3 debe estar en la misma AWS región que el servicio y bloquear el acceso público debe estar activado en todas las opciones. Para obtener más información, consulte Uso de Amazon S3 para bloquear el acceso público. También se debe usar una KMS clave propiedad de Amazon o propiedad del cliente para el cifrado. Para obtener más información sobre el uso de KMS claves, consulta Amazon Key Management Service.
2. Si aún no lo has hecho, crea un rol de servicio de acceso a datos HealthLake y dale permiso al HealthLake servicio para que lo asuma con la siguiente política de confianza. HealthLake usa esto para escribir el bucket Amazon S3 de salida. Si ya has creado uno enConfiguración de permisos para trabajos de importación, puedes reutilizarlo y concederle permisos para tu bucket de Amazon S3 en el siguiente paso.

   ```
   {
       "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. Añada una política de permisos a la función de acceso a los datos que le permita acceder a su bucket de Amazon S3 de salida. `amzn-s3-demo-bucket`Sustitúyalo por el nombre del bucket.

```
{
    "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"
    }]
}
```

Exportación de archivos desde su almacén de datos con la HealthLake consola o AWS SDKs

Una vez que haya terminadoConfiguración de permisos para trabajos de exportación, podrá exportar los archivos de su almacén de datos a un bucket de Amazon Simple Storage Service (Amazon S3). Para exportar archivos de un almacén de datos, debe iniciar un trabajo de exportación en HealthLake. Un trabajo de exportación exporta los archivos del banco de datos en un formato delimitado por líneas nuevas JSON (`.ndjson`), en el que cada línea consta de un recurso válidoFHIR. Al iniciar un trabajo de exportación, debe especificar una AWS KMS clave de cifrado. Para obtener más información sobre cómo crear una KMS clave, consulte Creación de claves en la Guía AWS para desarrolladores del Servicio de administración de claves.

En los temas siguientes se explica cómo iniciar un trabajo de exportación con la AWS HealthLake consola y AWS SDKs con la start-fhir-export-jobAPIoperación.

Temas

- Exportación de archivos desde su almacén de datos (consola)
- Exportación de archivos del almacén de datos (AWS SDKs)

Exportación de archivos desde su almacén de datos (consola)

Para exportar archivos (consola), siga los siguientes pasos.

1. Cree un depósito S3 de salida en la misma región que HealthLake.
2. Para iniciar un nuevo trabajo de exportación, identifique el bucket de Amazon S3 de salida y cree o identifique el IAM rol que quiere usar. Para obtener más información sobre las IAM

funciones y las políticas de confianza, consulte las IAMfunciones. Utilice también un cifrado de KMS clave. Para obtener más información sobre el uso de KMS claves, consulta Amazon Key Management Service.

3. Para ver el estado de su trabajo de exportación, utilice `ListFHIRExportJobs` API la operación.

Exportación de archivos del almacén de datos (AWS SDKs)

Para exportar archivos del almacén de datos con la AWS SDKs, utilice la start-fhir-export-joboperación. El siguiente código muestra cómo iniciar un trabajo de exportación con el SDK para 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'])
```

Para obtener el identificador, el nombreARN, la hora de inicio, la hora de finalización y el estado actual de un trabajo de FHIR exportación, utilice. describe-fhir-export-job Se utiliza list-fhir-export-jobspara enumerar todos los trabajos de exportación y sus estados.

El siguiente código muestra cómo obtener las propiedades de un trabajo de exportación específico con Python (Boto3). SDK

```
import boto3

client = boto3.client('healthlake')
```

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

Exportación de datos de su almacén de HealthLake datos con FHIR REST API operaciones

Una vez finalizada la operaciónConfiguración de permisos para trabajos de exportación, puede exportar los datos del almacén de HealthLake datos con FHIR REST API operaciones. Para realizar una solicitud de exportación mediante el FHIR RESTAPI, debe tener un IAM usuario, grupo o rol con los permisos necesarios, especificarlo `$export` como parte de la `POST` solicitud e incluir los parámetros de la solicitud en el cuerpo de la solicitud. Según la FHIR especificación, el FHIR servidor debe admitir `GET` las solicitudes y puede `POST` admitirlas. Para admitir parámetros adicionales, se necesita un organismo que inicie la exportación y, por lo tanto, HealthLake admite POST las solicitudes.

> **Important**
>
> HealthLake Los almacenes de datos creados antes del 1 de junio de 2023 solo admiten solicitudes de trabajo de exportación FHIR REST API basadas en exportaciones a nivel de todo el sistema.
> HealthLake Los almacenes de datos creados antes del 1 de junio de 2023 no permiten obtener el estado de una exportación mediante una `GET` solicitud en el punto final de un almacén de datos.

Todas las solicitudes de exportación que realice mediante el FHIR REST API se devuelven en `ndjson` formato y se exportan a un bucket de Amazon S3. Cada objeto de S3 contendrá un solo tipo FHIR de recurso.

Puede poner en cola las solicitudes de exportación según las cuotas de la AWS cuenta. Para obtener más información sobre los Service Quotas asociados a HealthLake, consulteAWS HealthLake puntos finales y cuotas.

HealthLake admite los siguientes tres tipos de solicitudes de punto final de exportación masiva.

Tipo	Descripciones	Sintaxis
Exportación del sistema	Exporte todos los datos del HealthLake FHIR servidor.	`POST https://healthlake. your-region .amazonaws.com/datastore/ your-datastore-id /r4/$export`
Todos los pacientes	Exporte todos los datos relacionados con todos los pacientes, incluidos los tipos de recursos asociados al tipo de recurso del paciente.	`POST https://healthlake. your-region .amazonaws.com/datastore/ your-datastore-id /r4/Patient/$export`
Grupo de pacientes	Exporte todos los datos relacionados con un grupo de pacientes especificado con un identificador de grupo.	`POST https://healthlake. your-region .amazonaws.com/datastore/ your-datastore-id /r4/Group/ ID/$export`

Antes de empezar

Cumpla los siguientes requisitos para realizar una solicitud de exportación mediante FHIR REST API el formulario HealthLake.

- Debe haber configurado un usuario, grupo o rol que tenga los permisos necesarios para realizar la solicitud de exportación. Para obtener más información, consulte Autorizar una solicitud `export`.
- Debe haber creado un rol de servicio que conceda HealthLake acceso al bucket de Amazon S3 al que quiere que se exporten sus datos. El rol de servicio también debe especificarse HealthLake como principal de servicio. Para obtener más información sobre la configuración de permisos, consulteConfiguración de permisos para trabajos de exportación.

Autorizar una solicitud **`export`**

Para realizar correctamente una solicitud de exportación mediante el FHIR RESTAPI, autorice su usuario, grupo o rol mediante una de las siguientes opciones: IAM o OAuth2 .0. También debe tener un rol de servicio.

Autorizar una solicitud mediante IAM

Al realizar una `$export` solicitud, el usuario, el grupo o el rol deben tener `StartFHIRExportJobWithPost` la política y `CancelFHIRExportJobWithDelete` IAM las acciones incluidas en ella. `DescribeFHIRExportJobWithGet`

> **Important**
>
> HealthLake SDKlas solicitudes de exportación mediante `StartFHIRExportJob` API la operación y las solicitudes de FHIR REST API exportación mediante `StartFHIRExportJobWithPost` API la operación tienen IAM acciones distintas. Los permisos de SDK autorización/denegación se pueden gestionar por `StartFHIRExportJobWithPost` separado para cada IAM acción (FHIRRESTAPIexportar con o exportar con). `StartFHIRExportJob` Si quieres restringir SDK tanto FHIR REST API las exportaciones como las exportaciones, asegúrate de denegar los permisos para cada IAM acción.

Autorizar una solicitud mediante SMART on FHIR (OAuth2.0)

Cuando realiza una `$export` solicitud SMART en un almacén de HealthLake datos FHIR habilitado, debe tener asignados los ámbitos adecuados. Para obtener más información sobre los ámbitos compatibles, consulte. HealthLake alcances específicos de FHIR los recursos del almacén de datos

Realizar una solicitud **`export`**

En esta sección se describen los pasos necesarios que debe seguir al realizar una solicitud de exportación mediante el FHIR RESTAPI.

Para evitar cargos accidentales en tu AWS cuenta, te recomendamos probar tus solicitudes realizando una `POST` solicitud sin proporcionar la `export` sintaxis.

Para realizar la solicitud, debes hacer lo siguiente:

1. Especifique `export` en la `POST` URL solicitud un punto final compatible.
2. Especifique los parámetros de cabecera necesarios.
3. Especifique el cuerpo de la solicitud que defina los parámetros necesarios.

Paso 1: especifique **`export`** en la **`POST`** URL solicitud un punto final compatible

HealthLake admite tres tipos de solicitudes de puntos finales de exportación masiva. Para realizar una solicitud de exportación masiva, debe realizar una solicitud `POST` basada en uno de los tres puntos de enlace compatibles. Los siguientes ejemplos muestran cómo especificar `export` en la solicitudURL.

- `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`

En esa cadena de `POST` solicitud, puede utilizar los siguientes parámetros de búsqueda compatibles.

Parámetros de búsqueda compatibles

HealthLake admite los siguientes modificadores de búsqueda en las solicitudes de exportación masiva.

Estos ejemplos incluyen caracteres especiales que deben codificarse antes de enviar la solicitud.

Nombre	¿Obligatorio?	Descripción	Ejemplo
`_outputFormat`	No	El formato para generar los archivos de datos masivos solicitados. Los valores aceptados son `application/fhir+ndjson` , `application/ndjson` , `ndjson`.	
`_type`	No	Cadena de tipos de FHIR recursos	`&_type=MedicationS`

Nombre	¿Obligatorio?	Descripción	Ejemplo
		delimitados por comas que desea incluir en su trabajo de exportación. Recomendamos incluirlo `_type` porque esto puede tener repercusiones en los costes cuando se exportan todos los recursos.	`tatement, Observation`
`_since`	No	Tipos de recursos modificados en o después de la marca de fecha y hora. Si un tipo de recurso no tiene una hora de última actualiza ción, se incluirá en la respuesta.	`&_since=2 024-05-09 T00%3A00%3A00Z`

Paso 2: especifique los parámetros de cabecera necesarios

Para realizar una solicitud de exportación mediante el FHIR RESTAPI, debe especificar los dos parámetros de encabezado siguientes.

- Content-Type: `application/fhir+json`
- Prefiero: `respond-async`

A continuación, debe especificar los elementos necesarios en el cuerpo de la solicitud.

Paso 3: especifique un cuerpo de solicitud que defina los parámetros necesarios.

La solicitud de exportación también requiere un cuerpo en `JSON` formato. El cuerpo puede incluir los siguientes parámetros.

Clave	¿Obligatorio?	Descripción	Valor
DataAccessRoleArn	Sí	Una función ARN de HealthLake servicio. El rol de servicio utilizado debe especificarse HealthLake como principal de servicio.	`arn:aws:iam:: 444455556666 :role/your-healthlake-service-role`
JobName	No	El nombre de la solicitud de exportación.	**`your-export-job-name`**
S3Uri	Sí	Parte de una OutputDataConfig clave. El S3 URI del depósito de destino donde se descargarán los datos exportados.	`s3://DOC-EXAMPLE-DESTINATION-BUCKET/ EXPORT-JOB /`
KmsKeyId	Sí	Parte de una OutputDataConfig clave. La ARN AWS KMS clave utilizada para proteger el bucket de Amazon S3.	`arn:aws:kms: region-of-bucket:123456789012 :key/1234abcd-12ab-34cd-56ef-1234567890ab`

Example — Cuerpo de una solicitud de exportación realizada mediante el FHIR REST API

Para realizar una solicitud de exportación mediante el FHIR RESTAPI, debe especificar un cuerpo, como se muestra a continuación.

```
{
  "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"
    }
  }
}
```

Cuando su solicitud se realice correctamente, recibirá la siguiente respuesta.

Encabezado de respuesta

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

Cuerpo de respuesta

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

Gestionar tu solicitud de exportación

Tras realizar correctamente una solicitud de exportación, puedes `export` gestionarla describiendo el estado de una solicitud de exportación actual y `export` cancelando una solicitud de exportación actual.

Si cancela una solicitud de exportación mediante el RESTAPI, solo se le facturará la parte de los datos que se exportaron hasta el momento en que envió la solicitud de cancelación.

En los siguientes temas se describe cómo puede obtener el estado de una solicitud de exportación actual o cancelarla.

Cancelar una solicitud de exportación

Para cancelar una solicitud de exportación, haga una `DELETE` solicitud e introduzca el identificador de trabajo en la solicitudURL.

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

Cuando la solicitud se haya realizado correctamente, recibirá lo siguiente.

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

Si su solicitud no se acepta, recibirá lo siguiente.

```
{
  "resourceType": "OperationOutcome",
  "issue": [
    {
      "severity": "error",
      "code": "not-supported",
      "diagnostics": "Interaction not supported."
    }
  ]
}
```

Describir una solicitud de exportación

Para conocer el estado de una solicitud de exportación, haga una `GET` solicitud utilizando `export` y su**`export-request-job-id`**.

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

La JSON respuesta contendrá un `ExportJobProperties` objeto. Puede contener los siguientes pares clave-valor.

Nombre	¿Obligatorio?	Descripción	Valor
DataAccessRoleArn	No	Una ARN de las funciones de HealthLake servicio. El rol de servicio utilizado debe especificarse HealthLake como principal de servicio.	`arn:aws:iam:: 444455556666 :role/your-healthlake-service-role`
SubmitTime	No	Fecha y hora en que se envió un trabajo de exportación.	`Apr 21, 2023 5:58:02`
EndTime	No	La hora en que se completó un trabajo de exportación.	`Apr 21, 2023 6:00:08 PM`
JobName	No	El nombre de la solicitud de exportación.	**`your-export-job-name`**
JobStatus	No		Los valores válidos son: `SUBMITTED | IN_PROGRESS | COMPLETED_WITH_ERRORS | COMPLETED | FAILED`
S3Uri	Sí	Parte de un OutputDataConfigobjeto. El Amazon S3 URI del depósito de destino	`s3://DOC-EXAMPLE-DESTINATION-BUCKET/` **`EXPORT-JOB /`**

Nombre	¿Obligatorio?	Descripción	Valor
		donde se descargarán los datos exportados.	
KmsKeyId	Sí	Parte de un OutputDataConfigobjeto. La ARN AWS KMS clave utilizada para proteger el bucket de Amazon S3.	`arn:aws:kms: region-of-bucket:123456789012 :key/1234abcd-12ab-34cd-56ef-1234567890ab`

Example : Cuerpo de una solicitud de exportación descrita realizada mediante el FHIR REST API

Si se ejecuta correctamente, obtendrá la siguiente JSON respuesta.

```
{
  "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",
  }
}
```

Eliminar un almacén de datos en HealthLake

La eliminación de un almacén de datos es una operación asíncrona. Una vez iniciado, el estado cambia a Eliminar. Un almacén de datos mantiene el estado de Eliminación hasta que se eliminen también todos los FHIR datos del almacén de datos y la infraestructura subyacente necesaria.

Una vez que se eliminan los datos y la infraestructura, el estado del almacén de HealthLake datos cambia a Eliminado. Tras la eliminación, los detalles de los almacenes de datos solo estarán disponibles mediante las `ListFHIRDataStores` operaciones `DescribeFHIRDataStore` y durante siete días. Transcurridos siete días, el almacén de datos eliminado no aparecerá en los resultados.

Para eliminar correctamente un almacén de datos, el usuario, grupo o rol que realiza la solicitud debe tener la IAM acción `glue:DeleteDatabase` agregada a su IAM política. Esta IAM acción no se incluye como parte de la política AWS gestionada,`AmazonHealthLakeFullAccess`.

Puede eliminar un banco de datos con AWS Management Console AWS SDKs,, o AWS CLI.

Temas

- Eliminar un almacén de datos (consola)
- Eliminar un almacén de datos (AWS SDKsy AWS CLI)

Eliminar un almacén de datos (consola)

Para eliminar un almacén de datos con la consola, elija su almacén de datos en la página Almacenes de datos y elija eliminar.

Para eliminar un almacén HealthLake de datos

1. Abre la HealthLake consola en https://console.aws.amazon.com//healthlake/casa.
2. Abre el panel de navegación (≡).
3. A continuación, elija Almacenes de datos.
4. En la página Almacenes de datos, elige la opción situada junto al almacén de datos que deseas eliminar.
5. A continuación, selecciona Eliminar

6. En el cuadro de diálogo, escriba **`delete`** para confirmar que desea eliminar el banco de datos seleccionado.
7. A continuación, elija Eliminar. A continuación, el estado del banco de datos cambiará de Activo a Eliminado.

Eliminar un almacén de datos (AWS SDKsy AWS CLI)

Puede utilizar los ejemplos de código que aparecen a continuación para eliminar un HealthLake banco de datos.

AWS CLI

En los siguientes ejemplos se muestra el uso de la `DeleteFHIRDatastore` operación con AWS CLI. Para ejecutar el ejemplo, debe instalar AWS CLI.

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

Si se realiza correctamente, obtendrá la siguiente JSON respuesta.

```
{
    "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 SDKPara Python es compatible con el `describe_fhir_datastore` método que incluye un solo parámetro`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)
```

Si tiene éxito, devuelve un diccionario de 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}}
```

Para obtener detalles sobre más de un almacén de datos a la vez, utilice `ListFHIRDatastore`

utilice el `DeleteFHIRDataStore` comando AWS CLI como se muestra en el siguiente ejemplo. También puede eliminar un almacén de datos mediante la consola delete-fhir-datastore APIo la consola. Al eliminar un banco de datos, se eliminan todas las versiones de FHIR recursos contenidas en el almacén de datos y en la infraestructura subyacente. Los registros relacionados con un almacén de datos eliminado se conservan en la cuenta de servicio de acuerdo con HIPAA las directrices.

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

Como se muestra en el siguiente ejemplo de JSON respuesta, el estado cambia a `DELETING` "» para confirmar que el almacén de datos y su contenido están en proceso de borrarse.

```
{
  "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)"
}
```

Uso de FHIR REST API interacciones con un almacén HealthLake de datos

En AWS HealthLake, utiliza las REST API interacciones de Fast Healthcare Interoperability Resources (FHIR) para administrar y buscar FHIR recursos en su almacén de datos. FHIRRESTAPIlas interacciones se utilizan para realizar interacciones de creación, lectura, actualización y eliminación (CRUD) en los recursos de un almacén de datos. También puede crear cadenas de búsqueda complejas mediante una `POST` HTTP solicitud `GET` o, ya que HealthLake admite un subconjunto de operaciones FHIR de búsqueda compatibles.

Por motivos de conformidad, los tipos de FHIR recursos se validan de acuerdo con el HL7 FHIR recurso R4. StructureDefinition Para encontrar las capacidades FHIR relacionadas de un banco de HealthLake datos activo, realice una `GET` solicitud donde `metadata` se especifique en elURL, de la siguiente manera.

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

Si se ejecuta correctamente, recibirá un código de `200` HTTP respuesta y la declaración de capacidad de su almacén de HealthLake datos. Para obtener más información, consulte `CapabilityStatement`la documentación del HL7 FHIR R4.

En la siguiente tabla se enumeran las FHIR interacciones admitidas por AWS HealthLake.

FHIRinteracciones compatibles con AWS HealthLake

FHIRinteracción	Descripción
Interacciones de todo el sistema	
`capabilities`	Obtenga una declaración de capacidad del sistema
`batch/transaction`	Actualice, cree o elimine un conjunto de recursos en una sola interacción
Interacciones a nivel de tipo	
`create`	Cree un nuevo recurso con un ID asignado por el servidor
`search`	Busque un tipo de recurso en función de algunos criterios de filtro

FHIRinteracción	Descripción
`history`	Recupera el historial de cambios de un tipo de recurso concreto
Interacciones a nivel de instancia	
`read`	Lee el estado actual de un recurso
`history`	Lea el historial de cambios de un recurso en particular
`vread`	Lea el estado de una versión específica del recurso
`update`	Actualiza un recurso por su ID (o créalo si es nuevo)
`delete`	Elimine un recurso

Temas

- Tipos FHIR de recursos compatibles en AWS HealthLake
- Realizar operaciones de creación, lectura, actualización y eliminación (CRUD) en almacenes HealthLake de datos
- Buscar en el almacén de HealthLake datos mediante las FHIR REST API operaciones
- Lectura del historial FHIR de recursos
- Obtención de datos de pacientes con la operación Patient $everything FHIR REST API
- Exportación de datos de su almacén HealthLake de datos mediante $export

Tipos FHIR de recursos compatibles en AWS HealthLake

En la siguiente tabla se enumeran los tipos de recursos FHIR R4 compatibles AWS HealthLake con. Para obtener más información, consulte el índice de recursos en la documentación de HL7FHIRR4.

FHIRLos tipos de recursos R4 son compatibles con HealthLake

Cuenta	DetectedIssue	Factura	Práctico
ActivityDefinition	Dispositivo	Library	PractitionerRole
AdverseEvent	DeviceDefinition	Vinculación	Procedimiento

AllergyIntolerance	DeviceMetric	Enumeración	Procedencia
Cita	DeviceUseStatement	Ubicación	Cuestionario
AppointmentResponse	DeviceRequest	Medida	QuestionnaireResponse
AuditEvent- Ver nota	DiagnosticReport	MeasureReport	RelatedPerson
Binario	DocumentManifest	Medios	RequestGroup
BodyStructure	DocumentReference	¿Medicamento	ResearchStudy
Paquete: consulte la nota	EffectEvidenceSynthesis	MedicationAdministration	ResearchSubject
CapabilityStatement	¿Encuentro	MedicationDispense	RiskAssessment
CarePlan	Punto de conexión	MedicationKnowledge	RiskEvidenceSynthesis
CareTeam	EpisodeOfCare	MedicationRequest	Programación
ChargeItem	EnrollmentRequest	MedicationStatement	ServiceRequest
ChargeItemDefinition	EnrollmentResponse	MessageHeader	Slot
Reclamación	ExplanationOfBenefit	MolecularSequence	Ejemplar
ClaimResponse	FamilyMemberHistory	NutritionOrder	StructureDefinition
Comunicación	Indicador	Observación	StructureMap
CommunicationRequest	Objetivo	OperationOutcome	Substance
Composición	Grupo	Organización	SupplyDelivery
ConceptMap	GuidanceResponse	OrganizationAffiliation	SupplyRequest
Condición	HealthcareService	Parámetros	Tarea

Consentimiento	ImagingStudy	Paciente	ValueSet
Contrato	Inmunización	PaymentNotice	VisionPrescription
Cobertura	ImmunizationEvaluation	PaymentReconciliation	VerificationResult
CoverageEligibilityRequest	ImmunizationRecommendation	Persona	
CoverageEligibilityResponse	InsurancePlan	PlanDefinition	

⚠ FHIRespecificaciones y HealthLake

- No puede realizar `GET` ni `POST` solicitar con estos tipos de recursos: binarios OperationOutcome, agrupados y parámetros.
- AuditEvent— Se puede crear o leer un AuditEvent recurso, pero no se puede actualizar ni eliminar.
- Paquete: hay varias formas de HealthLake gestionar las solicitudes de paquetes. Para obtener más información, consulta Administrar varios FHIR recursos mediante Bundle.
- VerificationResult— Este tipo de recurso solo es compatible con los almacenes de datos creados después del 9 de diciembre de 2023.

Realizar operaciones de creación, lectura, actualización y eliminación (CRUD) en almacenes HealthLake de datos

Si bien utiliza AWS acciones nativas al administrar los almacenes de datos, importar datos y exportar datos, utiliza cuatro FHIR HTTP operaciones principales para crear (`POST`), leer (`GET`), actualizar (`PUT`) y eliminar (`DELETE`) FHIR los recursos de un banco de HealthLake datos. En los temas siguientes se describe cómo realizar las operaciones de creación, lectura, actualización y eliminación (CRUD) en el banco de HealthLake datos mediante los FHIR REST API servicios. Debe utilizar un proceso de firma de la versión 4 de Signature para autenticar HealthLake API las solicitudes

enviadas a través de un HTTP cliente. Para obtener más información, consulte el proceso de firma de la versión 4 de Signature en. Referencia general de AWS

Temas

- Crear un recurso con POST
- Leer un recurso con GET
- Actualización de un recurso mediante PUT
- Eliminar un recurso mediante DELETE
- Administrar varios FHIR recursos mediante Bundle

Crear un recurso con **POST**

Se utiliza una `POST` solicitud para crear un nuevo recurso en un banco HealthLake de datos. `POST`las solicitudes no requieren que proporcione un `id` elemento. Los HealthLake servidores devuelven un código de HTTP estado `201` Creado cuando un recurso se ha creado correctamente.

> **Note**
>
> Al realizar una `POST` solicitud sobre el tipo de `DocumentReference` recurso, las extensiones existentes no se modifican. En su lugar, AWS HealthLake agrega las nuevas extensiones junto con las existentes al banco de datos. Para obtener más información sobre cómo se HealthLake utiliza el procesamiento del lenguaje natural (NLP) en el tipo de `DocumentReference` recurso para extraer datos médicos valiosos, consulteUso de la generación automática de recursos basada en el procesamiento del lenguaje natural (NLP) del tipo de FHIR DocumentReference recurso en AWS HealthLake.

Example Crear un **`Patient`** recurso mediante una **`POST`** solicitud.

Para crear una `POST` solicitud de HealthLake banco de datos, utilice el punto final de su banco de datos y proporcione el cuerpo de la JSON solicitud. Para encontrar el punto final de un banco de datos, busque en la HealthLake consola, en Almacenes de datos, o utilice la escribeFHIRDatastore operación D de la AWS HealthLake APIReferencia.

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"
}
```

Respuesta de JSON

Para confirmar la creación del recurso para pacientes, recibirá un código de HTTP estado 201 Creado y la siguiente JSON respuesta.

```
{
  "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"
  }
}
```

Leer un recurso con **GET**

En este ejemplo, se muestra cómo leer un FHIR recurso para pacientes mediante una GET solicitud.

Example Leer un **Patient** recurso específico mediante una **GET** solicitud.

Para crear una GET solicitud HealthLake de almacén de datos, utilice el punto final de su almacén de datos. Para encontrar el punto final de un almacén de datos, busque en la HealthLake consola, en Almacenes de datos, o utilice la escribeFHIRDatastore operación D de la AWS HealthLake APIReferencia.

También debe incluir el tipo de recurso **Patient** y un identificador válido**2de04858-ba65-44c1-8af1-f2fe69a977d9**.

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

Respuesta de JSON

Si se ejecuta correctamente, recibirá un código de 200 HTTP estado y la siguiente JSON respuesta.

```
{
    "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"
}
```

Actualización de un recurso mediante **PUT**

En el siguiente ejemplo, se muestra cómo PUT actualizar los detalles de un paciente en el tipo de FHIR recurso para pacientes. Además, al realizar una PUT solicitud sobre un recurso que aún no se ha creado, se creará una versión inicial.

Tu solicitud devolverá un código de `200` HTTP estado si el recurso se actualizó o devolverá un código de `201` HTTP estado si se creó un recurso nuevo.

> **Note**
>
> Al realizar una PUT solicitud sobre el tipo de `DocumentReference` recurso, las extensiones existentes no se modifican. En su lugar, AWS HealthLake agrega las nuevas extensiones junto con las existentes al banco de datos. Para obtener más información sobre cómo se HealthLake utiliza el procesamiento del lenguaje natural (NLP) en el tipo de `DocumentReference` recurso para extraer datos médicos valiosos, consulteUso de la generación automática de recursos basada en el procesamiento del lenguaje natural (NLP) del tipo de FHIR DocumentReference recurso en AWS HealthLake.

Example Actualizar un tipo de **`Patient`** recurso mediante una **`PUT`** solicitud

Al realizar una PUT solicitud, necesitará el punto final del banco de datos, el nombre del tipo de recurso que desea actualizar, un identificador y el cuerpo de la JSON solicitud.

Si se utiliza PUT para crear un recurso nuevo, se utilizará el identificador proporcionado para crear el nuevo recurso.

PUT Request

Ejemplo de estructura de una PUT solicitud válida:

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

JSON Request Body

Un ejemplo de JSON cuerpo utilizado para actualizar el recurso para pacientes especificado.

```
{
  "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"
}
```

Respuesta de JSON

Recibirá lo siguiente JSON en respuesta para confirmar el cambio:

```
{
 "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"
 }
}
```

Actualización condicional

La actualización condicional permite actualizar un recurso existente en función de algunos criterios de búsqueda de identificación, en lugar de hacerlo mediante una identificación lógica. Cuando el servidor procesa esta actualización, realiza una búsqueda utilizando sus capacidades de búsqueda estándar para el tipo de recurso, con el objetivo de resolver un único identificador lógico para esta solicitud.

La acción que lleve a cabo depende del número de coincidencias que se encuentren:

- No hay coincidencias ni se ha proporcionado ningún identificador en el cuerpo de la solicitud: el servidor crea el recurso.
- No hay coincidencias, se ha proporcionado un identificador y el recurso no existe aún con el identificador: el servidor trata la interacción como una interacción entre actualizar y crear.
- No hay coincidencias, se ha proporcionado un identificador y ya existe: el servidor rechaza la actualización con un `409 Conflict` error.
- Una coincidencia, no se ha proporcionado ningún identificador de recurso O (se ha proporcionado un identificador de recurso y coincide con el recurso encontrado): el servidor realiza la

actualización con el recurso coincidente como se indica anteriormente. Si el recurso se ha actualizado, el servidor SHALL devuelve un`200 OK`;

- Una coincidencia, el identificador del recurso se ha proporcionado pero no coincide con el recurso encontrado: el servidor devuelve un `409 Conflict` error que indica que la especificación del identificador de cliente era un problema, preferiblemente con un OperationOutcome
- Varias coincidencias: el servidor devuelve un `412 Precondition Failed` error que indica que los criterios del cliente no fueron lo suficientemente selectivos, preferiblemente con un OperationOutcome

Example — Actualice un recurso para pacientes cuyo nombre sea Peter, su fecha de nacimiento sea el 1 de enero de 2000 y el número de teléfono 1234567890:

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

Eliminar un recurso mediante **DELETE**

Para eliminar un recurso del almacén de HealthLake datos, debe realizar una `DELETE` HTTP solicitud.

Example Eliminar un tipo **`Patient`** de recurso específico mediante una **`DELETE`** solicitud.

Para crear una `DELETE` solicitud, utilice el punto final del almacén de datos. Para encontrar el punto final de un banco de datos, busque en la HealthLake consola, en Almacenes de datos, o utilice la escribeFHIRDatastore operación D que se encuentra en la AWS HealthLake APIReferencia.

También debe incluir el tipo de recurso y un identificador válido.

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

Respuesta de HTTP

Si se ejecuta correctamente, recibirá un código de `204` HTTP estado que confirmará que el recurso ya no está en el almacén de datos. Si se produce un error en una solicitud de eliminación, recibirá un código de HTTP estado de la serie 400 que indica el motivo del error en la `DELETE` solicitud.

Administrar varios FHIR recursos mediante Bundle

En la especificación HL7 FHIR R4, los paquetes son simplemente una colección de recursos. HealthLake admite la creación de un tipo de recurso de paquete en una FHIR REST API solicitud y el uso de una transacción de paquete para realizar múltiples CRUD operaciones en una sola FHIR REST API solicitud. En una transacción de paquete, debe especificar el tipo de paquete tal como se `batch` indica en la FHIR REST API solicitud.

Todas las solicitudes de paquetes se registran mediante AWS CloudTrail. Para obtener más información sobre el uso de CloudTrail with HealthLake, consulteRegistro de llamadas a la API de AWS HealthLake API con AWS CloudTrail.

HL7FHIRRecursos de R4 (externos)

- Para leer la especificación completa, consulte Tipo de recurso: paquete en el índice de FHIR documentación.
- Para obtener información sobre las interacciones por lotes mediante el FHIR RESTAPI, consulte Interacciones por lotes mediante el FHIR REST API en el índice de FHIR documentación.

En las siguientes secciones se describe cómo estructurar una FHIR REST API solicitud para crear un nuevo recurso de paquete o procesar los recursos de forma individual mediante transacciones de paquete.

> ⚠ Diferencias entre la HealthLake consola AWS CLI, la y la AWS SDKs
>
> La HealthLake consola solo admite operaciones de tipo paquete en las que el tipo de recurso de paquete se especifica en la FHIR REST API solicitudURL.

Realizar múltiples CRUD operaciones mediante FHIR paquetes

Si no se especifica ningún tipo de recurso en la solicitudURL, la FHIR REST API solicitud se analiza como transacciones individuales del almacén de datos. Se evalúa cada CRUD operación proporcionada en el JSON cuerpo y se devuelve un código de HTTP estado específico. HealthLake admite el tipo de paquete`batch`.

Para realizar varias CRUD operaciones en una sola FHIR REST API solicitud, haga lo siguiente:

La siguiente lista muestra las partes truncadas del cuerpo de una solicitud utilizadas en la solicitud de paquete FHIR RESTAPI. Para ver el cuerpo completo de la solicitud, consulte Crear una solicitud de paquete que incluya varias CRUD operaciones.

1. No especifiques ningún tipo de recurso en tu POST solicitud:

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

2. En el cuerpo de la solicitud, especifique el tipo de paquete como `"type": "batch"`
3. En el cuerpo de la solicitud, especifique los datos específicos del recurso para cada CRUD interacción, empezando por la `resource` clave.
4. Cada CRUD operación se especifica `request` en el cuerpo de la solicitud de la siguiente manera:

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

En la JSON respuesta, se obtiene un código de HTTP estado para cada CRUD operación especificada en la solicitud.

HealthLake limita las transacciones por paquetes

- Para obtener más información sobre los límites que se HealthLake imponen a los paquetes, consultaAWS HealthLake puntos finales y cuotas.

El siguiente es un ejemplo de una operación de paquete que contiene varias CRUD operaciones.

Example — Crear una solicitud de paquete que incluya varias CRUD operaciones.

Para realizar una FHIR REST API solicitud que realice varias CRUD operaciones, debe realizar una POST solicitud utilizando el punto de conexión del almacén de datos y proporcionar el cuerpo de la JSON solicitud.

Puede encontrar el punto final de su almacén de datos en la HealthLake consola, en Almacenes de datos, o bien mediante la escribeFHIRDatastore operación D de la AWS HealthLake APIReferencia.

POST Request

Realice una POST solicitud utilizando el punto final de su almacén de datos. Utilice la siguiente pestaña, Cuerpo de la JSON solicitud, para ver los elementos necesarios del cuerpo de la solicitud.

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

JSON Request Body

En el cuerpo de la solicitud, debes proporcionar los siguientes pares de clave y valor junto con cualquier otro dato FHIR específico del recurso sobre las solicitudes individuales. CRUD El primer ejemplo muestra un cuerpo de JSON solicitud truncado que resalta los elementos necesarios. El segundo ejemplo muestra el cuerpo completo de la JSON solicitud.

```
{
  "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"
      }
    }
  ]
}
```

Este es un ejemplo completo que muestra la creación de un nuevo `Patient` tipo `Medication` de recurso.

```
{
  "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"
      }
    }
  ]
}
```

Respuesta de JSON

Para confirmar la creación de los recursos especificados en la transacción de paquete de ejemplo, obtendrá el código de HTTP estado 201 Creado para cada CRUD operación incluida. Cuando se produce un error en una CRUD operación, se obtiene el HTTP estado de la serie 400, que indica el motivo del error en la solicitud individual.

```
{
  "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"
      }
    }
  ]
}
```

Agrupar los recursos como un tipo de recurso de paquete

Para crear un nuevo tipo de recurso de paquete, debe `Bundle` especificarlo en la FHIR REST API solicitud y proporcionar un JSON cuerpo válido que contenga los recursos que desea agrupar.

Cuando se especifica Bundle en la solicitudURL, el contenido del cuerpo de la JSON solicitud se guarda en el almacén de HealthLake datos tal cual. Por lo tanto, no se puede realizar ninguna CRUD operación en los tipos de recursos individuales. A los paquetes de este tipo se les asigna un único identificador de recurso nuevo. Como los recursos se guardan tal cual, no puede realizar `GET` ni `POST` solicitar recursos individuales guardados en el tipo de recurso Paquete.

> **Note**
>
> La especificación HL7 FHIR R4 también admite la agrupación de recursos mediante Grupo, Composición y Lista. Al crear estos tipos de recursos, los recursos individuales no se incluyen directamente. En su lugar, utilizan el `Reference` elemento para apuntar a los recursos individuales. Por lo tanto, el uso de estos tipos de recursos le permite modificar los recursos individuales que contienen.

Para crear un tipo de `Bundle` recurso, debe especificarlo en la `POST` solicitud y proporcionar JSON una lista de los recursos que desea incluir.

Example — Crear un recurso de paquete mediante una solicitud **`POST`**

Para crear un `bundle` recurso, haga lo siguiente

1. Formatee una FHIR REST API solicitud de la siguiente manera:

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

2. Proporcione un JSON cuerpo en el que se especifiquen los recursos que desea agrupar. Este ejemplo agrupa dos recursos para pacientes.

```
{
  "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"
          }
        ]
      }
    }
  ]
}
```

Buscar en el almacén de HealthLake datos mediante las FHIR REST API operaciones

HealthLake permite buscar en su almacén de datos mediante las REST API operaciones proporcionadas como parte del FHIR estándar. En esta sección, encontrará ejemplos de cómo realizar `GET` y `POST` solicitar varios tipos de recursos diferentes.

> **Note**
>
> Para consultas que involucren información de identificación personal (PII) o Información de Salud Protegida (PHI), se recomienda utilizar `POST` solicitudes. En una `POST` solicitud, PII o PHI se agrega como parte del cuerpo de la solicitud y se cifra durante el tránsito.

La FHIR especificación admite varios tipos de parámetros de búsqueda, pero solo HealthLake admite un subconjunto. Para obtener más información, consulte Tipos de parámetros de búsqueda compatibles y Parámetros de búsqueda avanzada compatibles con HealthLake.

Búsqueda en el banco de datos mediante las FHIR REST API operaciones.

Tipos de parámetros de búsqueda compatibles

En la siguiente tabla se muestran los tipos de parámetros de búsqueda admitidos en HealthLake.

Tipos de parámetros de búsqueda compatibles

Parámetro de búsqueda	Descripción
_id	ID del recurso (no completoURL)
_lastUpdated	Fecha de la última actualización. El servidor tiene discreción en cuanto a la precisión de los límites.
_tag	Busca por una etiqueta de recurso.
_perfil	Busca todos los recursos etiquetados con un perfil.

Parámetro de búsqueda	Descripción
_seguridad	Busque en las etiquetas de seguridad aplicadas a este recurso.
_fuente	Busque de dónde proviene el recurso.
_texto	Busque en la narrativa del recurso.
createdAt	Busque con una extensión personalizada - createdAt.

> **Note**
>
> Los siguientes parámetros de búsqueda solo se admiten en los almacenes de datos creados después del 9 de diciembre de 2023: _security, _source, _text,. createdAt

En la siguiente tabla se muestran ejemplos de cómo modificar las cadenas de consulta en función de los tipos de datos especificados para un tipo de recurso determinado. Para mayor claridad, los caracteres especiales de la columna de ejemplos no se han codificado. Para que la consulta se realice correctamente, asegúrese de que la cadena de consulta se haya codificado correctamente.

Tipos de parámetros de búsqueda	Detalles	Ejemplos

Número	Busca un valor numérico en un recurso específico. Se observan cifras significativas.
	
	El número de dígitos significativos es específico por valor de parámetro de búsqueda, excluyendo los ceros iniciales.
	
	Se permiten prefijos de comparación.
		`[parameter]=100`
		
		`[parameter]=1e2`
		
		`[parameter]=lt100`

Tipos de parámetros de búsqueda	Detalles	Ejemplos

Fecha/ DateTime	Busca una fecha u hora específica. El formato esperado es`yyyy-mm-ddThh:mm:ss[Z|(+|-)hh:mm]` , pero puede variar.
	
	Acepta los siguientes tipos de datos: `date`,`dateTime`,`instant`,`PeriodTiming`. Para obtener más información sobre el uso de estos tipos de datos en las búsquedas, consulte la fecha en el índice de FHIR documentación.
	
	Se permiten prefijos de comparación.
		`[parameter]=eq2013-01-14`
		
		`[parameter]=gt2013-01-14T10:00`
		
		`[parameter]=ne2013-01-14`

Cadena	Busca una secuencia de caracteres distinguiendo mayúsculas de minúsculas.
	
	Admite ambos `HumanName` `Address` tipos. Para obtener más información, consulte la entrada de tipos de HumanName datos y las entradas de tipos de `Address` datos en el índice de FHIR documentación.
	
	Se admite la búsqueda avanzada mediante `:text` modificadores.
		`[base]/Patient?given=eve`
		
		`[base]/Patient?given:contains=eve`

Tipos de parámetros de búsqueda	Detalles	Ejemplos
Token	Busca una close-to-exact coincidencia con una cadena de caracteres, a menudo comparada con un par de valores de códigos médicos. La distinción entre mayúsculas y minúsculas está vinculada al sistema de código utilizado al crear una consulta. Las consultas basadas en subsuposiciones pueden ayudar a reducir los problemas relacionados con la distinción entre mayúsculas y minúsculas. Para mayor claridad, no se ha codificado. `|`	`[parameter]=[system]|[code]` : Aquí `[system]` se refiere a un sistema de codificación y `[code]` se refiere al valor de código que se encuentra dentro de ese sistema específico. `[parameter]=[code]` : En este caso, la entrada coincidirá con un código o un sistema. `[parameter]=|[code]` : En este caso, la entrada coincidirá con un código y la propiedad del sistema no tiene ningún identificador.

Compuesto	Busca varios parámetros dentro de un mismo tipo de recurso mediante los modificadores `$` y la `,` operación.
	
	Se permiten los prefijos de comparación.
		`/Patient?language=FR,NL&language=EN`
		
		`Observation?component-code-value-quantity=http://loinc.org|8480-6$lt60`
		
		`[base]/Group?characteristic-value=gender$mixed`

Tipos de parámetros de búsqueda	Detalles	Ejemplos

Cantidad	Busca un número, un sistema y un código como valores. Se requiere un número, pero el sistema y el código son opcionales. Basado en el tipo de datos de cantidad. Para obtener más información, consulte la cantidad en el índice de FHIR documentación.
	
	Utiliza la siguiente sintaxis asumida `[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`

Referencia	Busca referencias a otros recursos.	`[base]/Observation?subject=Patient/23`
		
		`test`

URI	Busca una cadena de caracteres que identifique de forma inequívoca un recurso concreto.	`[base]/ValueSet?url=http://acme.org/fhir/ValueSet/123`
Especial	Búsquedas basadas en extensiones médicas NLP integradas.	

Parámetros de búsqueda avanzada compatibles con HealthLake

HealthLake admite los siguientes parámetros de búsqueda avanzada.

Nombre	Descripción	Ejemplo	Funcionalidad
`_includ`	Se utiliza para solicitar que se devuelvan recursos adicionales en una solicitud de búsqueda. Devuelve los recursos a los que hace referencia la instancia de recurso de destino.	`Encounter?_include=Encounter:subject`	
`_revinc de`	Se utiliza para solicitar que se devuelvan recursos adicionales en una solicitud de búsqueda. Devuelve los recursos que hacen referencia a la instancia de recurso principal.	`Patient?_id=` **`patient-identifier`** `&_revinclude=Encounter:patient`	
`_summar`	El resumen se puede utilizar para solicitar un subconjunto del recurso.	`Patient?_summary=text`	Se admiten los siguientes parámetros de resumen:`_summary=true` ,, `_summary=false` `_summary=text` ,`_summary=data` .
`_elemen`	Solicite que se devuelva un conjunto específico de elementos como parte de un recurso en los resultados de la búsqueda.	`Patient?_elements=identifier,active,link`	
`_total`	Devuelve el número de recursos que coinciden con los parámetros de búsqueda.	`Patient?_total=accurate`	Support_total=`accurate` ,`_total=none` .

Nombre	Descripción	Ejemplo	Funcionalidad
`_sort`	Indique el orden de clasificación de los resultados de búsqueda devueltos mediante una lista separada por comas. El `-` prefijo se puede usar para cualquier regla de clasificación de la lista separada por comas para indicar un orden descendente.	`Observation?_sort=status,-date`	Support ordenar por campos con tipos`Number, String, Quantity, Token, URI, Reference` . La clasificación por solo `Date` se admite en los almacenes de datos creados después del 9 de diciembre de 2023. Support hasta 5 reglas de clasificación.
`_count`	Controle cuántos recursos se devuelven por página del paquete de búsqueda.	`Patient?_count=100`	El tamaño máximo de página es 100.
`chainin`	Elementos de búsqueda de los recursos referenciados. `.`Dirige la búsqueda encadenada al elemento dentro del recurso referenciado.	`DiagnosticReport?subject:Patient.name=peter`	
`reverse chainin (_has)`	Busque un recurso en función de los elementos de los recursos que hacen referencia a él.	`Patient?_has:Observation:patient:code=1234-5`	

`_include`

Si `_include` se utiliza en una consulta de búsqueda, también se pueden devolver FHIR recursos específicos adicionales. Se usa `_include` para incluir recursos que están enlazados hacia adelante.

Example — Para localizar **`_include`** a los pacientes o al grupo de pacientes a los que se les ha diagnosticado tos

Debería buscar en el tipo de `Condition` recurso especificando el código de diagnóstico de la tos y, a continuación, `_include` especificando que también desea que se devuelva ese diagnóstico. `subject` En el tipo `Condition` de recurso, `subject` se refiere al tipo de recurso del paciente o al tipo de recurso del grupo.

Para mayor claridad, los caracteres especiales del ejemplo no se han codificado. Para que la consulta se realice correctamente, asegúrese de que la cadena de consulta se haya codificado correctamente.

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

`_revinclude`

Si `_revinclude` se utiliza en una consulta de búsqueda, también se pueden devolver FHIR recursos específicos adicionales. `_revinclude`Utilícelo para incluir recursos que estén enlazados hacia atrás.

Example — **`_revinclude`** Para incluir tipos de recursos relacionados con el encuentro y la observación vinculados a un paciente específico

Para realizar esta búsqueda, primero debe definir a la persona `Patient` especificando su identificador en el parámetro `_id` de búsqueda. Luego, especificaría FHIR recursos adicionales utilizando la estructura `Encounter:patient` y`Observation:patient`.

Para mayor claridad, los caracteres especiales del ejemplo no se han codificado. Para que la consulta se realice correctamente, asegúrese de que la cadena de consulta se haya codificado correctamente.

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

`_summary`

El uso `_summary` en una consulta de búsqueda permite al usuario solicitar un subconjunto del FHIR recurso. Puede contener uno de los siguientes valores:`true, text, data, false`. Cualquier

otro valor se considerará inválido. Los recursos devueltos se marcarán con una `'SUBSETTED'` meta.tag para indicar que los recursos están incompletos.

- `true`: Devuelve todos los elementos compatibles que estén marcados como «resumen» en la definición básica de los recursos.
- `text`: Devuelve solo los elementos «texto», «identificador» y «meta» y solo los elementos obligatorios de nivel superior.
- `data`: Devuelve todas las partes excepto el elemento «texto».
- `false`: Devuelve todas las partes de los recursos

En una sola solicitud de búsqueda, `_summary=text` no se puede combinar con `_include` los parámetros `_revinclude` de búsqueda.

Example — Obtenga el elemento «texto» de los recursos para pacientes en un almacén de datos.

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

`_elements`

El uso `_elements` en una consulta de búsqueda permite solicitar elementos FHIR de recursos específicos. Los recursos devueltos se marcarán con una `'SUBSETTED'` etiqueta meta para indicar que los recursos están incompletos.

El `_elements` parámetro consiste en una lista de nombres de elementos base separados por comas, como los elementos definidos en el nivel raíz del recurso. Solo se devolverán los elementos que estén en la lista. Si los valores de los `_elements` parámetros contienen elementos no válidos, el servidor los ignorará y devolverá los elementos obligatorios y los elementos válidos.

`_elements`no se aplicará a los recursos incluidos (recursos devueltos cuyo modo de búsqueda es`include`).

En una sola solicitud de búsqueda, `_elements` no se puede combinar con los parámetros `_summary` de búsqueda.

Example — Obtenga los elementos «identificadores», «activos» y «enlaces» de los recursos para pacientes en su HealthLake almacén de datos.

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

_total

Si se utiliza `_total` en una consulta de búsqueda, se devolverá el número de recursos que coinciden con los parámetros de búsqueda solicitados. HealthLake devolverá el número total de recursos coincidentes (recursos devueltos cuyo modo de búsqueda es`match`) en la respuesta `Bundle.total` de búsqueda.

`_total`admite los valores `accurate` de los `none` parámetros. `_total=estimate`no es compatible. Cualquier otro valor se considerará inválido. `_total`no se aplica a los recursos incluidos (recursos devueltos cuyo modo de búsqueda es`include`).

Example — Obtenga el número total de recursos para pacientes en un almacén de datos:

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

_sort

Al `_sort` utilizarlos en la consulta de búsqueda, se ordenan los resultados en un orden específico. Los resultados se ordenan según la lista de reglas de clasificación separadas por comas en orden de prioridad. Las reglas de clasificación deben ser parámetros de búsqueda válidos. Los demás valores se considerarán inválidos.

En una sola solicitud de búsqueda, puede utilizar hasta 5 parámetros de búsqueda de clasificación. Si lo desea, puede utilizar un `-` prefijo para indicar el orden descendente. El servidor ordenará en orden ascendente de forma predeterminada.

Los tipos de parámetros de búsqueda de clasificación admitidos son:`Number, String, Date, Quantity, Token, URI, Reference`. Si un parámetro de búsqueda hace referencia a un elemento que está anidado, este parámetro de búsqueda no se admite para la ordenación. Por ejemplo, busque por «nombre» del tipo de recurso Paciente hace referencia a Patient.El elemento NAME con el tipo de HumanName datos se considera anidado. Por lo tanto, no se permite ordenar los recursos de los pacientes por «nombre».

Example — Coloque los recursos para pacientes en un almacén de datos y ordénelos por fecha de nacimiento en orden ascendente:

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

_count

El parámetro `_count` se define como una instrucción al servidor sobre cuántos recursos deben devolverse en una sola página.

El tamaño máximo de página es 100. Los valores superiores a 100 no son válidos. `_count=0`no se admite.

Example — Busque el recurso para pacientes y establezca el tamaño de la página de búsqueda en 25:

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

Chaining and Reverse Chaining(_has)

El encadenamiento y el encadenamiento inverso FHIR proporcionan una forma más eficiente y compacta de obtener datos interconectados, lo que reduce la necesidad de realizar múltiples consultas independientes y hace que la recuperación de datos sea más cómoda para los desarrolladores y los usuarios.

Si cualquier nivel de recursión devuelve más de 100 resultados, HealthLake devolverá 4xx para evitar que el almacén de datos se sobrecargue y provoque múltiples paginaciones.

Example — Encadenar: obtiene todo DiagnosticReport lo que hace referencia a un paciente cuyo nombre es peter.

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

Example — Encadenamiento inverso: obtenga los recursos para pacientes, donde al menos una observación hace referencia al recurso para pacientes, donde la observación tiene el código 1234 y donde la observación hace referencia al recurso para pacientes en el parámetro de búsqueda de pacientes.

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

Modificadores de búsqueda compatibles

Los modificadores de búsqueda se utilizan con campos basados en cadenas. Todos los modificadores de búsqueda HealthLake utilizan una lógica booleana. Por ejemplo, puede especificar `:contains` que un campo de cadena más grande incluya una cadena pequeña para que se incluya en los resultados de la búsqueda.

Modificadores de búsqueda compatibles

Modificador de búsqueda	Tipo
:falta	Todos los parámetros excepto `Composite`
:exacto	Cadena
:contiene	Cadena
:no	Token
:texto	Token
:identificador	Referencia

Comparadores de búsqueda compatibles

Puede utilizar los comparadores de búsqueda para controlar la naturaleza de la coincidencia en una búsqueda. Puede utilizar los comparadores al buscar en los campos de número, fecha y cantidad. En la siguiente tabla se enumeran los comparadores de búsqueda y sus definiciones compatibles con. HealthLake

Comparadores de búsqueda compatibles

Comparador de búsquedas	Descripción
eq	El valor del parámetro en el recurso es igual al valor proporcionado.

Comparador de búsquedas	Descripción
Uno	El valor del parámetro del recurso no es igual al valor proporcionado.
gt	El valor del parámetro en el recurso es mayor que el valor proporcionado.
lt	El valor del parámetro en el recurso es inferior al valor proporcionado.
Edad	El valor del parámetro del recurso es mayor o igual al valor proporcionado.
le	El valor del parámetro en el recurso es menor o igual al valor proporcionado.
sa	El valor del parámetro del recurso comienza después del valor proporcionado.
eb	El valor del parámetro del recurso finaliza antes del valor proporcionado.

Los parámetros de búsqueda no son compatibles con HealthLake

Para obtener una lista completa de los parámetros de búsqueda compatibles, consulte el registro FHIR de parámetros de búsqueda. HealthLake admite todos los parámetros de búsqueda excepto los que aparecen en la tabla.

Parámetros de búsqueda no compatibles

Composición del paquete	Ubicación: cerca
Identificador de paquete	C onsent-source-reference
Paquete: mensaje	Paciente contratado
Tipo de paquete	Contenido del recurso

Marca de tiempo del paquete	Consulta de recursos

Búsqueda con POST ejemplos

Puede buscar en un HealthLake banco de datos realizando POST solicitudes. Puede proporcionar los parámetros de consulta en el cuerpo de la solicitud URI o en él, pero no puede usar ambos en una sola solicitud.

Los ejemplos de este tema siguen esa práctica recomendada.

> **Note**
>
> Para consultas que involucren información de identificación personal (PII) o Información de Salud Protegida (PHI), se recomienda utilizar `POST` solicitudes. En una `POST` solicitud, PII o PHI se agrega como parte del cuerpo de la solicitud y se cifra durante el tránsito.

Al realizar una POST solicitud con un parámetro en el cuerpo de la solicitud, `Content-Type: application/x-www-form-urlencoded` utilícelo como parte del encabezado.

En este tema se proporcionan ejemplos de cómo realizar búsquedas con POST los siguientes tipos de recursos.

- Edad: la edad no es un tipo de recurso definido enFHIR. En su lugar, la edad se captura como parte del tipo de recurso del paciente. Para buscar un grupo de pacientes en función de una edad o rango de edad específicos, utilice unthe section called “Comparadores de búsqueda compatibles”. Para obtener más detalles, consulte el tipo de recurso: Paciente en el índice de FHIR documentación.
- Condición: este tipo de recurso almacena detalles relacionados con conceptos clínicos, como un diagnóstico, situaciones, una afección clínica y problemas que han suscitado cierta preocupación. Para obtener más información, consulte Tipo de recurso: Condición en el índice de FHIR documentación. HealthLake crea nuevas condiciones en función de los documentos que se encuentran en DocumentReference. Estas adiciones se excluyen de forma predeterminada al realizar una `POST` solicitud. Para incluirlas, debe especificar un identificador válido para un recurso de condición en la búsqueda.
- DocumentReferencePara incluirlos, debe HealthLake especificar un identificador válido para un recurso de condición en su búsqueda. ----SEP----:Este tipo de recurso es compatible con.

Este tipo de recurso permite hacer referencia a documentos de cualquier tipo. Para obtener más información, consulte Tipo de recurso: DocumentReference en el índice de FHIR documentación. HealthLake también proporciona un procesamiento de lenguaje natural integrado (NLP) de los documentos que se encuentran en el DocumentReference. Para obtener más información, consulte Uso de la generación automática de recursos basada en el procesamiento del lenguaje natural (NLP) del tipo de FHIR DocumentReference recurso en AWS HealthLake.

- Ubicación: este tipo de recurso incluye tanto ubicaciones secundarias (un lugar que se utiliza para la atención médica sin designación o autorización previa) como ubicaciones dedicadas y designadas formalmente. Para obtener más información, consulte Tipo de recurso: ubicación en el índice de FHIR documentación.
- Observación: mediciones y afirmaciones simples realizadas sobre un paciente, un dispositivo u otro tema. HealthLake crea nuevos recursos de observación basados en los documentos que se encuentran en el DocumentReference recurso. Para obtener más información sobre cómo HealthLake crear nuevos recursos, consulteUso de la generación automática de recursos basada en el procesamiento del lenguaje natural (NLP) del tipo de FHIR DocumentReference recurso en AWS HealthLake. Estas adiciones se excluyen de forma predeterminada al realizar una `POST` solicitud. Para incluirlas, debe especificar un identificador válido para un recurso de observación en su búsqueda. Para obtener más información, consulte Tipo de recurso: observación en el índice de FHIR documentación.

Cada pestaña muestra ejemplos de cómo buscar en el tipo de recurso especificado. Incluye un ejemplo de cómo especificar la solicitud en el cuerpo de la solicitud.

Age

Utilice lo siguiente para realizar una solicitud de búsqueda `POST` basada en el tipo `Patient` de recurso. Esta búsqueda utiliza el `eq` comparador de búsquedas para buscar personas nacidas en 1997.

Debe especificar una solicitud URL y un cuerpo de la solicitud. Este es un ejemplo de solicitudURL.

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

Para especificar el año 1997 en la búsqueda, debe agregar el siguiente elemento al cuerpo de la solicitud.

```
birthdate=eq1997
```

Respuesta de JSON

Si se realiza correctamente, obtendrá un código de `200` HTTP respuesta y una JSON respuesta similar.

Condition

Utilice lo siguiente para realizar una `POST` solicitud sobre el tipo `Condition` de recurso. Esta búsqueda busca ubicaciones en su almacén de HealthLake datos que contienen el código médico`72892002`.

Debe especificar una solicitud URL y un cuerpo de la solicitud. Este es un ejemplo de solicitudURL.

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

Para especificar el código médico que desea buscar, añada este JSON elemento al cuerpo de la solicitud.

```
code=72892002
```

Respuesta de JSON

Si tiene éxito, recibirá un código de `200` HTTP respuesta. La siguiente JSON respuesta se ha truncado para mayor claridad.

```
{
 "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

Para ver los resultados del procesamiento HealthLake del lenguaje natural integrado (NLP) al realizar una `POST` solicitud sobre el tipo de `DocumentReference` recurso, el formato de la solicitud es el siguiente.

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

Para especificar el DocumentReference elemento al que desea hacer referencia, consulteParámetros de búsqueda. Los especificará en el cuerpo de la solicitud comoJSON.

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

Esta cadena de consulta utiliza varios parámetros de búsqueda para buscar en Amazon Comprehend API Medical las operaciones utilizadas para generar los resultados NLP médicos integrados.

Location

Utilice lo siguiente para realizar una `POST` solicitud sobre el tipo de `Location` recurso. Esta búsqueda busca ubicaciones en su banco de HealthLake datos que contienen el nombre de la ciudad Boston como parte de la dirección.

Debe especificar una solicitud URL y un cuerpo de la solicitud. A continuación, se muestra un ejemplo de solicitudURL.

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

Para especificar Boston en la búsqueda, añada el siguiente elemento al cuerpo de la solicitud:

```
address=Boston
```

Respuesta de JSON

Si se realiza correctamente, recibirá un código de `200` HTTP respuesta. La JSON respuesta se ha truncado para mayor claridad.

```
{
 "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

Utilice lo siguiente para realizar una solicitud de búsqueda `POST` basada en el tipo de `Observation` recurso. Esta búsqueda utiliza el parámetro `value-concept` de búsqueda para buscar el código médico,`266919005`. Este estado indica`Never smoker`.

Debe especificar una solicitud URL y un cuerpo de la solicitud. Este es un ejemplo de solicitudURL.

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

Para especificar el estado`Never smoker`, defina `value-concept=266919005` en el cuerpo delJSON.

```
value-concept=266919005
```

Respuesta de JSON

Si tiene éxito, obtendrá un código de `200` HTTP respuesta. La siguiente JSON respuesta se ha truncado para mayor claridad.

```
{
```

```
 "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"
   }
  }
 ]
}
```

Búsqueda con GET ejemplos

Puede buscar en un HealthLake banco de datos realizando GET solicitudes. HealthLake solo permite proporcionar parámetros de consulta como parte del URI cuerpo de la solicitud y no como parte del mismo.

> **Note**
>
> Para consultas que involucren información de identificación personal (PII) o Información de Salud Protegida (PHI), se recomienda utilizar `POST` solicitudes. En una `POST` solicitud, PII o PHI se agrega como parte del cuerpo de la solicitud y se cifra durante el tránsito.

En este tema se proporcionan ejemplos de cómo realizar búsquedas `GET` utilizando los tipos de recursos compatibles en HealthLake.

- Edad: la edad no es un tipo de recurso definido enFHIR. En cambio, la edad se captura como parte del tipo de recurso del paciente. Para buscar un grupo de pacientes en función de una edad o rango de edad específicos, necesitas usar unthe section called “Comparadores de búsqueda compatibles”. Para obtener más detalles, consulte el tipo de recurso: Paciente en el índice de FHIR documentación.
- Condición: este tipo de recurso almacena detalles relacionados con conceptos clínicos, como un diagnóstico, situaciones, una afección clínica y problemas que han suscitado preocupación. Para obtener más información, consulte Tipo de recurso: Condición en el índice de FHIR documentación. HealthLake crea nuevas condiciones en función de los documentos que se encuentran en DocumentReference. Estas adiciones se excluyen de forma predeterminada al

realizar una `POST` solicitud. Para incluirlas, debe especificar un identificador válido para un recurso de condición en la búsqueda.

- DocumentReferencePara incluirlos, debe HealthLake especificar un identificador válido para un recurso de condición en su búsqueda. ----SEP----:Este tipo de recurso es compatible con. Este tipo de recurso permite hacer referencia a documentos de cualquier tipo. Para obtener más información, consulte Tipo de recurso: DocumentReference en el índice de FHIR documentación. HealthLake también proporciona un procesamiento de lenguaje natural integrado (NLP) de los documentos que se encuentran en DocumentReference. Para obtener más información, consulte Uso de la generación automática de recursos basada en el procesamiento del lenguaje natural (NLP) del tipo de FHIR DocumentReference recurso en AWS HealthLake.
- Ubicación: este tipo de recurso incluye tanto ubicaciones secundarias (un lugar que se utiliza para la atención médica sin designación ni autorización previas) como ubicaciones dedicadas y designadas formalmente. Para obtener más información, consulte Tipo de recurso: ubicación en el índice de FHIR documentación.
- Observación: mediciones y afirmaciones sencillas realizadas sobre un paciente, un dispositivo u otro tema. HealthLake crea nuevos recursos de observación basados en los documentos que se encuentran en el DocumentReference recurso. Para obtener más información sobre cómo HealthLake crear nuevos recursos, consulteUso de la generación automática de recursos basada en el procesamiento del lenguaje natural (NLP) del tipo de FHIR DocumentReference recurso en AWS HealthLake. Estas adiciones se excluyen de forma predeterminada al realizar una `POST` solicitud. Para incluirlas, debe especificar un identificador válido para un recurso de observación en su búsqueda. Para obtener más información, consulte Tipo de recurso: observación en el índice de FHIR documentación.

Cada pestaña muestra un ejemplo de cómo buscar en el tipo de recurso especificado. Incluye un ejemplo de cómo especificar la solicitud en la URI JSON respuesta y la correspondiente.

Age

Utilice lo siguiente para realizar una solicitud de búsqueda `GET` basada en el tipo `Patient` de recurso. Esta búsqueda utiliza el `eq` comparador de búsquedas para buscar personas nacidas en 1997.

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

Respuesta de JSON

Si tiene éxito, obtendrá un código de 200 HTTP respuesta.

Condition

Utilice lo siguiente para realizar una GET solicitud sobre el tipo Condition de recurso. Esta búsqueda busca ubicaciones en su almacén de HealthLake datos que contienen el código médico72892002.

Debe especificar una solicitud URL y un cuerpo de la solicitud. Este es un ejemplo de solicitudURL.

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

Respuesta de JSON

Si tiene éxito, recibirá un código de 200 HTTP respuesta. La siguiente JSON respuesta se ha truncado para mayor claridad.

```
{
 "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

En este ejemplo se muestra cómo crear una solicitud de búsqueda sobre el tipo de `DocumentReference` recurso para pacientes a los que se les ha diagnosticado un estreptococo y a los que también se les ha recetado amoxicilina.

```
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
```

Si tiene éxito, obtendrá la siguiente respuesta. JSON

```
{
    "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

Utilice lo siguiente para realizar una `GET` solicitud sobre el tipo `Location` de recurso. Esta búsqueda busca ubicaciones en su banco de HealthLake datos que contienen el nombre de la ciudad Boston como parte de la dirección.

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

Respuesta de JSON

Si se realiza correctamente, obtendrá un código de `200` HTTP respuesta. La JSON respuesta se ha truncado para mayor claridad.


```
{
    "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

Utilice lo siguiente para realizar una solicitud de búsqueda GET basada en el tipo de `Observation` recurso. Esta búsqueda utiliza el parámetro `value-concept` de búsqueda para buscar el código médico,`266919005`. Este estado indica`Never smoker`.

Debe especificar una solicitud URL y una cadena de consulta. Este es un ejemplo de solicitudURL.

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

Para especificar el estado`Never smoker`, `value-concept=266919005` establézcalo como cadena de consulta.

Respuesta de JSON

Si tiene éxito, obtendrá un código de `200` HTTP respuesta. La siguiente JSON respuesta se ha truncado para mayor claridad.

```
{
 "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"
 }
}
```

```
 ]
}
```

Lectura del historial FHIR de recursos

La FHIR `history` interacción recupera el historial de un FHIR recurso concreto en un almacén de HealthLake datos. Con esta interacción, puede determinar cómo ha cambiado el contenido de un FHIR recurso a lo largo del tiempo. También resulta útil en coordinación con los registros de auditoría para ver el estado de un recurso antes y después de la modificación.

> **Note**
>
> FHIRel recurso `history` está habilitado de forma predeterminada en todos los almacenes de HealthLake datos creados después del 25 de octubre de 2024. Si su banco de datos se creó antes de esta fecha, puede enviar un ticket de soporte para activar la FHIR `history` interacción. Cree un caso utilizando AWS Support Center Console. Para crear tu caso, inicia sesión en tu cuenta Cuenta de AWS y selecciona Crear caso.

La `history` interacción se realiza mediante el `HTTP GET` comando. Las FHIR interacciones `create` y `delete` dan como resultado una versión histórica del recurso que se va a guardar. `update` HealthLake admite los siguientes parámetros de búsqueda para la FHIR `history` interacción.

HealthLake parámetros de búsqueda compatibles para la FHIR **`history`** interacción

Parámetro de búsqueda	Descripción
`_count : integer`	El número máximo de resultados de búsqueda en una página. El servidor devolverá el número solicitado o el número máximo de resultados de búsqueda permitido de forma predeterminada para el almacén de datos, lo que sea inferior.
`_since : instant`	Incluya únicamente las versiones de recursos que se hayan creado en un instante determina do o con posterioridad a él.

Parámetro de búsqueda	Descripción
`_at : date(Time)`	Incluya únicamente las versiones de recursos que estuvieran actualizadas en algún momento durante el período de tiempo especificado en el valor de fecha y hora. Para obtener más información, consulte `date`la HL7FHIRRE STfulAPIdocumentación.

El siguiente ejemplo devuelve 100 resultados de búsqueda históricos por página para un FHIR `Patient` recurso en HealthLake. Para ver la URL ruta completa, desplázate sobre el botón Copiar. URLTiene la forma:

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

El contenido devuelto de una interacción histórica está contenido en un FHIR recurso `Bundle`, con el tipo establecido en`history`. Contiene el historial de versiones especificado, ordenado en último lugar por las versiones más antiguas, e incluye los recursos eliminados. Para obtener información adicional sobre la `history` interacción, consulte `history`la HL7 FHIR RESTful API documentación.

> **Note**
>
> Puede optar por no utilizar tipos de `history` FHIR recursos específicos. Para excluirse, cree un caso utilizando AWS Support Center Console. Para crear tu caso, inicia sesión en tu cuenta Cuenta de AWS y selecciona Crear caso.

Leyendo el historial de recursos específico de la versión FHIR

La FHIR `vread` interacción realiza una lectura específica de la versión de un recurso en un almacén de datos. HealthLake Con esta interacción, puede ver el contenido de un FHIR recurso tal como estaba en un momento determinado del pasado.

HealthLake declara que admite el control de versiones `CapabilityStatement.rest.resource.versioning`para cada recurso compatible. Todos

los almacenes de HealthLake datos incluyen `Resource.meta.versionId` (`vid`) en todos los recursos.

Cuando FHIR `history` la interacción está habilitada (de forma predeterminada para los almacenes de datos creados después del 25 de octubre de 2024 o cuando se solicita para los almacenes de datos más antiguos), la `Bundle` respuesta incluye la `vid` como parte del. `location` En el siguiente ejemplo, se `vid` muestra como un número. 1 Para ver el ejemplo completo, consulta Example bundle/bundle-response (). JSON

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

La `vread` interacción se realiza mediante el comando. `HTTP GET` La siguiente `vread` interacción devuelve una sola instancia con el contenido especificado para el FHIR `Patient` recurso para la versión de los metadatos del recurso especificada por`vid`. Para ver la URL ruta completa en el siguiente ejemplo, desplácese sobre el botón Copiar. Tiene URL el siguiente formato:

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

> **Note**
>
> Si utiliza la `history` interacción sin ella `vread` al leer un FHIR recurso, HealthLake siempre devuelve la última versión de los metadatos del recurso.

Para obtener información adicional sobre la `vread` interacción, consulte `vread`la APIdocumentación de HL7 FHIR Restful.

Obtención de datos de pacientes con la operación Patient $everything FHIR REST API

La operación Patient $everything se utiliza para consultar un recurso para FHIR pacientes junto con cualquier otro recurso relacionado con ese paciente. Esta operación se puede utilizar para proporcionar a un paciente acceso a su historial completo o para que un proveedor realice una

descarga masiva de datos relacionados con un paciente. HealthLake admite $everything para un identificador de paciente específico.

> **Note**
>
> La operación Patient $everything se admite actualmente en los almacenes de datos creados después del 27 de febrero de 2024.

Obtenga todos los recursos relacionados con un paciente

Patient $everything es una REST API operación que se puede invocar como se muestra en los ejemplos siguientes.

GET Request

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

> **Note**
>
> Los recursos en respuesta se ordenan por tipo de recurso e identificador del recurso.
> La respuesta siempre se rellena con bundle.total.

Parámetros de Patient $everything

HealthLake admite los siguientes parámetros de consulta

Parámetro	Detalles
iniciar	Obtenga todos los datos del paciente después de una fecha de inicio especificada.
end	Obtenga todos los datos del paciente antes de una fecha de finalización especific ada.
since	Actualice todos los datos del paciente después de una fecha específica.

Parámetro	Detalles
_tipo	Obtenga datos de pacientes para tipos de recursos específicos.
_count	Obtenga los datos del paciente y especifique el tamaño de la página.

Example - Obtenga todos los datos del paciente después de una fecha de inicio específica

El paciente $everything puede usar el `start` filtro para consultar datos solo después de una fecha específica.

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

Example - Obtenga todos los datos del paciente antes de una fecha de finalización específica

El paciente $everything puede usar el end filtro solo para consultar datos anteriores a una fecha específica.

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

Example - Actualice todos los datos del paciente después de una fecha específica

El paciente $everything puede usar el `since` filtro para consultar únicamente los datos actualizados después de una fecha específica.

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

Example - Obtenga datos de pacientes para tipos de recursos específicos

El paciente $everything puede usar el `_type` filtro para especificar tipos de recursos específicos que se incluirán en la respuesta. Se pueden especificar varios tipos de recursos en una lista separada por comas.

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

Example - Obtenga los datos del paciente y especifique el tamaño de la página

Patient $everything puede usar el `_count` para configurar el tamaño de la página.

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

Paciente: $todo **`start`** y sus atributos **`end`**

HealthLake admite los siguientes atributos de recursos para los parámetros de inicio y finalización de la consulta.

Recurso	Elemento de recurso
Cuenta	Cuenta. servicePeriod.start
AdverseEvent	AdverseEvent.date
AllergyIntolerance	AllergyIntolerance.recordedDate
Cita	Cita. Inicio
AppointmentResponse	AppointmentResponse.iniciar
AuditEvent	AuditEvent.periodo.inicio
Basic	Básico. Creado
BodyStructure	NO_ DATE

Recurso	Elemento de recurso
CarePlan	CarePlan.periodo.inicio
CareTeam	CareTeam.periodo.inicio
ChargeItem	ChargeItem. occurrenceDateTime, ChargeItem. occurrencePeriod.start, ChargeItem. occurrenceTiming.evento
Reclamación	Reclamación. billablePeriod.iniciar
ClaimResponse	ClaimResponse.creado
ClinicalImpression	ClinicalImpression.fecha
Comunicación	Comunicación. Enviada
CommunicationRequest	CommunicationRequest. occurrenceDateTime, CommunicationRequest. occurrencePeriod.iniciar
Composición	Composición.fecha
Condición	Condición. recordedDate
Consentimiento	Consentimiento. dateTime
Cobertura	Cobertura. Periodo. Inicio
CoverageEligibilityRequest	CoverageEligibilityRequest.creado

Recurso	Elemento de recurso
CoverageEligibilityResponse	CoverageEligibilityResponse.creado
DetectedIssue	DetectedIssue.identificado
DeviceRequest	DeviceRequest.authoredOn
DeviceUseStatement	DeviceUseStatement.recordedOn
DiagnosticReport	DiagnosticReport.efectivo
DocumentManifest	DocumentManifest.creado
DocumentReference	DocumentReference.context.period.start
Encuentro	Encuentro, punto, inicio
EnrollmentRequest	EnrollmentRequest.creado
EpisodeOfCare	EpisodeOfCare.periodo.inicio
ExplanationOfBenefit	ExplanationOfBenefit. billablePeriod.iniciar
FamilyMemberHistory	NO_ DATE
Indicador	Bandera, punto, inicio

Recurso	Elemento de recurso
Objetivo	Gol. statusDate
Grupo	NO_ DATE
ImagingStudy	ImagingStudy.empezado
Inmunización	Inmunización. Registrada
ImmunizationEvaluation	ImmunizationEvaluation.fecha
ImmunizationRecommendation	ImmunizationRecommendation.fecha
Factura	Factura.fecha
Enumeración	Fecha de lista
MeasureReport	MeasureReport.periodo.inicio
Medios	Medios de comunicación. Emitido
MedicationAdministration	MedicationAdministration... efectivo
MedicationDispense	MedicationDispense.whenPrepared
MedicationRequest	MedicationRequest.authoredOn

Recurso	Elemento de recurso
MedicationStatement	MedicationStatement.dateAsserted
MolecularSequence	NO_ DATE
NutritionOrder	NutritionOrder.dateTime
Observación	Observación. Efectiva
Paciente	NO_ DATE
Persona	NO_ DATE
Procedimiento	Procedimiento. Realizado
Procedencia	Procedencia. occurredPeriod.start, Provenance. occurredDateTime
QuestionnaireResponse	QuestionnaireResponse.de autor
RelatedPerson	NO_ DATE
RequestGroup	RequestGroup.authoredOn
ResearchSubject	ResearchSubject.período
RiskAssessment	RiskAssessment. occurrenceDateTime, RiskAssessment. occurrencePeriod.iniciar

Recurso	Elemento de recurso
Programación	Horario. planningHorizon
ServiceRequest	ServiceRequest.authoredOn
Ejemplar	Espécimen. receivedTime
SupplyDelivery	SupplyDelivery. occurrenceDateTime, SupplyDelivery. occurrencePeriod.start, SupplyDelivery. occurrenceTiming.evento
SupplyRequest	SupplyRequest.authoredOn
VisionPrescription	VisionPrescription.dateWritten

Exportación de datos de su almacén HealthLake de datos mediante $export

Para realizar una solicitud de exportación, utilice la FHIR REST API especificación `$export` como parte de la `POST` solicitud e incluya los parámetros de la solicitud en el cuerpo de la solicitud. Según la FHIR especificación, el FHIR servidor debe admitir `GET` las solicitudes y puede `POST` admitirlas. Para admitir parámetros adicionales, se necesita un organismo que inicie la exportación y, por lo tanto, HealthLake admite POST las solicitudes.

> **Important**
>
> HealthLake Los almacenes de datos creados antes del 1 de junio de 2023 solo admiten solicitudes de trabajo de exportación FHIR REST API basadas en exportaciones a nivel de todo el sistema.
> HealthLake Los almacenes de datos creados antes del 1 de junio de 2023 no permiten obtener el estado de una exportación mediante una `GET` solicitud en el punto final de un almacén de datos.

Todas las solicitudes de exportación que realice mediante el FHIR REST API se devuelven en `ndjson` formato y se exportan a un bucket de Amazon S3. Cada objeto de S3 contendrá un solo tipo FHIR de recurso.

Puede realizar una única solicitud de exportación para cada AWS cuenta a la vez. Para obtener más información sobre los Service Quotas asociados a HealthLake, consulteAWS HealthLake puntos finales y cuotas.

Para obtener más información sobre cómo realizar una solicitud de exportación mediante el FHIR RESTAPI, consulteExportación de datos de su almacén de HealthLake datos con FHIR REST API operaciones.

Consulte AWS HealthLake los almacenes de datos con SQL Amazon Athena

Al crear un banco de HealthLake datos, la estructura de FHIR datos altamente anidada se incorpora a Amazon Athena y se transforma automáticamente en tablas Iceberg con las que se pueden consultar. SQL El acceso a este nuevo recurso se gestiona mediante AWS Lake Formation. Cada tipo de FHIR recurso se representa como una tabla individual en Athena.

> **Important**
>
> En el caso de los almacenes de datos creados antes del 14 de noviembre de 2022, debe migrar su banco de datos existente a uno nuevo para consultarlo mediante SQL él. Para obtener ayuda, consulte Migración de un almacén de datos existente para usar Amazon Athena.

> **Note**
>
> Después del 20 de febrero de 2023, HealthLake los almacenes de datos no utilizan el procesamiento de lenguaje natural integrado (NLP) de forma predeterminada. Si está interesado en activar esta función en su almacén de datos, consulte el capítulo ¿Cómo activo la función HealthLake de procesamiento de lenguaje natural integrada? de solución de problemas.

Para crear un HealthLake banco de datos, debe añadir IAM políticas adicionales y un rol de servicio a su IAM usuario o rol de HealthLake administrador. Para obtener más información sobre la configuración de permisos, consulteConfigurar los permisos para empezar a usar AWS HealthLake.

HealthLake Los almacenes de datos se incorporan a Athena como tablas de iceberg. Para obtener más información sobre el funcionamiento de las tablas Iceberg en Athena, consulte Uso de tablas Iceberg en la Guía del usuario de Athena.

HealthLake admite `READ` las operaciones de sus almacenes de HealthLake datos y almacenes de datos en Athena. Para obtener más información sobre las operaciones de creación, lectura, actualización y eliminación (CRUD) que utilizan las FHIR REST API operaciones, consulte Uso de

FHIR REST API interacciones con un almacén HealthLake de datos para obtener más información sobre cómo afectan CRUD las operaciones a sus datos en Athena.

En los temas de este capítulo se describe cómo conectar el HealthLake banco de datos a Athena, cómo consultarlo mediante SQL Athena y cómo conectar los resultados con otros AWS servicios para su posterior análisis.

Contenido

Conexión de su almacén de datos a Amazon Athena

> **Important**
>
> Después del 14 de noviembre de 2022, los IAM requisitos de acceso HealthLake cambiaron. Tanto para crear almacenes de datos como para conceder acceso a ellos en Athena, debe añadir la política `AWSLakeFormationDataAdmin` gestionada a su IAM usuario, grupo o función. Puede usar la `AWSLakeFormationDataAdmin` política para crear administradores de lagos de datos y conceder acceso a los almacenes de datos en Athena.

En este tema se describen los pasos necesarios para crear un usuario, grupo o rol de Athena y concederle acceso a los FHIR recursos que se encuentran en un almacén de HealthLake datos.

- Otorgar a un usuario, grupo o rol acceso a un almacén de HealthLake datos (AWS Lake Formation Console)
- Configuración de una cuenta de Athena

Otorgar a un usuario, grupo o rol acceso a un almacén de HealthLake datos (AWS Lake Formation Console)

> Persona: HealthLake administrador
>
> La persona HealthLake administradora es un administrador de un lago de datos en AWS Lake Formation. Permiten el acceso a HealthLake los almacenes de datos en Lake Formation.

Para cada banco de datos creado, hay dos entradas visibles en la consola de AWS Lake Formation. Una entrada es un enlace a un recurso. Los nombres de los enlaces de recursos siempre se muestran en cursiva. Cada enlace a un recurso se muestra con el nombre y el propietario del recurso compartido vinculado. En todos los almacenes de HealthLake datos, el propietario del recurso compartido es la cuenta HealthLake de servicio. La otra entrada es el almacén HealthLake de datos de la cuenta HealthLake de servicio. En los pasos de este procedimiento se utiliza el banco de datos que es el enlace de recursos.

Para obtener más información sobre los enlaces de recursos, consulte Cómo funcionan los enlaces de recursos en Lake Formation en la Guía para desarrolladores de AWS Lake Formation.

Para que un usuario, grupo o rol pueda consultar datos en Athena, debe conceder el permiso Describe en la base de datos de recursos. A continuación, debe conceder las opciones Seleccionar y Describir en las tablas.

STEP1: Para conceder DESCRIBEpermisos en una base de HealthLake datos de enlaces a recursos de un almacén de datos

1. Abre la consola de AWS Lake Formation: https://console.aws.amazon.com/lakeformation/
2. En la barra de navegación principal, selecciona Bases de datos.
3. En la página Bases de datos, pulse el botón de opción situado junto al nombre del banco de datos que aparece en cursiva.
4. Elija Acciones (▾).
5. Elija Conceder.
6. En la página Otorgar permisos de datos, en Directores, elija IAMusuarios o roles.
7. En IAMUsuarios o roles, utilice la flecha hacia abajo (▾) o busque el IAM usuario, rol o grupo sobre el que desee realizar consultas en Athena.

8. En las etiquetas LF o en la tarjeta de recursos del catálogo, elija la opción Recursos del catálogo de datos con nombre asignado.
9. En Bases de datos, utilice la flecha hacia abajo (▼) para elegir la base de HealthLake datos del almacén de datos a la que desea compartir el acceso.
10. En la tarjeta de permisos de enlace de recursos, en Permisos de enlace de recursos, elija Describir.

Cuando la concesión se haya concedido correctamente, aparecerá el aviso de autorización de concesión correcta. Para ver el permiso que acabas de conceder, selecciona Permisos de Data Lake. Busque el usuario, el grupo y el rol en la tabla. En la columna Permisos, verá la lista Describe.

Ahora debe usar Grant on target para conceder las opciones Seleccionar y Describir en todas las tablas de la base de datos.

STEP2: Otorgue acceso a todas las tablas de un enlace de recursos del almacén de HealthLake datos

1. Abre la consola de AWS Lake Formation: https://console.aws.amazon.com/lakeformation/
2. En la barra de navegación principal, selecciona Bases de datos.
3. En la página Bases de datos, pulse el botón de opción situado junto al nombre del banco de datos que aparece en cursiva.
4. Elija Acciones (▼).
5. Elige Subvencionar según el objetivo.
6. En la página Otorgar permisos de datos, en Directores, selecciona IAMusuarios o roles.
7. En IAMUsuarios o roles, utilice la flecha hacia abajo (▼) o busque el IAM usuario, grupo o rol sobre el que desee realizar consultas en Athena.
8. En las etiquetas LF o en la tarjeta de recursos del catálogo, elija la opción Recursos del catálogo de datos con nombre asignado.
9. En Bases de datos, utilice la flecha hacia abajo (▼) para elegir la base de HealthLake datos del almacén de datos a la que desea conceder acceso.
10. En Tablas, elija Todas las tablas para compartir todas las tablas con un HealthLake usuario.
11. En la tarjeta de permisos de tabla, en Permisos de tabla, elija Describir y seleccionar.
12. Elija Conceder.

Tras seleccionar la concesión, aparece un aviso de autorización de autorización de concesión de permisos. El usuario especificado ahora puede realizar consultas en un almacén de HealthLake datos de Athena.

Cómo empezar con Athena

> **HealthLake usuario**
>
> El HealthLake usuario utilizará la consola de Athena o AWS SDKs consultará un almacén de HealthLake datos compartido con él por el HealthLake administrador. AWS CLI

Para consultar un banco de datos con Athena, debe hacer las tres cosas siguientes.

- Conceda al IAM usuario o rol acceso al almacén de HealthLake datos a través de Lake Formation. Para obtener más información, consulte Otorgar a un usuario, grupo o rol acceso a un almacén de HealthLake datos (AWS Lake Formation Console).
- Cree un grupo de trabajo para su almacén HealthLake de datos.
- Designe un bucket de Amazon S3 para almacenar los resultados de las consultas.

Para empezar a utilizar Athena, añada las políticas gestionadas por Amazon S3 AmazonAthenaFullAccessy las políticas FullAccess AWS gestionadas por Amazon S3 a su usuario, grupo o función. El uso de una política AWS administrada es una excelente forma de empezar a utilizar un nuevo servicio. Tenga presente que es posible que las políticas administradas de AWS no concedan permisos de privilegios mínimos para sus casos de uso concretos, ya que están disponibles para que las utilicen todos los clientes de AWS. Al establecer permisos con IAM políticas, conceda solo los permisos necesarios para realizar una tarea. Para obtener más información sobre los permisos con privilegios mínimos IAM y cómo aplicarlos, consulte Aplicar permisos con privilegios mínimos en la Guía del usuario. IAM

> **Important**
>
> Para consultar un banco HealthLake de datos en Athena, debe utilizar la versión 3 del motor Athena.

Los grupos de trabajo son recursos y, por lo tanto, puede utilizar políticas IAM basadas para controlar el acceso a grupos de trabajo específicos. Para obtener más información, consulte Uso de grupos de trabajo para controlar el acceso y los costos de las consultas en la Guía del usuario de Athena.

Para obtener más información sobre la configuración de grupos de trabajo, consulte la https://docs.aws.amazon.com/athena/latest/ug/workgroups-procedure.html Guía del usuario de Athena.

> **Note**
>
> La región en la que se encuentra su bucket de Amazon S3 y la consola Athena deben coincidir.

Antes de poder ejecutar una consulta, se debe especificar una ubicación de bucket de resultado de consulta en Amazon S3 o debe utilizar un grupo de trabajo que tenga especificado un bucket y cuya configuración sobrescriba la configuración del cliente. Los archivos de salida se guardan automáticamente por cada consulta que se hace.

Para obtener más información sobre cómo especificar las ubicaciones de los resultados de las consultas en la consola de Athena, consulte Especificar una ubicación de resultados de consulta mediante la consola de Athena en la Guía del usuario de Amazon Athena.

Para ver ejemplos de cómo consultar el banco de HealthLake datos en Athena, consulte. Consulte su almacén HealthLake de datos mediante SQL

Consulte su almacén HealthLake de datos mediante SQL

> **Note**
>
> Después del 20 de febrero de 2023, HealthLake los almacenes de datos no utilizan el procesamiento de lenguaje natural integrado (NLP) de forma predeterminada. Si está interesado en activar esta función en su banco de datos, consulte el capítulo ¿Cómo activo la función HealthLake de procesamiento de lenguaje natural integrada? de solución de problemas.
>
> Todos los ejemplos de este tema utilizan datos ficticios creados con Synthea. Para obtener más información sobre cómo crear un banco de datos precargado con datos de Synthea, consulte. Crear un almacén de datos en AWS HealthLake

Al importar el banco de HealthLake datos a Athena, cada tipo de recurso del banco de HealthLake datos se convierte en una tabla. Estas tablas se pueden consultar individualmente o en grupo mediante consultas SQL basadas. Debido a la estructura de los almacenes de datos, los datos se importan a Athena como varios tipos de datos diferentes. Para obtener más información sobre la creación de SQL consultas que puedan acceder a estos tipos de datos, consulte Consultas de matrices con tipos complejos y estructuras anidadas en la Guía del usuario de Amazon Athena.

Para cada elemento de un tipo de recurso, la FHIR especificación define una cardinalidad. La cardinalidad de un elemento define los límites inferior y superior del número de veces que puede aparecer este elemento. Al crear una SQL consulta, debe tener esto en cuenta. Por ejemplo, veamos algunos elementos de Tipo de recurso: paciente.

- Elemento: nombre La FHIR especificación establece la cardinalidad como`0..*`.

  El elemento se captura como una matriz.

```
[{
 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
}]
```

  En Athena, para ver cómo se ha ingerido un tipo de recurso, búsquelo en Tablas y vistas. Para acceder a los elementos de esta matriz, puede utilizar la notación de puntos. A continuación, se muestra un ejemplo sencillo en el que se accedería a los valores de `given` y`family`.

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

```
FROM Patient
```

- Elemento: MaritalStatus la FHIR especificación establece la cardinalidad como`0..1`.

  Este elemento se captura comoJSON.

```
{
 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
}
```

  En Athena, para ver cómo se ha ingerido un tipo de recurso, búsquelo en Tablas y vistas. Para acceder a los pares clave-valor enJSON, puede utilizar la notación de puntos. Como no es una matriz, no se requiere ningún índice de matriz. A continuación, se muestra un ejemplo sencillo con el que se accedería al valor `detext`.

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

Para obtener más información sobre el acceso y la búsquedaJSON, consulte Consultas JSON en la Guía del usuario de Athena.

Las instrucciones de consulta de Athena Data Manipulation Language (DML) se basan en Trino. Athena no es compatible con todas las funciones de Trino y existen diferencias significativas. Para obtener más información, consulte DMLconsultas, funciones y operadores en la Guía del usuario de Amazon Athena.

Además, Athena admite varios tipos de datos que puede encontrar al crear consultas en su banco de HealthLake datos. Para obtener más información sobre los tipos de datos en Athena, consulte Tipos de datos en Amazon Athena en la Guía del usuario de Amazon Athena.

Para obtener más información sobre cómo funcionan SQL las consultas en Athena, consulte la SQLreferencia de Amazon Athena en la Guía del usuario de Amazon Athena.

Cada pestaña muestra ejemplos de cómo buscar en los tipos de recursos especificados y los elementos asociados con Athena.

Element: Extension

El elemento `extension` se utiliza para crear campos personalizados en un banco de datos.

En este ejemplo, se muestra cómo acceder a las funciones del `extension` elemento que se encuentra en el tipo de `Patient` recurso.

Cuando el banco de HealthLake datos se importa a Athena, los elementos de un tipo de recurso se analizan de forma diferente. Como la estructura de `element` es variable, no se puede especificar completamente en el esquema. Para gestionar esa variabilidad, los elementos de la matriz se pasan como cadenas.

En la descripción de la tabla`Patient`, puede ver el elemento `extension` descrito como`array<string>`, lo que significa que puede acceder a los elementos de la matriz mediante un valor de índice. Sin embargo, para acceder a los elementos de la cadena, debe utilizar`json_extract`.

Esta es una única entrada del `extension` elemento que se encuentra en la tabla de pacientes.

```
[{
  "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"
 }
]
```

Aunque esto es válidoJSON, Athena lo trata como una cadena.

En este ejemplo de SQL consulta se muestra cómo se puede crear una tabla que contenga los `patient-birthPlace` elementos `patient-mothersMaidenName` y. Para acceder a estos elementos, debe utilizar diferentes índices de matriz y `json_extract.`

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

Para obtener más información sobre las consultas que implicanJSON, consulte Extracción de datos de la JSON Guía del usuario de Amazon Athena.

Element: birthDate (Age)

La edad no es un elemento del tipo de recurso para pacientes en. FHIR Estos son dos ejemplos de búsquedas que se filtran según la edad.

Como la edad no es un elemento, utilizamos la `birthDate` para las SQL consultas. Para ver cómo se ha introducido un elementoFHIR, busque el nombre de la tabla en Tablas y vistas. Puede ver que es de tipo cadena.

Ejemplo 1: Calcular un valor para la edad

En esta SQL consulta de ejemplo, utilizamos una SQL herramienta integrada `year` para extraer esos componentes. `current_date` Luego, los restamos para obtener la edad real del paciente, como se denomina `age` en una columna.

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

Ejemplo 2: Filtrar los pacientes que nacieron antes `2019-01-01` y que lo son`male`.

La SQL consulta muestra cómo utilizar la `CAST` función para convertir el `birthDate` elemento en un tipo `DATE` y cómo filtrar en función de dos criterios de la `WHERE` cláusula. Como el elemento se introduce como cadena de texto de forma predeterminada, debemos `CAST` incorporarlo como tipo`DATE`. Luego puede usar el < operador para compararlo con una fecha diferente,`2019-01-01`. Al usar`AND`, puede agregar un segundo criterio a la `WHERE` cláusula.

```
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

En este ejemplo, se muestran las búsquedas de ubicaciones dentro del tipo de recurso Ubicación donde el nombre de la ciudad es 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 condición del tipo de recurso almacena datos de diagnóstico relacionados con problemas que han llegado a ser preocupantes. HealthLakeEl procesamiento médico integrado del lenguaje natural (NLP) genera nuevos `Condition` recursos en función de los detalles que se encuentran en el tipo de DocumentReference recurso. Cuando se genera un nuevo recurso, HealthLake añade la etiqueta `SYSTEM_GENERATED` al `meta` elemento. Esta SQL consulta de ejemplo muestra cómo se puede buscar en la tabla de condiciones y devolver los resultados cuando los `SYSTEM_GENERATED` resultados se han eliminado.

Para obtener más información sobre HealthLake el procesamiento de lenguaje natural integrado (NLP), consulteUso de la generación automática de recursos basada en el procesamiento del lenguaje natural (NLP) del tipo de FHIR DocumentReference recurso en AWS HealthLake.

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

También puede buscar dentro de un elemento de cadena específico para filtrar aún más la consulta. El `modifierextension` elemento contiene detalles sobre qué `DocumentReference` recurso se utilizó para generar un conjunto de condiciones. De nuevo, debes usarlos `json_extract` para acceder a los JSON elementos anidados que se traen a Athena como una cadena.

En este ejemplo de SQL consulta se muestra cómo se puede buscar todo lo `Condition` que se ha generado a partir de una consulta específica. `DocumentReference` Se utiliza `CAST` para establecer el JSON elemento como una cadena que se puede utilizar `LIKE` para comparar.

```
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

El tipo de recurso Observation almacena las mediciones y las afirmaciones sencillas realizadas sobre un paciente, un dispositivo u otro tema. HealthLakeEl procesamiento integrado del lenguaje natural (NLP) genera nuevos `Observation` recursos en función de los detalles que se encuentran en un `DocumentReference` recurso. Este ejemplo de SQL consulta incluye `WHERE meta.tag[1] is NULL` comentarios, lo que significa que se incluyen los `SYSTEM_GENERATED` resultados.

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

Esta columna se importó como `struct`. Por lo tanto, puede acceder a los elementos que contiene mediante la notación de puntos.

Resource type: MedicationStatement

MedicationStatement es un tipo de FHIR recurso que puede utilizar para almacenar detalles sobre los medicamentos que un paciente ha tomado, está tomando o tomará en el futuro. HealthLakeEl procesamiento médico integrado del lenguaje natural (NLP) genera nuevos MedicationStatement recursos a partir de los documentos que se encuentran en ese tipo de DocumentReference recurso. Cuando se generan nuevos recursos, HealthLake añade la etiqueta `SYSTEM_GENERATED` al `meta` elemento. En este ejemplo de SQL consulta se muestra cómo crear una consulta que filtre en función de un solo paciente mediante su identificador y busque los recursos que ha agregado HealthLake's IntegratedNLP.

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

Para obtener más información sobre HealthLake la medicina integradaNLP, consulteUso de la generación automática de recursos basada en el procesamiento del lenguaje natural (NLP) del tipo de FHIR DocumentReference recurso en AWS HealthLake.

Ejemplos de SQL consultas con filtrado complejo

> **Note**
>
> Después del 20 de febrero de 2023, HealthLake los almacenes de datos no utilizan el procesamiento de lenguaje natural integrado (NLP) de forma predeterminada. Si está interesado en activar esta función en su banco de datos, consulte el capítulo ¿Cómo activo la función HealthLake de procesamiento de lenguaje natural integrada? de solución de problemas.

Los ejemplos de este tema incluyen SQL consultas para HealthLake la integración con Athena que utilizan un filtrado complejo.

Example Crear criterios de filtrado basados en datos demográficos

Es importante identificar los datos demográficos correctos de los pacientes a la hora de crear una cohorte de pacientes. Este ejemplo de consulta demuestra cómo puede utilizar la notación de puntos de Trino y filtrar `json_extract` los datos del banco de HealthLake datos.

```
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;
```

Con la consola Athena, puede ordenar y descargar aún más los resultados.

Example Crear filtros para un paciente y sus afecciones relacionadas

Este ejemplo de consulta muestra cómo encontrar y ordenar todas las afecciones relacionadas con los pacientes que se encuentran en un banco de HealthLake datos.

```
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;
```

Puede utilizar la consola Athena para ordenar aún más estos resultados o descargarlos para analizarlos más a fondo.

Example Crear filtros para un paciente y sus observaciones relacionadas

Este ejemplo de consulta demuestra cómo encontrar y ordenar todas las observaciones relacionadas de los pacientes que se encuentran en un banco de HealthLake datos.

```
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 Crear condiciones de filtrado para un paciente y sus procedimientos relacionados

Conectar los procedimientos con los pacientes es un aspecto importante de la atención médica. Esta SQL consulta muestra cómo puede utilizar los tipos de recursos para pacientes y procedimientos para hacerlo en Athena. Esta SQL consulta devolverá todos los pacientes y sus procedimientos relacionados que se encuentran en su almacén de HealthLake datos.

```
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;
```

Ahora puede usar la consola Athena para descargar los resultados para analizarlos más a fondo o ordenarlos para comprenderlos mejor.

Example Crear condiciones de filtrado para un paciente y sus recetas relacionadas

Es importante ver una lista actualizada de los medicamentos que toman los pacientes. Con Athena, puede escribir una SQL consulta que utilice los tipos de paciente y de MedicationRequest recursos que se encuentran en su almacén de HealthLake datos.

Esta SQL consulta une el paciente y MedicationRequest las tablas importadas a Athena. También organiza las recetas en sus entradas individuales mediante la notación de puntos.

```
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
```

Puede utilizar la consola Athena para ordenar los resultados o descargarlos para analizarlos más a fondo.

Example Ver los medicamentos que se encuentran en el tipo de MedicationStatement recurso

La consulta de ejemplo muestra cómo organizar los anidados JSON importados a SQL Athena utilizando. La consulta utiliza el `meta` elemento para indicar cuándo se ha agregado un medicamento mediante el procesamiento HealthLake de lenguaje natural integrado ()NLP. Para obtener más información sobre HealthLake la integración con Amazon Comprehend Medical, Uso de la generación automática de recursos basada en el procesamiento del lenguaje natural (NLP) del tipo de FHIR DocumentReference recurso en AWS HealthLake consulte. También se utiliza `json_extract` para buscar datos dentro de la matriz de JSON cadenas.

```
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;
```

Puede utilizar la consola Athena para descargar estos resultados u ordenarlos.

Example Filtra por un tipo de enfermedad específico

El ejemplo muestra cómo encontrar un grupo de pacientes de entre 18 y 75 años a los que se les haya diagnosticado diabetes.

```
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%');
```

Ahora puede usar la consola Athena para ordenar los resultados o descargarlos para analizarlos más a fondo.

AWS HealthLake y VPC puntos finales de interfaz ()AWS PrivateLink

Puedes establecer una conexión privada entre tu dispositivo VPC y el tuyo AWS HealthLake mediante la creación de un VPCpunto final de interfaz. Los VPC puntos finales de la interfaz funcionan con una tecnología que puede utilizar para acceder de forma privada HealthLake; APIs sin una puerta de enlace, NAT dispositivo, VPN conexión o AWS Direct Connect conexión a Internet. AWS PrivateLink Las instancias VPC que tenga no necesitan direcciones IP públicas para comunicarse con ellas HealthLake;. APIs El tráfico entre usted VPC y HealthLake; no sale de la red de Amazon.

Cada punto de conexión de la interfaz está representado por una o más interfaces de red elásticas en las subredes.

Para obtener más información, consulte Interface VPC endpoints (AWS PrivateLink) en la Guía del VPC usuario de Amazon.

Consideraciones sobre los puntos finales HealthLake VPC

Antes de configurar un VPC punto final de interfaz para HealthLake, asegúrese de revisar las propiedades y limitaciones del punto final de interfaz en la Guía del VPC usuario de Amazon.

HealthLake permite realizar llamadas a todas sus API acciones desde suVPC.

Crear un VPC punto final de interfaz para HealthLake;

Puedes crear un VPC punto de conexión para el servicio HealthLake; mediante la VPC consola de Amazon o con AWS Command Line Interface (AWS CLI). Para obtener más información, consulte Creación de un punto final de interfaz en la Guía del VPC usuario de Amazon.

Cree un VPC punto final para HealthLake; utilizando el siguiente nombre de servicio:

- com.amazonaws. *region*.healthlake

Si activas la opción privada DNS en el punto final, puedes API solicitar que se HealthLake utilice su DNS nombre predeterminado para la región. Por ejemplo, *healthlake.us-east-1.amazonaws.com*.

Para obtener más información, consulte Acceder a un servicio a través de un punto final de interfaz en la Guía del VPC usuario de Amazon.

Crear una política VPC de puntos finales para HealthLake

Puede adjuntar una política de puntos finales a su VPC punto final que controle el acceso a HealthLake. La política especifica la siguiente información:

- La entidad principal que puede realizar acciones.
- Las acciones que se pueden realizar.
- Los recursos en los que se pueden llevar a cabo las acciones.

Para obtener más información, consulta Cómo controlar el acceso a los servicios con VPC puntos de conexión en la Guía del VPC usuario de Amazon.

Ejemplo: política de VPC puntos finales para la adopción de medidas HealthLake

El siguiente es un ejemplo de una política de puntos finales para HealthLake. Cuando se adjunta a un punto final, esta política otorga acceso a la HealthLake `CreateFHIRDatastore` acción a todos los directores de todos los recursos.

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

Etiquetado de recursos en AWS HealthLake

Puede asignar metadatos a los recursos de AWS en forma de etiquetas. Cada etiqueta es una marca que consta de una clave y un valor definidos por el usuario. Las etiquetas pueden ayudarle a administrar, identificar, organizar, buscar y filtrar recursos.

En este tema se describen las categorías y estrategias de etiquetado de uso común para ayudarle a implementar una estrategia de etiquetado coherente y eficaz. En las siguientes secciones se presupone un conocimiento básico de AWS los recursos, el etiquetado, la facturación detallada e AWS Identity and Access Management (IAM).

Cada etiqueta de tiene dos partes:

- Una clave de etiqueta (por ejemplo CostCenter, entorno o proyecto). Las claves de etiqueta distinguen entre mayúsculas y minúsculas.
- Un valor de etiqueta (por ejemplo, 111122223333 o Producción). Al igual que las claves de etiqueta, los valores de etiqueta distinguen entre mayúsculas y minúsculas.

Utilice etiquetas para clasificar los recursos según su finalidad, propietario, entorno u otro criterio. Para obtener más información, consulte Estrategias de etiquetado de AWS.

Puede agregar, cambiar o eliminar etiquetas un recurso a la vez desde la consola de servicio, el servicio API o el. AWS CLI

Para habilitar el etiquetado, asegúrese de que TagResources están autorizados. Puede autorizarlo TagResources adjuntando una IAM política como la del siguiente ejemplo.

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

```
        }
    ]
}
```

Aviso importante

AWS HealthLake protege los datos de los clientes según las políticas del modelo de responsabilidad AWS compartida. Esto significa que todos los datos de los clientes están cifrados tanto en transición como en reposo. Sin embargo, no todos los nombres introducidos por los clientes para los almacenes de datos o las operaciones basadas en tareas están cifrados. Nunca deben contener información de identificación personal o información de salud protegida. Para obtener más información, consulte el capítulo AWS HealthLake de seguridad.

Prácticas recomendadas

Cuando cree una estrategia de etiquetado para los recursos de AWS, siga las prácticas recomendadas:

- No almacene información de identificación personal (PII), información de salud personal (PHI) u otra información confidencial en las etiquetas.
- Utilice un formato estandarizado que distinga mayúsculas de minúsculas para las etiquetas y aplíquelo de forma coherente a todos los tipos de recursos.
- Tenga en cuenta las directrices de etiquetas que admiten múltiples propósitos, como administrar el control de acceso a recursos, el seguimiento de costos, la automatización y la organización.
- Use herramientas automatizadas para ayudar a administrar las etiquetas de recursos. AWS Resource Groups y Resource Groups Tagging API permiten el control programático de las etiquetas, lo que permite administrar, buscar y filtrar etiquetas y recursos automáticamente.
- El etiquetado es más eficaz cuando se utilizan más etiquetas.
- Las etiquetas se pueden editar o modificar según las necesidades del usuario. Sin embargo, para actualizar las etiquetas de control de acceso, también debe actualizar las políticas que hacen referencia a esas etiquetas para controlar el acceso a sus recursos.

Requisitos de etiquetado

Las etiquetas tienen los siguientes requisitos:

- Las claves no pueden llevar el prefijo aws:.
- Las claves deben ser únicas dentro de un conjunto de etiquetas.
- Una clave debe tener entre 1 y 128 caracteres permitidos.
- Un valor debe tener entre 0 y 256 caracteres permitidos.
- No es necesario que los valores sean únicos dentro de un conjunto de etiquetas.
- Los caracteres permitidos para las claves y los valores son letras y dígitos Unicode, espacios en blanco y cualquiera de estos símbolos: _ . : / = + - @.
- Las claves y los valores distinguen entre mayúsculas y minúsculas.

Añadir una etiqueta a un almacén de datos

Agregar etiquetas a un banco de datos puede ayudarle a identificar y organizar sus AWS recursos y a administrar el acceso a ellos. En primer lugar, agregue una o más etiquetas (pares clave-valor) a un banco de datos. Puede utilizar hasta cincuenta etiquetas por usuario. También hay restricciones en cuanto a los caracteres que puede utilizar en los campos de clave y valor.

Una vez que tenga las etiquetas, puede crear IAM políticas para administrar el acceso al banco de datos en función de estas etiquetas. Puede usar la HealthLake consola o la AWS CLI para agregar etiquetas a un banco de datos. Añadir etiquetas a un repositorio puede afectar al acceso a dicho repositorio. Antes de añadir una etiqueta a un banco de datos, asegúrese de revisar IAM las políticas que puedan utilizar etiquetas para controlar el acceso a recursos como los almacenes de datos.

Siga estos pasos para usar el AWS CLI para agregar una etiqueta a un banco HealthLake de datos. Para agregar una etiqueta a un banco de datos al crearlo, consulteCrear un almacén de datos en AWS HealthLake.

En la terminal o en la línea de comandos, ejecute el comando tag-resource y especifique el nombre del recurso de Amazon (ARN) del almacén de datos al que desea añadir las etiquetas y la clave y el valor de la etiqueta que desea añadir. Puede añadir más de una etiqueta a un banco de datos. También hay restricciones en cuanto a los caracteres que puede utilizar en los campos de clave y valor, tal y como se indica en Por ejemplo, Requisitos de etiquetado para añadir etiquetas a un banco de datos mientras se está creando, debería utilizar el siguiente comando en el AWS CLI. El nombre del banco de datos es Test_Data_Store y las dos etiquetas añadidas con claves son clave1 y clave2 con valores como valor1 y valor2, respectivamente :

```
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
```

Para añadir etiquetas a un banco de datos existente, ejecute el siguiente comando de ejemplo:

```
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
```

Si se ejecuta correctamente, este comando no devuelve ninguna respuesta.

Listar las etiquetas de un banco de datos

Siga estos pasos para utilizarlos AWS CLI para ver una lista de las AWS etiquetas de un almacén HealthLake de datos. Si no se han añadido etiquetas, la lista obtenida está vacía.

En la terminal o en la línea de comandos, ejecute el list-tags-for-resource comando como se muestra en el siguiente ejemplo.

```
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"
    }
}
```

Eliminar etiquetas de un almacén de datos

Puede eliminar una o más etiquetas asociadas a un banco de datos. La eliminación de una etiqueta no la elimina de otros recursos de AWS que están asociados con esa etiqueta.

En la terminal o en la línea de comandos, ejecute el comando untag-resource y especifique el nombre del recurso de Amazon (ARN) del almacén de datos del que desea eliminar las etiquetas y la clave de etiqueta de la etiqueta que desea eliminar.

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

Si se ejecuta correctamente, este comando no devuelve ninguna respuesta. Para comprobar las etiquetas asociadas al banco de datos, ejecute el list-tags-for-resource comando.

Monitorización HealthLake

La supervisión es una parte importante del mantenimiento de la confiabilidad, la disponibilidad y el rendimiento de HealthLake AWS las demás soluciones. AWSproporciona las siguientes herramientas de monitoreo para observar HealthLake, informar cuando algo anda mal y tomar medidas automáticas cuando sea apropiado:

- Amazon CloudWatch monitorea tus AWS recursos y las aplicaciones en las que AWS ejecutas en tiempo real. Puede recopilar métricas y realizar un seguimiento, crear paneles personalizados y configurar alarmas que le notifiquen o tomen medidas cuando una métrica específica alcance un umbral específico. Por ejemplo, puedes CloudWatch hacer un seguimiento CPU del uso u otras métricas de tus EC2 instancias de Amazon y lanzar automáticamente nuevas instancias cuando sea necesario. Para obtener más información, consulta la Guía del CloudWatch usuario de Amazon.
- AWS CloudTrailcaptura API las llamadas y los eventos relacionados realizados por tu AWS cuenta o en su nombre. A continuación, entrega los archivos log al bucket de Amazon S3 que se especifique. Puedes identificar qué usuarios y cuentas llamaron AWS, la dirección IP de origen de esas llamadas y cuándo se produjeron. Para obtener más información, consulte la AWS CloudTrail Guía del usuario de .

Temas

- Monitorización HealthLake con Amazon CloudWatch

Monitorización HealthLake con Amazon CloudWatch

Puede monitorizar el HealthLake uso CloudWatch, que recopila datos sin procesar y los procesa para convertirlos en métricas legibles prácticamente en tiempo real. Estas estadísticas se mantienen durante 15 meses, de modo que pueda consultar información histórica y disponer de una mejor perspectiva sobre el desempeño de su aplicación web o servicio. También puede establecer alarmas que vigilen determinados umbrales y enviar notificaciones o realizar acciones cuando se cumplan dichos umbrales. Para obtener más información, consulta la Guía del CloudWatch usuario de Amazon.

Se informa de todas las métricas HealthLake APIs, incluidas las siguientes.

- Administración de almacenes de datosAPIs: CreateFHIRDatastore, DeleteFHIRDatastore, DescribeFHIRDatastore, L istFHIRDatastores
- Importación y exportaciónAPIs: S tartFHIRImport Job, L istFHIRImport Jobs, D escribeFHIRImport Job, S tartFHIRExport Job, L istFHIRExport Jobs, D escribeFHIRExport Job
- HTTPRESTGestión de clientes y recursosAPIs: CreateResource, DeleteResource, GetCapabilities, ReadResource, SearchAll, SearchWithGet SearchWithPost, UpdateResource.
- Etiquetado APIs — ListTagsForResource,, TagResource UntagResource

En las siguientes tablas se muestran las métricas y dimensiones de HealthLake.

Se reportan las siguientes métricas. Cada una se presenta como un recuento de frecuencias para un rango de datos especificado por el usuario.

Métricas

Métricas	Descripción
Conteo de llamadas	El número de llamadas aAPIs. que se puede informar, o bien para la cuenta, o bien para un almacén de datos específico. Unidades: recuento Estadísticas válidas: Sum, Count Dimensiones: funcionamiento, ID del almacén de datos, tipo de almacén de datos
Solicitudes satisfactorias	El número de API solicitudes aceptadas. Unidades: recuento Estadísticas válidas: Sum, Average Dimensiones: funcionamiento, almacén de datos, tipo de almacén de datos
Errores de usuario	El número de solicitudes que fallaron debido a un error del usuario.

Métricas	Descripción
	Unidades: recuento Estadísticas válidas: Sum, Average Dimensiones: funcionamiento, ID del almacén de datos, tipo de almacén de datos
Errores de servidor	El número de solicitudes que fallaron debido a un error del servidor. Unidades: recuento Estadísticas válidas: Sum, Average Dimensiones: funcionamiento, ID del almacén de datos, tipo de almacén de datos
Solicitudes limitadas	El número de solicitudes que se han limitado. Esta métrica no se incluye en los recuentos de errores del usuario o del servidor. Unidades: recuento Estadísticas válidas: Sum, Average Dimensiones: funcionamiento, ID del almacén de datos, tipo de almacén de datos
Latencia	El tiempo en milisegundos que se tardó en procesar la solicitud del usuario. Unidad: milisegundos Estadísticas válidas: mínimo, máximo, promedio Dimensiones: funcionamiento, ID del almacén de datos, tipo de almacén de datos

Se indican las siguientes dimensiones.

Dimensiones

Dimensiones	Descripción
Operación	¿Qué API operación se utilizó
DataStoreID	El almacén de datos incluido en la API solicitud
DataStoreType	El tipo de almacén de datos (actualmente solo se admite FHIR R4)

Puede obtener métricas HealthLake con la consola AWS de administración AWS CLI, el o el CloudWatch API. Puede utilizarlos CloudWatch API a través de uno de los kits de desarrollo de AWS software de Amazon (SDKs) o de las CloudWatch API herramientas. La HealthLake consola muestra gráficos basados en los datos sin procesar del CloudWatch API.

Debe tener los CloudWatch permisos adecuados para poder monitorizar HealthLake con CloudWatch. Para obtener más información, consulta Autenticación y control de acceso para Amazon CloudWatch en la Guía del CloudWatch usuario de Amazon.

Visualización de HealthLake las métricas

Para ver las métricas (CloudWatch consola)

1. Inicie sesión en la consola de administración de AWS y abra la consola de CloudWatch.
2. Elija Métricas, elija Todas las métricas y, a continuación, elija AWS/HealthLake.
3. Elija la dimensión, un nombre de métrica y, a continuación, Add to graph (Añadir al gráfico).
4. Elija un valor para el intervalo de fechas. El recuento de las métricas del intervalo de fechas seleccionado se muestra en el gráfico.

Crear una alarma mediante CloudWatch

Una CloudWatch alarma vigila una única métrica durante un período de tiempo específico y realiza una o más acciones: enviar una notificación a un tema de Amazon Simple Notification Service (AmazonSNS) o a una política de Auto Scaling. La acción o las acciones se basan en el valor de la métrica en relación con un umbral determinado durante un número de períodos de tiempo que usted

especifique. CloudWatch también puede enviarte un SNS mensaje de Amazon cuando la alarma cambia de estado.

CloudWatch las alarmas invocan acciones solo cuando el estado cambia y ha persistido durante el período que especifiques.

Para ver las métricas (consola) CloudWatch

1. Inicie sesión en la consola de administración de AWS y abra la consola de CloudWatch.
2. Elija Alarms y, a continuación, seleccione Create Alarm.
3. Elija AWS/yHealthLake, a continuación, elija una métrica.
4. En Time Range, elija un intervalo de tiempo para monitorizar y, a continuación, elija Next.
5. Introduzca un Nombre y una Descripción.
6. En Whenever, elija >= y escriba un valor máximo.
7. Si quieres CloudWatch enviar un correo electrónico cuando se alcance el estado de alarma, en la sección Acciones, en Siempre que se produzca esta alarma, selecciona Estado es ALARM. En Enviar notificación a, selecciona una lista de correo o selecciona Nueva lista y crea una nueva lista de correo.
8. Obtenga una vista previa de la alarma en la sección Vista previa de la alarma. Si está satisfecho con la alarma, elija Create Alarm.

SMARTIntegrándose FHIR con AWS HealthLake

Las aplicaciones médicas sustituibles y las tecnologías reutilizables (SMART) en un almacén de HealthLake datos FHIR habilitado permiten que SMART las aplicaciones FHIR compatibles accedan a los datos almacenados en un almacén de HealthLake datos. HealthLake Se accede a los datos autenticando y autorizando las solicitudes mediante un servidor de autorización externo y configurando recursos adicionales en. AWS

Para usarlo FHIR con SMART su almacén de HealthLake datos, debe proporcionar lo siguiente en su solicitud C. reateFHIRDatastore API

- Establece el AuthorizationStrategyvalor igual a`SMART_ON_FHIR_V1`.
- Establezca un IdpLambdaArn valor igual al ARN AWS Lambda que creó para administrar la decodificación de los tokens con su servidor de autorización.
- Defina los elementos de metadatos especificados en su servidor de autorización. Estos elementos de metadatos se devuelven en el documento de descubrimiento. Para obtener más información, consulte Obtener un documento SMART de descubrimiento de un banco de HealthLake datos FHIR activado.
- Opcional: FineGrainedAuthorizationEnabledactívela si ha configurado una autorización detallada en su servidor de autorización.

Puede crear un almacén de datos SMART FHIR activado mediante AWS Command Line Interface (AWS CLI) o mediante uno de los AWS compatiblesSDKs. No se admite SMART la creación de un banco de HealthLake datos FHIR activado mediante la HealthLake consola. Para obtener más información, consulte Cree un SMART banco de datos FHIR activado.

Para especificar estos parámetros en la solicitud, debe configurar los recursos en otros AWS servicios (AWS Secrets Manager y AWS Lambda), crear nuevas funciones de IAM servicio y configurar un SMART servidor de autorización que no FHIR cumpla con los requisitos. Utilice la sección Configuración de los recursos necesarios para implementar un banco SMART de datos FHIR compatible para obtener más información sobre la configuración de los recursos necesarios y para ver una descripción general de alto nivel de cómo interactúa una SMART FHIR aplicación activa. HealthLake

Esto significa que, en lugar de administrar las credenciales de los usuarios a través de AWS Identity and Access Management usted, lo hace utilizando un SMART servidor de autorización que no FHIR cumpla con las normas.

HealthLake es compatible con SMART la FHIR versión 1.0. Para obtener más información sobre este marco, consulte la Guía de implementación de SMART Application Launch Framework, versión 1.0.

Para autorizar y autenticar las solicitudes de almacenamiento de datos mediante SMART onFHIR, HealthLake admite el uso de:

- Integración con OpenID (AuthN): se utiliza para autenticar que esa persona o aplicación cliente es quien (o qué) dice ser.
- OAuthIntegración 2.0 (AuthZ): se utiliza para autorizar qué FHIR recursos del almacén de HealthLake datos también pueden leer o escribir datos una solicitud autenticada. Esto se define mediante los ámbitos configurados en el servidor de autorización

Contenido

Requisitos de autenticación para SMART FHIR

Para acceder a FHIR los recursos de un SMART almacén de FHIR HealthLake datos interno, una aplicación cliente debe estar autorizada por un servidor de autorización OAuth compatible con la versión 2.0 y presentar un token de OAuth portador como parte de la solicitud. FHIR REST API Para encontrar el punto final del servidor de autorización, utilice el documento HealthLake SMART on FHIR Discovery mediante un identificador uniforme de recursos muy conocido. Para obtener más información sobre este proceso, consulte Obtener un documento SMART de descubrimiento de un banco de HealthLake datos FHIR activado.

Al crear un almacén SMART de FHIR HealthLake datos activo, debe definir el punto final del servidor de autorización y el punto final del token en el `metadata` elemento de la reateFHIRDatastore solicitud C. Para obtener más información sobre la definición del `metadata` elemento, consulteCreación de un SMART banco de HealthLake datos no FHIR habilitado.

Mediante los puntos finales del servidor de autorización, la aplicación cliente autenticará a un usuario con el servicio de autorización. Una vez autorizado y autenticado, el servicio de autorización genera un JSON Web Token (JWT) y lo pasa a la aplicación cliente. Este token contiene los ámbitos de FHIR recursos que la aplicación cliente puede utilizar, lo que a su vez restringe los datos a los que puede acceder el usuario. Opcionalmente, si se proporcionó el alcance del lanzamiento, la respuesta contendrá esos detalles. Para obtener más información SMART sobre los FHIR alcances compatibles HealthLake, consulteSoportado SMART en FHIR OAuth osciloscopios por HealthLake.

Utilizando lo JWT otorgado por el servidor de autorización, una aplicación cliente realiza FHIR REST API llamadas a un almacén de HealthLake datos SMART FHIR activado. Para validarla y

JWT decodificarla, debe crear una función Lambda. HealthLake invoca esta función Lambda en su nombre cuando recibe FHIR REST API una solicitud. Para ver un ejemplo de función Lambda de inicio, consulte. AWS Lambda Utilización para la validación de tokens con un SMART almacén HealthLake de datos FHIR activado

Elementos del servidor de autorización necesarios para crear un SMART almacén de HealthLake datos FHIR activado

En la reateFHIRDatastore solicitud C, debes proporcionar el punto final de autorización y el punto final del token como parte del `metadata` elemento del `IdentityProviderConfiguration` objeto. Se requieren tanto el punto final de autorización como el punto final del token. Para ver un ejemplo de cómo se especifica esto en una reateFHIRDatastore solicitud de C, consulteCreación de un SMART banco de HealthLake datos no FHIR habilitado.

Reclamaciones obligatorias para completar una FHIR REST API solicitud SMART en un almacén de HealthLake datos no FHIR habilitado

Su AWS Lambda función debe incluir las siguientes afirmaciones para que sea una FHIR REST API solicitud válida SMART en un almacén de HealthLake datos no FHIR habilitado.

- `nbf`: Reclamación (no anterior): la reclamación «nbf» (no anterior) identifica el momento en que JWT MUST NOT debe aceptarse su tramitación. La tramitación de la reclamación «nbf» exige que la reclamación actual date/time MUST be after or equal to the not-before date/time figure en la reclamación «nbf». La función Lambda de ejemplo que proporcionamos convierte la respuesta `iat` del servidor en. `nbf`
- `exp`: (fecha de caducidad): la afirmación «exp» (fecha de caducidad) identifica la fecha de caducidad a partir de la cual no se JWT debe aceptar para su procesamiento.
- `isAuthorized`: Un booleano establecido en. `True` Indica que la solicitud se ha autorizado en el servidor de autorización.
- `aud`Afirmación: (audiencia): la afirmación «aud» (audiencia) identifica a los destinatarios a los que JWT va destinada. Debe ser un punto final SMART de almacén HealthLake de datos FHIR activado.
- `scope`: Debe ser al menos un ámbito relacionado con los FHIR recursos. Este ámbito se define en el servidor de autorización. Para obtener más información sobre FHIR los ámbitos relacionados con los recursos aceptados por HealthLake, consulteHealthLake alcances específicos de FHIR los recursos del almacén de datos.

Soportado SMART en FHIR OAuth osciloscopios por HealthLake

HealthLake utiliza OAuth 2.0 como protocolo de autorización. El uso de este protocolo en su servidor de autorización le permite definir a qué FHIR recursos del almacén de HealthLake datos también puede tener acceso de lectura o escritura una aplicación cliente.

El FHIR marco SMART on define un conjunto de ámbitos que se pueden solicitar al servidor de autorización. Para ver las definiciones de alcance en el FHIR marco SMART on, consulte la sección SMARTsobre los FHIR ámbitos en la Guía de HL7FHIRrecursos.

Por ejemplo, una aplicación cliente que solo esté diseñada para permitir a los pacientes ver sus resultados de laboratorio o ver sus datos de contacto solo debería estar autorizada a solicitar (mediante FHIR REST solicitud) `read` osciloscopios. Para definirlos como ámbito de aplicación, debe proporcionar una cadena como la siguiente`patient/Observation.read`. Esto permitiría a la aplicación cliente solicitar acceso al tipo de `Observation` recurso de solo lectura en el tipo de `Patient` recurso.

Alcance de lanzamiento independiente

HealthLake admite el ámbito del modo de lanzamiento independiente. `launch/patient`

En el modo de inicio independiente, una aplicación cliente solicita acceso a los datos clínicos del paciente porque la aplicación cliente no conoce al usuario ni al paciente. Por lo tanto, la solicitud de autorización de la aplicación cliente solicita explícitamente que se devuelva la sonda del paciente. Tras una autenticación correcta, el servidor de autorización emite un token de acceso que contiene el osciloscopio de pacientes de lanzamiento solicitado. El contexto del paciente necesario se proporciona junto con el token de acceso en la respuesta del servidor de autorización.

Alcances del modo de lanzamiento compatibles

Ámbito	Descripción
`launch/patient`	Parámetro de una solicitud de autorización OAuth 2.0 que solicita que se devuelvan los datos del paciente en la respuesta de autorización.

HealthLake alcances específicos de FHIR los recursos del almacén de datos

HealthLake define tres niveles de ámbitos.

- Los alcances específicos para cada paciente permiten el acceso a datos específicos sobre un solo paciente. Qué paciente se especifica en el contexto del lanzamiento.
- Los ámbitos a nivel de usuario permiten el acceso a datos específicos a los que puede acceder un usuario.
- Los ámbitos a nivel del sistema otorgan acceso de lectura y escritura a todos los recursos que se encuentran en el almacén de datosFHIR. HealthLake

En la siguiente tabla se muestra la sintaxis para construir los ámbitos relacionados con FHIR los recursos compatibles con. HealthLake El formato general es el siguiente:

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

Ámbitos de autorización admitidos en los almacenes HealthLake de datos

Sintaxis del ámbito	Ejemplo de ámbito	Resultado
`patient/(fhir-resource | '*'). ('read' | 'write' | '*')`	`patient/A llergyInt olerance.*`	Una aplicación cliente tendría acceso de lectura/escritura a las alergias.
`user/(fhir-resource | '*').('re ad' | 'write' | '*')`	`user/Obse rvation.read`	Una aplicación cliente tendría acceso de lectura a todas las observaciones registradas.
`system/('read' | 'write' | *)`	`system/*.*`	Una aplicación cliente tendría acceso de lectura/escritura a todos los datos.

AWS Lambda Utilización para la validación de tokens con un SMART almacén HealthLake de datos FHIR activado

Al crear un banco SMART de HealthLake datos que no FHIR esté habilitado, debe proporcionar ARN la AWS Lambda función en la `CreateFHIRDatastore` solicitud. La función Lambda ARN se especifica en el `IdentityProviderConfiguration` objeto mediante el `IdpLambdaArn` parámetro.

Debe crear la función Lambda antes de crear su propio banco SMART de HealthLake datos FHIR habilitado. Una vez creado el banco de datos, la Lambda ARN no se puede cambiar. Para ver la Lambda ARN que especificó cuando se creó el banco de datos, utilice la `DescribeFHIRDatastore` API operación.

Para que una FHIR REST solicitud se realice correctamente SMART en un almacén de HealthLake datos FHIR activado, la función Lambda debe hacer lo siguiente:

- La función Lambda debe devolver una respuesta en menos de 1 segundo al punto final del almacén de HealthLake datos.
- Decodifique el token de acceso proporcionado en el encabezado de autorización de la REST API solicitud enviada por la aplicación cliente.
- Asigne un rol de IAM servicio que tenga permisos suficientes para llevar a cabo la FHIR REST API solicitud.
- Se requieren las siguientes solicitudes para completar una FHIR REST API solicitud. Para obtener más información, consulte Reclamaciones obligatorias.
  - `nbf`
  - `exp`
  - `isAuthorized`
  - `aud`
  - `scope`

Al trabajar con Lambda, además de la función Lambda, debe crear un rol de ejecución y una política basada en recursos. El rol de ejecución de una función de Lambda es aquel que otorga a la función permiso para acceder a los AWS servicios y recursos necesarios en tiempo de ejecución. IAM La política basada en recursos que proporcione debe permitir HealthLake invocar su función en su nombre.

En las secciones de este tema se describe un ejemplo de solicitud de una aplicación cliente y una respuesta decodificada, los pasos necesarios para crear una función AWS Lambda y cómo crear una política basada en recursos que se pueda asumir. HealthLake

- Parte 1: Creación de una función Lambda
- Parte 2: Crear un rol HealthLake de servicio utilizado por la función AWS Lambda
- Parte 3: Actualización del rol de ejecución de la función Lambda
- Parte 4: Añadir una política de recursos a la función Lambda
- Parte 5: Aprovisionamiento simultáneo para su función Lambda

Creación de una AWS función Lambda

La función Lambda creada en este tema se activa cuando HealthLake recibe una solicitud a un banco de HealthLake datos SMART FHIR activado. La solicitud de la aplicación cliente contiene una REST API llamada y un encabezado de autorización que contiene un token de acceso.

```
GET https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/
Authorization: Bearer i8hweunweunweofiwweoijewiwe
```

El ejemplo de la función Lambda de este tema se utiliza AWS Secrets Manager para ocultar las credenciales relacionadas con el servidor de autorización. Recomendamos encarecidamente no proporcionar los detalles de inicio de sesión del servidor de autorización directamente en una función Lambda.

Example validar una FHIR REST solicitud que contenga un token portador de autorización

La función Lambda de ejemplo muestra cómo validar una FHIR REST solicitud enviada a un HealthLake data SMART store FHIR activado. Para ver step-by-steps instrucciones sobre cómo implementar esta función Lambda, consulte. Creación de una función Lambda mediante el AWS Management Console

Si la FHIR REST API solicitud no contiene un punto final del almacén de datos, un token de acceso y una REST operación válidos, la función Lambda fallará. Para obtener más información sobre los elementos del servidor de autorización necesarios, consulteReclamaciones obligatorias.

```
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()
```

Creación de una función Lambda mediante el AWS Management Console

En este procedimiento se supone que ya ha creado la función de servicio que desea asumir HealthLake al gestionar una FHIR REST API solicitud SMART en un banco de HealthLake datos no FHIR habilitado. Si no ha creado el rol de servicio, aún puede crear la función Lambda. Deberá añadir la función ARN de servicio para que la función Lambda funcione. Para obtener más información sobre cómo crear un rol de servicio y especificarlo en la función Lambda, consulte: Crear un rol HealthLake de servicio para usarlo en la función AWS Lambda utilizada para decodificar un JWT

Para crear una función Lambda ()AWS Management Console

1. Abra la página de Functions (Funciones) en la consola de Lambda.
2. Seleccione Crear función.
3. Seleccione Crear desde cero.
4. En Información básica, introduzca un nombre de función. En Tiempo de ejecución, elija un tiempo de ejecución basado en Python.
5. En Execution role (Rol de ejecución), elija Create a new role with basic Lambda permissions (Crear un nuevo rol con permisos básicos de Lambda).

   Lambda crea un rol de ejecución que otorga a la función permiso para cargar registros en Amazon. CloudWatch La función Lambda asume la función de ejecución cuando se invoca la función y utiliza la función de ejecución para crear credenciales para la. AWS SDK
6. Seleccione la pestaña Código y añada la función Lambda de ejemplo.

Si aún no ha creado el rol de servicio que va a utilizar la función Lambda, tendrá que crearlo antes de que funcione la función Lambda de ejemplo. Para obtener más información sobre la creación de un rol de servicio para la función Lambda, consulte. Crear un rol HealthLake de servicio para usarlo en la función AWS Lambda utilizada para decodificar 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()
```

Modificación del rol de ejecución de una función Lambda

Tras crear la función Lambda, debe actualizar el rol de ejecución para incluir los permisos necesarios para llamar a Secrets Manager. En Secrets Manager, cada secreto que crees tiene unARN. Para aplicar el mínimo de privilegios, la función de ejecución solo debe tener acceso a los recursos necesarios para que se ejecute la función Lambda.

Puede modificar la función de ejecución de una función de Lambda buscándola en la IAM consola o seleccionando Configuración en la consola de Lambda. Para obtener más información sobre la administración de su función de ejecución de funciones de Lambda, consulte. Rol de ejecución de Lambda

Example Función de ejecución de la función Lambda que otorga acceso a **`GetSecretValue`**

Al añadir la IAM acción `GetSecretValue` a la función de ejecución, se otorga el permiso necesario para que funcione la función Lambda de muestra.

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "secretsmanager:GetSecretValue",
      "Resource": "arn:aws:secretsmanager:your-region:your-aws-account-
id:secret:secret-name-DKodTA"
    }
  ]
}
```

En este punto, ha creado una función Lambda que se puede utilizar para validar el token de acceso proporcionado como parte de la FHIR REST solicitud enviada a su almacén SMART de HealthLake datos FHIR habilitado.

Crear un rol HealthLake de servicio para usarlo en la función AWS Lambda utilizada para decodificar un JWT

Persona: administrador IAM

Un usuario que puede añadir o eliminar IAM políticas y crear nuevas IAM identidades.

Rol de servicio

Un rol de servicio es un rol de IAM que asume un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener más información, consulte Crear un rol para delegar permisos a un Servicio de AWS en la Guía del IAM usuario.

Una vez decodificado el JSON Web Token (JWT), Lambda también necesita obtener una autorización para devolver IAM un rol. ARN Este rol debe tener los permisos necesarios para llevar a cabo la REST API solicitud o fallará por falta de permisos.

Al configurar una política personalizadaIAM, es mejor conceder los permisos mínimos necesarios. Para obtener más información, consulte Aplicar permisos con privilegios mínimos en la Guía del IAM usuario.

La creación HealthLake de un rol de servicio para designarlo en la función Lambda de autorización requiere dos pasos.

- En primer lugar, debe crear una IAM política. La política debe especificar el acceso a los FHIR recursos para los que ha proporcionado ámbitos en el servidor de autorización.
- En segundo lugar, debe crear el rol de servicio. Al crear el rol, designa una relación de confianza y adjunta la política que creó en el primer paso. La relación de confianza se designa HealthLake como principal del servicio. En este paso, debe especificar un almacén de HealthLake datos ARN y un identificador de AWS cuenta.

Crear una nueva IAM política

Los ámbitos que defina en el servidor de autorización determinan a qué FHIR recursos tiene acceso un usuario autenticado en un almacén de HealthLake datos.

La IAM política que cree se puede personalizar para que coincida con los ámbitos que haya definido.

Se pueden definir las siguientes acciones en el `Action` elemento de una declaración de IAM política. Para cada uno `Action` de los elementos de la tabla, puede definir un`Resource types`. En HealthLake un banco de datos es el único tipo de recurso admitido que se puede definir en el `Resource` elemento de una declaración de política de IAM permisos.

Los FHIR recursos individuales no son un recurso que se pueda definir como un elemento de una política de IAM permisos.

Acciones definidas por HealthLake

Acciones	Descripción	Nivel de acces	Tipo de recurso (obligatorio)
CreateResource	Otorga permiso para crear un recurso	Escrit	Almacén de datosARN: arn:aws:healthlake: ::datastore/fhir/ **your-region 111122223333 your-datastore-id**
DeleteResource	Otorga permiso para eliminar un recurso	Escrit	Almacén de datosARN: arn:aws:healthlake: **your-region** ::datastore/fhir/ **111122223333 your-datastore-id**
ReadResource	Otorga permiso para leer el recurso	Lectu	Almacén de datosARN: arn:aws:healthlake: **your-region** ::datastore/fhir/ **111122223333 your-datastore-id**

Acciones	Descripción	Nivel de acces	Tipo de recurso (obligatorio)
SearchWithGet	Otorga permiso para buscar recursos con el método GET	Lectu	Almacén de datosARN: arn:aws:healthlake: ::datastore/fhir/ **your-region 111122223333 your-datastore-id**
SearchWithPost	Otorga permiso para buscar recursos con el método POST	Lectu	Almacén de datosARN: arn:aws:healthlake: ::datastore/fhir/ **your-region 111122223333 your-datastore-id**
StartFHIRExportJobWithPost	Otorga permiso para iniciar un trabajo de FHIR exportación con GET	Escrit	Almacén de datosARN: arn:aws:healthlake: ::datastore/fhir/ **your-region 111122223333 your-datastore-id**
UpdateResource	Otorga permiso para actualizar el recurso	Escrit	Almacén de datosARN: arn:aws:healthlake: **your-region** ::datastore/fhir/ **111122223333 your-datastore-id**

Para empezar, puede utilizar. `AmazonHealthLakeFullAccess` Esta política permitiría leer, escribir, buscar y exportar todos los FHIR recursos que se encuentren en un almacén de datos. Para conceder permisos de solo lectura en un almacén de datos, utilice. `AmazonHealthLakeReadOnlyAccess`

Para obtener más información sobre cómo crear una política personalizada mediante, o AWS Management Console AWS CLI IAMSDKs, consulte Creación de IAM políticas en la Guía del IAMusuario.

Crear un rol de servicio para HealthLake (IAMconsola)

Utilice este procedimiento para crear un rol de servicio. Al crear un servicio, también necesitará designar una IAM política.

Para crear el rol de servicio para HealthLake (IAMconsola)

1. Inicie sesión en AWS Management Console y abra la IAM consola en https://console.aws.amazon.com/iam/.
2. En el panel de navegación de la consola de IAM, elija Roles (Funciones).
3. A continuación, elija Create role (Crear rol).
4. En la página Seleccionar entidad de confianza, elija Política de confianza personalizada.
5. A continuación, en Política de confianza personalizada, actualice la política de ejemplo de la siguiente manera. Sustituya **`your-account-id`** por su número de cuenta y añada el ARN del banco de datos que desee utilizar en sus trabajos de importación o exportación.

```
{
  "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. A continuación, elija Siguiente.
7. En la página Añadir permisos, elija la política que desee que asuma el HealthLake servicio. Para encontrar tu política, búscala en Políticas de permisos.
8. A continuación, selecciona Adjuntar política.
9. A continuación, en la página Nombre, revisión y creación, situada en Nombre del rol, introduzca un nombre.

10. (Opcional) A continuación, en Descripción, añade una breve descripción de tu función.
11. De ser posible, escriba un nombre o sufijo de nombre para el rol, que pueda ayudarle a identificar su finalidad. Los nombres de los roles deben ser únicos en su nombre Cuenta de AWS. No distinguen entre mayúsculas y minúsculas. Por ejemplo, no puede crear funciones denominado tanto **`PRODROLE`** como **`prodrole`**. Dado que varias entidades pueden hacer referencia al rol, no puede editar el nombre del rol después de crearlo.
12. Revise los detalles del rol y, a continuación, seleccione Crear rol.

Para obtener información sobre cómo especificar la función ARN en la función Lambda de ejemplo, consulte. Creación de una AWS función Lambda

Rol de ejecución de Lambda

El rol de ejecución de una función Lambda es un IAM rol que otorga a la función permiso para acceder a los AWS servicios y recursos. Esta página proporciona información sobre cómo crear, ver y administrar el rol de ejecución de una función de Lambda.

De forma predeterminada, Lambda crea un rol de ejecución con permisos mínimos al crear una nueva función de Lambda mediante. AWS Management Console Para administrar los permisos concedidos en la función de ejecución, consulte Creación de una función de ejecución en la IAM consola en la Guía para desarrolladores de Lambda.

La función Lambda de ejemplo que se proporciona en este tema usa Secrets Manager para ocultar las credenciales del servidor de autorización.

Al igual que con cualquier IAM rol que cree, es importante seguir las prácticas recomendadas con el mínimo privilegio. Durante la frase de desarrollo, a veces es posible que concedas permisos más allá de lo necesario. Antes de publicar la función en el entorno de producción, la práctica recomendada consiste en ajustar la política para incluir solo los permisos necesarios. Para obtener más información, consulte Aplicar los privilegios mínimos en la Guía del usuario. IAM

Permita HealthLake activar su función Lambda

Para HealthLake poder invocar la función Lambda en su nombre, debe hacer lo siguiente:

- Debe establecer un `IdpLambdaArn` valor igual al ARN de la función Lambda que desea HealthLake invocar en la solicitud. `CreateFHIRDatastore`

- Necesita una política basada en recursos que le permita HealthLake invocar la función Lambda en su nombre.

Cuando HealthLake recibe una FHIR REST API solicitud SMART en un HealthLake data store FHIR activado, necesita permisos para invocar en su nombre la función Lambda especificada en la creación del data store. Para conceder el HealthLake acceso, utilizará una política basada en los recursos. Para obtener más información sobre cómo crear una política basada en recursos para una función de Lambda, consulte Permitir que un AWS servicio llame a una función de Lambda en la Guía para desarrolladores.AWS Lambda

Aprovisionamiento simultáneo para su función Lambda

> **Important**
>
> HealthLake requiere que el tiempo máximo de ejecución de la función Lambda sea inferior a un segundo (1000 milisegundos).
> Si la función Lambda supera el límite de tiempo de ejecución, se produce una TimeOutexcepción.

Para evitar esta excepción, se recomienda configurar la simultaneidad aprovisionada. Mediante la asignación de la simultaneidad aprovisionada antes de un aumento en las invocaciones, puede asegurarse de que todas las solicitudes se atiendan mediante instancias inicializadas con una latencia baja. Para obtener más información sobre la configuración de la simultaneidad aprovisionada, consulte Configuración de la simultaneidad aprovisionada en la Guía para desarrolladores de Lambda

Para ver el tiempo medio de ejecución de la función Lambda en la actualidad, utilice la página de supervisión de la función Lambda en la consola de Lambda. De forma predeterminada, la consola Lambda proporciona un gráfico de duración que muestra la cantidad media, mínima y máxima de tiempo que el código de función dedica a procesar un evento. Para obtener más información sobre la supervisión de las funciones de Lambda, consulte Supervisión de las funciones en la consola de Lambda en la Guía para desarrolladores de Lambda.

Si ya ha aprovisionado la simultaneidad para la función de Lambda y desea supervisarla, consulte Supervisión de la simultaneidad en la Guía para desarrolladores de Lambda.

Creación de un SMART banco de HealthLake datos no FHIR habilitado

Para usar el FHIR marco SMART on HealthLake, cree un banco de HealthLake datos con el `IdentityProviderConfiguration` parámetro especificado en su reateFHIRDatastore solicitud C. En el `IdentityProviderConfiguration` parámetro, especifique la siguiente información:

- Establezca el AuthorizationStrategyvalor igual a`SMART_ON_FHIR_V1`.
- Establezca un IdpLambdaArnvalor igual al ARN AWS Lambda que creó para administrar la decodificación de los tokens con su servidor de autorización.
- Defina los elementos de metadatos especificados en el servidor de autorización como un JSON bloque. Estos elementos de metadatos se devuelven en el documento de descubrimiento.
- Opcional: activar FineGrainedAuthorizationEnabled. Especifique `True` el uso de la autorización detallada proporcionada por HealthLake

Puede crear un almacén de datos SMART FHIR activado mediante AWS Command Line Interface (AWS CLI) o mediante uno de los AWS compatiblesSDKs. No se admite SMART la creación de un banco de HealthLake datos FHIR activado mediante la HealthLake consola.

Utilización de AWS CLI para crear un SMART banco de HealthLake datos FHIR activado

Puede usar el siguiente ejemplo de código para crear un SMART FHIR banco HealthLake de datos habilitado mediante AWS CLI. Al crear un SMART banco de HealthLake datos FHIR activado, debe especificar el `identity-provider-configuration`parámetro.

En el `identity-provider-configuration` parámetro, si lo desea, puede habilitar una autorización detallada estableciendo `FineGrainedAuthorizationEnabled` un valor igual a. `True` Para obtener más información sobre la autorización detallada, consulte. Uso de una autorización detallada con un almacén de datos no SMART habilitado FHIR HealthLake El siguiente ejemplo contiene un carácter especial \ para indicar saltos de línea o como carácter de escape. Esto es para mayor claridad.

```
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\"]}"}'
```

Si tiene éxito, obtendrá la siguiente JSON respuesta:

```
{
  "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"
}
```

Uso de una autorización detallada con un almacén de datos no SMART habilitado FHIR HealthLake

Los ámbitos por sí solos no proporcionan la especificidad necesaria sobre los datos a los que el solicitante está autorizado a acceder en un almacén de datos. El uso de una autorización detallada permite un mayor nivel de especificidad al conceder acceso a un SMART almacén de datos no habilitado. FHIR HealthLake Para usar una autorización

detallada, establezca un `FineGrainedAuthorizationEnabled` valor igual a `True` en el `IdentityProviderConfiguration` parámetro de su solicitud C. reateFHIRDatastore

Si habilitaste la autorización detallada, tu servidor de autorización devuelve un `fhirUser` ámbito junto con el `id_token` token de acceso. Esto permite que la aplicación cliente recupere la información sobre el usuario. La aplicación cliente debe tratar la `fhirUser` reclamación como la URI de un FHIR recurso que representa al usuario actual. Este valor puede ser `Patient`, `Practitioner` o `RelatedPerson`. La respuesta del servidor de autorización también incluye un `user/` ámbito que define a qué datos puede acceder el usuario. Utiliza la sintaxis definida para los ámbitos relacionados con los ámbitos específicos del FHIR recurso:

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

Los siguientes son ejemplos de cómo se puede utilizar la autorización detallada para especificar con más detalle los tipos de recursos relacionados con el acceso a los datos. FHIR

- Cuándo `fhirUser` es una autorización detalladaPractitioner, determina el conjunto de pacientes a los que puede acceder el usuario. Solo `fhirUser` se permite el acceso a aquellos pacientes en los que el paciente tenga referencias `fhirUser` como médico generalista.

  ```
  Patient.generalPractitioner : [{Reference(Practitioner)}]
  ```

- Cuando `fhirUser` es un `Patient` o `RelatedPerson` y el paciente al que se hace referencia en la solicitud es diferente del `fhirUser` indicado, la autorización detallada determina el acceso del paciente `fhirUser` solicitado. Se permite el acceso cuando existe una relación especificada en el recurso solicitado. `Patient`

  ```
  Patient.link.other : {Reference(Patient|RelatedPerson)}
  ```

Obtener un documento SMART de descubrimiento de un banco de HealthLake datos FHIR activado

Para que una aplicación cliente pueda realizar una FHIR REST solicitud correctamente, debe reunir los requisitos de autorización definidos en el almacén de HealthLake datos. No se requiere ninguna autorización (token de portador) para que esta solicitud se realice correctamente.

Para ello, haga una `GET` solicitud y añádala `/.well-known/smart-configuration` al punto final del almacén de datos

```
GET https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/.well-known/smart-configuration
```

Esto devuelve el documento de descubrimiento del almacén de HealthLake datos como un JSON blob. En él, encontrará el `authorization_endpoint` almacén de datos `token_endpoint` junto con las especificaciones y capacidades definidas en el almacén de HealthLake datos.

```
{
  "authorization_endpoint": "https://oidc.example.com/authorize",
  "token_endpoint": "https://oidc.example.com/oauth/token",
  "capabilities": [
    "launch-ehr",
    "client-public"
  ]
}
```

URLsnecesario para lanzar una aplicación cliente correctamente

- Punto final de autorización: el URL necesario para autorizar una aplicación cliente o un usuario.
- Punto final simbólico: punto final del servidor de autorización que la aplicación cliente utiliza para comunicarse con él.

Realizar una FHIR REST API solicitud en un almacén de HealthLake datos SMART habilitado

Puede realizar FHIR REST API solicitudes SMART en un almacén FHIR de HealthLake datos activado. El siguiente ejemplo muestra una solicitud de una aplicación cliente que contiene un JWT en el encabezado de autorización y cómo Lambda debe decodificar la respuesta. Una vez autorizada y autenticada la solicitud de la aplicación cliente, debe recibir un token portador del servidor de autorización. Utilice el token portador en el encabezado de autorización cuando envíe una FHIR REST API solicitud a un almacén FHIR de HealthLake datos SMART activado.

```
GET https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/Patient/[ID]
Authorization: Bearer auth-server-provided-bearer-token
```

Como se encontró un token de portador en el encabezado de autorización y no se detectó ninguna AWS IAM identidad, se HealthLake invoca la función Lambda especificada cuando se creó el HealthLake data SMART store FHIR activado. Cuando la función Lambda decodifica correctamente el token, aquí hay un ejemplo de respuesta que se envía a. 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
}
```

Configurar los recursos necesarios para implementar un SMART almacén de datos que no FHIR cumpla con las normas

En este tema se describen los recursos que debe aprovisionar en su AWS cuenta externa HealthLake, la creación de un almacén de HealthLake datos SMART FHIR activado y la forma SMART en que una aplicación FHIR cliente interactuaría con un servidor de autorización y un almacén de HealthLake datos.

Los pasos de este flujo de trabajo definen los pasos básicos SMART sobre cómo se gestionan FHIR las solicitudes y qué recursos se necesitan para que tengan éxito.

SMARTEn un proceso FHIR a pedido, tres aplicaciones funcionan juntas:

- El usuario final: por lo general, un paciente o un médico que utiliza una FHIR aplicación de terceros SMART para acceder a los datos de un almacén de HealthLake datos.
- La SMART FHIR aplicación interna (denominada aplicación cliente): una aplicación que desea acceder a los datos que se encuentran en el almacén de HealthLake datos.

- El servidor de autorización: un servidor compatible con OpenID Connect que puede autenticar a los usuarios y emitir tokens de acceso.
- El almacén de HealthLake datos: un almacén de HealthLake datos SMART FHIR activado que utiliza una función Lambda para responder a FHIR REST las solicitudes que proporcionan un token portador.

Para que estas aplicaciones funcionen juntas, es necesario crear los siguientes recursos.

Se recomienda crear el almacén de HealthLake datos SMART FHIR activado una vez que haya configurado el servidor de autorización, definido los ámbitos necesarios y creado una AWS Lambda función para gestionar la introspección de los símbolos.

1. Configuración de un punto final del servidor de autorización: servidor de autorización

Para usar el FHIR marco SMART on, debe configurar un servidor de autorización de terceros que pueda validar FHIR REST las solicitudes realizadas en un almacén de datos. Para obtener más información sobre cómo configurar un punto final de servidor de autorización que funcione con HealthLake él, consulteRequisitos de autenticación para SMART FHIR.

2. Defina los ámbitos para controlar quién puede acceder a qué datos de su almacén de HealthLake datos en su servidor de autorización (servidor de autorización)

El FHIR marco SMART on utiliza los OAuth ámbitos para determinar a qué FHIR recursos tiene acceso una solicitud autenticada y en qué medida. Definir los ámbitos es una forma de diseñar pensando en los privilegios mínimos. Para obtener más información sobre los ámbitos definidos por el FHIR marco SMART on y compatibles con, consulte,. HealthLake Soportado SMART en FHIR OAuth osciloscopios por HealthLake

3. Configura una AWS Lambda función capaz de realizar una introspección simbólica: tu cuenta AWS

Una FHIR REST solicitud enviada por la aplicación cliente a un SMART almacén de datos no FHIR habilitado contendrá un token JSON web (). JWT Para obtener más información sobre cómo configurar una función Lambda capaz de decodificarla y validarla, consulte Decodificación de un. JWT

4. Cree un almacén SMART de HealthLake datos no FHIR habilitado: su cuenta AWS

Para crear un almacén SMART de FHIR HealthLake datos propio, debe proporcionar un`IdentityProviderConfiguration`. Para obtener más información sobre

`IdentityProviderConfiguration` los parámetros necesarios en una reateFHIRDatastore solicitud de C, consulte Creación de un banco de HealthLake datos SMART FHIR activado.

Cómo se inicia y solicita datos una aplicación cliente desde un SMART almacén FHIR de HealthLake datos activado

En esta sección se explica cómo se inicia una aplicación cliente SMART en un FHIR contexto y cómo puede realizar una FHIR REST solicitud correcta en un almacén de HealthLake datos.

1. La aplicación cliente realiza una **`GET`** solicitud a Known Uniform Resource Identifier

Una aplicación cliente SMART habilitada debe realizar una `GET` solicitud para encontrar los puntos finales de autorización de su almacén de HealthLake datos. Esto se hace mediante una solicitud de identificador uniforme de recursos (URI) bien conocido. Para obtener más información al respecto, consulte Obtener un documento de descubrimiento SMART de un banco de HealthLake datos que FHIR esté activado.

2. Solicitar el acceso y los alcances

La aplicación cliente utiliza el punto final de autorización del servidor de autorización para que el usuario pueda iniciar sesión. Este proceso autentica al usuario. Los ámbitos se utilizan para definir a qué FHIR recursos del almacén de HealthLake datos puede acceder una aplicación cliente. Para obtener más información sobre la definición de los ámbitos, consulte. Soportado SMART en FHIR OAuth osciloscopios por HealthLake

3. Tokens de acceso

Ahora que el usuario se ha autenticado, una aplicación cliente recibe un token de JWT acceso del servidor de autorización. Este token se proporciona cuando la aplicación cliente envía una FHIR REST solicitud a HealthLake. Para obtener más información sobre cómo JWT se decodifica mediante una función Lambda, consulte. Realizar la validación del token

4. Realizar una FHIR REST solicitud en un almacén SMART de datos FHIR habilitado HealthLake

Ahora, la aplicación cliente puede enviar una FHIR REST solicitud a un punto final del almacén de HealthLake datos utilizando el token de acceso proporcionado por el servidor de autorización. Para ver un ejemplo de FHIR REST solicitud, consulteRealizar una FHIR REST API solicitud en un almacén de HealthLake datos SMART habilitado.

5. Validar el token de JWT acceso

Para validar el token de acceso enviado en la FHIR REST solicitud, utilice una función Lambda. Para ver cómo crear una función Lambda que pueda realizar una introspección simbólica, consulte. Creación de una AWS función Lambda

Uso de la generación automática de recursos basada en el procesamiento del lenguaje natural (NLP) del tipo de FHIR DocumentReference recurso en AWS HealthLake

> **Note**
>
> Después del 20 de febrero de 2023, HealthLake los almacenes de datos no utilizan el procesamiento de lenguaje natural integrado (NLP) de forma predeterminada. Si está interesado en activar esta función en su almacén de datos, consulte el capítulo ¿Cómo activo la función HealthLake de procesamiento de lenguaje natural integrada? de solución de problemas.
>
> Si has activado la versión integrada de Amazon Comprehend MedicalNLP, cuando crees o actualices `DocumentReference` recursos, se te cobrarán cargos en tu cuenta. AWS Para obtener más información, consulta AWS HealthLake los precios.
>
> Amazon Comprehend Medical no está disponible en Asia Pacífico (Bombay). HealthLake Los almacenes de datos creados en la región de Asia Pacífico (Bombay) no admiten el procesamiento integrado del lenguaje natural (NLP).

HealthLake le proporciona automáticamente un procesamiento de lenguaje natural integrado (NLP) mediante Amazon Comprehend Medical para el procesamiento de datos no estructurados de los datos almacenados en `DocumentReference` el tipo de recurso. Para ello, HealthLake llama a Amazon Comprehend `DetectEntities-V2` Medical`InferICD10-CM`, `InferRxNorm` API y a sus operaciones. Los resultados se añaden automáticamente al `DocumentReference` recurso como una extensión. Cuando las operaciones de Amazon Comprehend API Medical detectan rasgos que `SIGN` son`SYMPTOM`, `DIAGNOSIS` y, se genera `Linkage` automáticamente un tipo de recurso. Los nuevos recursos de observación y estado se crean a partir de entidades identificadas con los rasgos de `SIGN`, `SYMPTOM`, o`DIAGNOSIS`, y se enlazan al documento fuente con este recurso de enlace.

En el caso de los recursos generados por el sistema integrado`NLP`, puede realizar `GET` solicitudes, pero no se admite la búsqueda de estos nuevos recursos.

Para obtener más información sobre la búsqueda de estas extensiones mediante HealthLake la integración con Athena, consulte. Consulte su almacén HealthLake de datos mediante SQL

Contenido

Cómo se integra Amazon Comprehend Medical con HealthLake

HealthLake deduce los datos encontrados en el tipo de `DocumentReference` recurso mediante Amazon Comprehend Medical. Los Amazon Comprehend API Medical `DetectEntities-V2` Operations `InferICD10-CM` `InferRxNorm` y detectan las afecciones médicas como rasgos. Cada operación proporciona información diferente.

> ⚠ Lenguajes compatibles
>
> Las operaciones de Amazon Comprehend API Medical solo detectan entidades médicas en textos en inglés.

- DetectEntities-V2: inspecciona el texto clínico para detectar diversas entidades médicas y devuelve información específica sobre ellas, como la categoría de la entidad, la ubicación y el puntaje de confianza.
- Inferir el ICD1 0-CM: detecta las afecciones médicas en el registro de un paciente como entidades y vincula esas entidades con los identificadores de conceptos normalizados de la base de conocimientos del ICD -10-CM del Centro CDC Nacional de Estadísticas de Salud, con la autorización de la Organización Mundial de la Salud (). WHO
- InferRxNorm: Detecta los medicamentos como entidades incluidas en la historia clínica de un paciente y los vincula a los identificadores conceptuales normalizados de la base de datos de la RxNorm Biblioteca Nacional de Medicina.

Los rasgos compatibles para cada API operación son `SIGNSYMPTOM`, y. `DIAGNOSIS` Si se detectan rasgos, se agregan como extensiones FHIR compatibles a diferentes ubicaciones del almacén de HealthLake datos.

Ubicaciones donde se agregan las extensiones.

- `DocumentReference`: Los resultados de las operaciones de Amazon Comprehend API Medical se añaden como `extension` a cada documento que se encuentra dentro DocumentReference del tipo de recurso. Los resultados de la extensión se dividen en dos grupos. Puede encontrarlos en los resultados en función de sus`URL`.
  - `http://healthlake.amazonaws.com/system-generated-resources/`
    - Estos son tipos de recursos que han sido creados o añadidos por HealthLake.
  - `http://healthlake.amazonaws.com/aws-cm/`
    - Donde el resultado bruto de las operaciones de Amazon Comprehend API Medical se añade a HealthLake su almacén de datos.
- `Linkage`: Este tipo de recurso se agrega o se crea como resultado de la integraciónNLP. Una `GET` solicitud sobre un objeto específico `Linkage` devuelve una lista de recursos enlazados. Para identificar si a `Linkage` fue agregado por HealthLake, busca el par `"tag": [{"display": "SYSTEM_GENERATED"}]` clave-valor agregado. Para obtener más información sobre las FHIR especificaciones de la vinculación, consulte Tipo de recurso: Vinculación en el índice de documentación. FHIR
- FHIRtipos de recursos generados como resultado de las operaciones de Amazon Comprehend API Medical.
  - `Observation`: Se le añaden los resultados de las DetectEntities operaciones de Amazon Comprehend API Medical -V2 e ICD1 Infer 0-CM cuando los rasgos son o. `SIGN` `SYMPTOM`
  - `Condition`: Se le añaden los resultados de las DetectEntities operaciones de Amazon Comprehend API Medical -V2 e ICD1 Infer 0-CM cuando los rasgos son. `DIAGNOSIS`
  - `MedicationStatement`: Se le han añadido los resultados de la InferRxNorm operación Amazon Comprehend API Medical.

Integración con las operaciones FHIR REST API

De forma predeterminada, las características detectadas por las operaciones de Amazon Comprehend API Medical no se devuelven al realizar `GET` una solicitud.

Para ver los resultados de las NLP operaciones integradas para estos tipos de recursos, debe especificar uno conocido`ID`.

- `Linkage`

- `Observation`
- `Condition`
- `MedicationStatement`

Los resultados de las NLP operaciones integradas fuera del tipo de DocumentReference recurso solo están disponibles mediante una `GET` solicitud en la que `ID` se sepa que lo especificado contiene resultados de las operaciones de Amazon Comprehend API Medical.

Ejemplos de cómo se integran las operaciones de Amazon Comprehend API Medical en HealthLake

Ejemplo 1: registro de un paciente ingresado en un HealthLake almacén de datos

Este es un ejemplo de nota clínica basada en el encuentro de un paciente con un profesional médico.

> ⚠ Datos sintéticos
>
> El texto de este ejemplo es contenido sintético y no contiene información de salud personal (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)
```

Como recordatorio, esta información está codificada en formato base64 en el DocumentReference recurso. Cuando haya ingerido este documento HealthLake y se hayan completado API las operaciones de Amazon Comprehend Medical, para ver los resultados, puede empezar con la `GET` solicitud del tipo de recurso. `DocumentReference`

```
 GET https://https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/
r4/eeb8005725ae22b35b4edbdc68cf2dfd/r4/DocumentReference
```

Cuando las operaciones de Amazon Comprehend API Medical tengan éxito, busque estos pares clave-valor dentro `extension` del enlace a lo siguiente `"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."
}
```

Las siguientes pestañas muestran cómo se registra la historia clínica ingresada en su almacén de HealthLake datos en función del tipo de recurso.

DocumentReference

Para ver los resultados de un solo tipo de `DocumentReference` recurso, realice una `GET` solicitud en la que se proporcione el `id` de un recurso específico.

```
GET https://https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/eeb8005725ae22b35b4edbdc68cf2dfd/r4/DocumentReference/0e938f03-da7f-4178-acd8-eea9586c46ed
```

Si tiene éxito, obtendrá un código de `200` HTTP respuesta y la siguiente JSON respuesta (que se ha truncado para mayor claridad).

Esta es la parte. `http://healthlake.amazonaws.com/system-generated-resources/` Puede ver que se `Linkage/`*`e366d29f-2c22-4c19-866e-09603937935a`* ha agregado una nueva. También puede ver dónde se HealthLake han agregado los hallazgos basados en inferencias a tipos específicos `Observation` y de `Condition` recursos.

Para ver cómo se han modificado estos tipos de recursos, selecciona las pestañas relacionadas.

```
{
  "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

Para ver los resultados de un solo tipo de `Linkage` recurso, realiza una `GET` solicitud en la que se proporcione el `ID` de un recurso específico.

```
GET https://https://healthlake.your-region.amazonaws.com/
datastore/your-datastore-id/r4/eeb8005725ae22b35b4edbdc68cf2dfd/r4/
Linkage/e366d29f-2c22-4c19-866e-09603937935a
```

Si se ejecuta correctamente, obtendrá un código de `200` HTTP respuesta y la siguiente respuesta truncada. JSON

La respuesta contiene el `item` elemento. En él, el par clave-valor `"type": "source"` indica la `DocumentReference` entrada específica utilizada para modificar el par clave-valor `Condition` y que `Observations` aparece debajo del par `"type": "alternate"` clave-valor.

También puede ver el *`meta`* elemento y el par clave-valor correspondiente, lo que indica que estos recursos fueron *`"tag": [{"display": "SYSTEM_GENERATED"}]`* creados por. 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

Para ver los resultados de un solo tipo de `Observation` recurso, realiza una `GET` solicitud en la que se proporcione el `ID` de un recurso específico.

```
GET https://https://healthlake.your-region.amazonaws.com/
datastore/your-datastore-id/r4/eeb8005725ae22b35b4edbdc68cf2dfd/r4/
Observation/e366d29f-2c22-4c19-866e-09603937935a
```

Los resultados de las operaciones de Amazon Comprehend API Medical se modifican para incluir los siguientes elementos`code`:`meta`,, `modifierExtension` y.

`code`

Un elemento de tipo`CodeableConcept`. Para obtener más información, consulte CodeableConceptel índice de FHIR documentación.

HealthLake anexa los tres pares clave-valor siguientes.

- `"system": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/"`: Donde se URL refiere a una operación específica de Amazon Comprehend API Medical. En este caso, inferir 0 cm. ICD1
- `"code": "A52.06"`: ¿Dónde `A52.06` está el código ICD -10-CM que identifica el concepto que se encuentra en la base de conocimientos de los Centros para el Control de Enfermedades?
- `"display": "Other syphilitic heart involvement"`: ¿Dónde `"Other syphilitic heart involvement"` está la descripción larga del código ICD -10-CM en la ontología?

La siguiente JSON respuesta truncada contiene solo el elemento. `code`

```
"code": {
  "coding":
  [
    {
   "system": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/",
   "code": "A52.06",
   "display": "Other syphilitic heart involvement"
    }
  ],
  "text": "Other syphilitic heart involvement"
 }
```

Para entender la confianza del modelo en cuanto a que el código de ICD 10 CM asignado es correcto, utilice el elemento. `modifierExtension`

`meta`

El `meta` elemento contiene metadatos que indican si el `code` elemento contiene detalles añadidos por las operaciones de Amazon Comprehend API Medical.

La siguiente JSON respuesta truncada contiene solo el elemento. `meta`

```
"meta": {
  "lastUpdated": "2022-10-21T19:38:30.879Z",
  "tag": [{
   "display": "SYSTEM_GENERATED"
  }]
```

```
 }
```

modifierExtension

El `modifierExtension` elemento contiene más detalles sobre el nivel de confianza de los códigos asignados que se encuentran en el `code` elemento. También tiene pares clave-valor que proporcionan un enlace al original DocumentReference utilizado para generar los resultados y al tipo de recurso de enlace relacionado.

Para cada `coding` elemento agregado, verá una `entity-score` y una `entity-Concept-Score` agregada a. modifierExtension Para cada valor del par clave-valor, verá una puntuación. Pues`entity-score`, esta puntuación es el nivel de confianza que Amazon Comprehend Medical tiene en la precisión de la detección. `entity-Concept-Score`Pues esta puntuación es el nivel de confianza que Amazon Comprehend Medical tiene en cuanto a que la entidad está vinculada con precisión al concepto de ICD -10-CM.

La siguiente JSON respuesta truncada contiene solo el elemento. `modifierExtension`

```
"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"
   }
  }
 ]
```

Respuesta completa JSON


```
{
  "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

Para ver los resultados de un solo tipo de `Condition` recurso, realiza una `GET` solicitud en la que se proporcione el `ID` de un recurso específico.

```
GET https://https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/eeb8005725ae22b35b4edbdc68cf2dfd/r4/Condition/b06d343d-ddb8-4f36-82cb-853fcd434dfd
```

Los resultados de las operaciones de Amazon Comprehend API Medical se modifican para incluir los siguientes elementos`code`:`meta`,, `modifierExtension` y.

`code`

Un elemento de tipo`CodeableConcept`. Para obtener más información, consulte CodeableConceptel índice de FHIR documentación.

HealthLake anexa los tres pares clave-valor siguientes.

- `"system": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/"`: Donde se URL refiere a una operación específica de Amazon Comprehend API Medical. En este caso, inferir 0 cm. ICD1
- `"code": "I70.0"`: ¿Dónde `A52.06` está el código ICD -10-CM que identifica el concepto que se encuentra en la base de conocimientos de los Centros para el Control de Enfermedades?
- `"display": "Atherosclerosis of aorta"`: ¿Dónde `"Other syphilitic heart involvement"` está la descripción larga del código ICD -10-CM en la ontología?

La siguiente JSON respuesta truncada contiene solo el elemento. `code`

```
"code": {
  "coding":
  [
    {
   "system": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/",
```

```
   "code": "I70.0",
   "display": "Atherosclerosis of aorta"
    }
  ],
  "text": "Atherosclerosis of aorta"
 }
```

Para entender la confianza del modelo en cuanto a que el código de ICD 10 CM asignado es correcto, utilice el elemento. `modifierExtension`

`meta`

El `meta` elemento contiene metadatos que indican si el `code` elemento contiene detalles añadidos por las operaciones de Amazon Comprehend API Medical.

La siguiente JSON respuesta truncada contiene solo el elemento. `meta`

```
"meta": {
  "lastUpdated": "2022-10-21T19:38:30.877Z",
  "tag": [{
   "display": "SYSTEM_GENERATED"
  }]
 }
```

`modifierExtension`

El `modifierExtension` elemento contiene más detalles sobre el nivel de confianza de los códigos asignados que se encuentran en el `code` elemento. También tiene pares clave-valor que proporcionan un enlace al original DocumentReference utilizado para generar los resultados y al tipo de recurso de enlace relacionado.

Para cada `coding` elemento agregado, verá una `entity-score` y una `entity-Concept-Score` agregada a. modifierExtension Para cada valor del par clave-valor, verá una puntuación. Pues`entity-score`, esta puntuación es el nivel de confianza que Amazon Comprehend Medical tiene en la precisión de la detección. `entity-Concept-Score`Pues esta puntuación es el nivel de confianza que Amazon Comprehend Medical tiene en cuanto a que la entidad está vinculada con precisión al concepto de ICD -10-CM.

La siguiente JSON respuesta truncada contiene solo el elemento. `modifierExtension`

```
"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"
   }
  }
 ]
```

Respuesta completa JSON

```
{
  "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"
 }
```

Ejemplo 2: Una **`DocumentReference`** que contiene un tipo MedicationStatement de recurso

Este es un ejemplo de una nota clínica basada en el encuentro de un paciente con un profesional médico.

> ⚠ Datos sintéticos
>
> El texto de este ejemplo es contenido sintético y no contiene información de salud personal (PHI).

```
Tom is not prescribed Advil
```

En las siguientes pestañas, se muestra cómo se registra el registro médico ingerido en el almacén de HealthLake datos en función del tipo de recurso.

DocumentReference

Para ver los resultados de un solo tipo de `DocumentReference` recurso, realice una `GET` solicitud en `ID` la que se proporcione un recurso específico.

```
GET https://https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/eeb8005725ae22b35b4edbdc68cf2dfd/r4/DocumentReference/c549125d-a218-421f-b8bf-23614c5e796c
```

Si tiene éxito, obtendrá un código de `200` HTTP respuesta y la siguiente respuesta truncada. JSON

El par clave-valor,`"url": "http://healthlake.amazonaws.com/system-generated-resources/"`, indica que las operaciones de Amazon API Comprehend Medical `extension` agregaron los tipos de recursos que contiene. Puede ver el nuevo tipo de `Linkage` recurso y varios recursos. `MedicationStatement`

```
"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

Para ver los resultados de un solo tipo de `Linkage` recurso, realiza una `GET` solicitud en `ID` la que se proporcione un recurso específico.

```
GET https://https://healthlake.your-region.amazonaws.com/
datastore/your-datastore-id/r4/eeb8005725ae22b35b4edbdc68cf2dfd/r4/
Linkage/394bb244-177b-4409-8657-26b20ed56dd7
```

Si tiene éxito, obtendrá un código de `200` HTTP respuesta y la siguiente JSON respuesta.

La respuesta contiene el `item` elemento. En él, el par clave-valor `"type": "source"` indica la `DocumentReference` entrada específica utilizada para modificar los tipos de `MedicationStatement` recursos.

También puede ver el `meta` elemento y el par clave-valor correspondiente`"tag": [{"display": "SYSTEM_GENERATED"}]`, lo que indica que estos recursos fueron creados por. 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

Para ver los resultados de un solo tipo de `MedicationStatement` recurso, realiza una `GET` solicitud en la que se proporcione el `ID` de un recurso específico.

```
GET https://https://healthlake.your-region.amazonaws.com/
datastore/your-datastore-id/r4/eeb8005725ae22b35b4edbdc68cf2dfd/r4/
MedicationStatement/9a89b0d3-6681-45ca-9926-27951edce5c7
```

El tipo de MedicationStatement recurso es donde se encuentran los resultados de la operación Amazon Comprehend InferRxNorm API Medical. Los resultados se modifican para incluir los siguientes elementos:`medicationCodeableConcept`,`meta`, y`modifierExtension`.

medicationCodeableConcept

Un elemento de tipo`CodeableConcept`. Para obtener más información, consulte CodeableConceptel índice de FHIR documentación.

HealthLake anexa los tres pares clave-valor siguientes.

- `"system": ""http://healthlake.amazonaws.com/aws-cm/infer-rxnorm/`: Donde se URL refiere a una operación específica de Amazon Comprehend API Medical. En este caso, InferRxNorm.
- `"code": "731533"`: ¿Dónde `731533` está un identificador RxNorm conceptual, también conocido como RxCUI?
- `"display": "ibuprofen 200 MG Oral Capsule [Advil]"`: ¿Dónde `ibuprofen 200 MG Oral Capsule [Advil]` está la descripción del RxNorm concepto?

La siguiente JSON respuesta truncada contiene solo el `MedicationStatement` elemento.

```
"medicationCodeableConcept": {
  "coding": [
   {
```

```
    "system": "http://healthlake.amazonaws.com/aws-cm/infer-rxnorm/",
    "code": "731533",
    "display": "ibuprofen 200 MG Oral Capsule [Advil]"
   }
  ]
 }
```

meta

El `meta` elemento contiene metadatos que indican si el `code` elemento contiene detalles añadidos por las operaciones de Amazon Comprehend API Medical.

La siguiente JSON respuesta truncada contiene solo el elemento. `meta`

```
"meta": {
  "lastUpdated": "2022-10-24T20:05:02.800Z",
  "tag": [
   {
    "display": "SYSTEM_GENERATED"
   }
  ]
 }
```

modifierExtension

El `modifierExtension` elemento contiene pares clave-valor que proporcionan un enlace al original DocumentReference utilizado para generar los resultados y al tipo de recurso de enlace relacionado.

```
"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"
    }
   }
```

```
]
```

Parámetros de búsqueda

En la siguiente tabla se enumeran los atributos de búsqueda de Integrated MedicalNLP.

Parámetros de búsqueda

Parámetros de búsqueda	Busca coincidencias para
detectEntities-entidad-categoría	Categoría de entidad dentro de la DetectEntities subextensión de la extensión CM AWS
detectEntities-entidad-texto	Texto de entidad dentro de la DetectEntities subextensión dentro de la extensión CM AWS
detectEntities-tipo de entidad	Tipo de entidad dentro de la DetectEntities subextensión dentro de la extensión CM AWS
detectEntities-puntuación de entidad	Entity Score dentro de la DetectEntities subextensión de la extensión CM AWS
infer-icd10 cm-entity-text	Texto de entidad dentro de la subextensión Infer ICD1 0CM dentro de la extensión CM AWS
infer-icd10 cm-entity-score	Puntuación de entidad dentro de la subextensión Infer ICD1 0CM dentro de la extensión CM AWS
infer-icd10 cm-entity-concept-code	Código conceptual de entidad dentro de la subextensión Infer ICD1 0CM dentro de la extensión CM AWS
infer-icd10 cm-entity-concept-description	Descripción del concepto de entidad dentro de la subextensión Infer ICD1 0CM dentro de la extensión CM AWS
infer-icd10 cm-entity-concept-score	Puntuación del concepto de entidad dentro de la subextensión Infer ICD1 0CM dentro de la extensión CM AWS

Parámetros de búsqueda	Busca coincidencias para
infer-rxnorm-entity-score	Puntuación de entidad dentro de la InferRxNorm subextensión de la extensión CM AWS
infer-rxnorm-entity-text	Texto de entidad dentro de la InferRxNorm subextensión dentro de la AWS extensión CM
infer-rxnorm-entity-concept-código	Código conceptual de entidad dentro de la InferRxNorm subextensión de la AWS extensión CM
infer-rxnorm-entity-concept-descripción	Descripción del concepto de entidad dentro de la InferRxNorm subextensión de la AWS extensión CM
infer-rxnorm-entity-concept-puntuación	Entity Concept Score dentro de la InferRxNorm subextensión de la AWS extensión CM

Para que coincidan con los criterios en los `EntityCategory` que la misma entidad es parte de ella `EntityText` y la forman parte, HealthLake proporciona una búsqueda especial. En la siguiente tabla se describen los parámetros de búsqueda especiales que se admiten en él HealthLake.

Parámetros de búsqueda

Parámetros de búsqueda	Coincidencias devueltas
detectEntities-entity-text-category	Si hay al menos una entidad en la DetectEntities subextensión que coincida con los valores entityText yentityCategory.
detectEntities-entity-type-score	Si hay al menos una entidad en la DetectEntities subextensión que coincida con los entityType valores y. entityScore
detectEntities-entity-text-score	Si hay al menos una entidad en la DetectEntities subextensión que coincida con los entityText valores y. entityScore

Parámetros de búsqueda	Coincidencias devueltas
detectEntities-entity-text-type	Si hay al menos una entidad en la DetectEntities subextensión que coincida con los entityText valores y. entityType
detectEntities-entity-category-score	Si hay al menos una entidad que coincida con los valores entityCategory yentityScore.
infer-icd10 -code cm-entity-text-concept	Si hay al menos una entidad en la subextensión Infer ICD1 0CM que coincide con el código entityText y hay al menos una conceptCode para esa entidad que coincide con el código.
infer-icd10 -score cm-entity-text-concept	Si hay al menos una entidad en la subextensión Infer ICD1 0CM que coincide con la puntuación entityTex t y hay al menos una conceptScore para esa entidad que coincide con la puntuación.
infer-icd10 -concept-score cm-entity-concept-description	Si hay al menos un concepto dentro de la entidad en la subextensión Infer ICD1 0CM que coincida con la descripción del concepto y con. conceptScore
infer-rxnorm-entity-text-código conceptual	Si hay al menos una entidad en la InferRxNorm subextensión que coincide con el código entityText y hay al menos una conceptCode para esa entidad que coincide con el código.
infer-rxnorm-entity-text-puntuación conceptual	Si hay al menos una entidad en la InferRxNorm subextensión que coincide con la puntuación entityText y hay al menos una conceptScore para esa entidad que coincide con la puntuación.
infer-rxnorm-entity-concept-description-concept-score	Si hay al menos un concepto dentro de la entidad de la InferRxNorm subextensión que coincida con la descripción del concepto y con el. conceptScore

Seguridad en AWS HealthLake

La seguridad en la nube AWS es la máxima prioridad. Como AWS cliente, usted se beneficia de una arquitectura de centro de datos y red diseñada para cumplir con los requisitos de las organizaciones más sensibles a la seguridad.

La seguridad es una responsabilidad compartida entre usted AWS y usted. El modelo de responsabilidad compartida la describe como seguridad de la nube y seguridad en la nube:

- La seguridad de la nube AWS es responsable de proteger la infraestructura que ejecuta AWS los servicios en la AWS nube. AWS también le proporciona servicios que puede utilizar de forma segura. Los auditores externos prueban y verifican periódicamente la eficacia de nuestra seguridad como parte de los AWS programas de de . Para obtener más información sobre los programas de cumplimiento aplicables HealthLake, consulte AWSServicios incluidos en el ámbito de aplicación por programa de conformidad y AWS servicios incluidos .
- Seguridad en la nube: su responsabilidad viene determinada por el AWS servicio que utilice. También eres responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y la normativa aplicables.

Esta documentación le ayuda a comprender cómo aplicar el modelo de responsabilidad compartida cuando se utiliza HealthLake. Los siguientes temas muestran cómo configurarlo HealthLake para cumplir sus objetivos de seguridad y conformidad. También aprenderá a utilizar otros AWS servicios que le ayudan a supervisar y proteger sus HealthLake recursos.

Temas

Protección de datos en AWS HealthLake

El modelo de responsabilidad AWS compartida modelo se aplica a la protección de datos en AWS HealthLake. Como se describe en este modelo, AWS es responsable de proteger la infraestructura global que ejecuta todos los Nube de AWS. Eres responsable de mantener el control sobre el contenido alojado en esta infraestructura. También eres responsable de las tareas de administración y configuración de seguridad para los Servicios de AWS que utiliza. Para obtener más información sobre la privacidad de los datos, consulte la sección Privacidad de datos FAQ. Para obtener información sobre la protección de datos en Europa, consulte el modelo de responsabilidad AWS compartida y la entrada del GDPR blog sobre AWS seguridad.

Para proteger los datos, le recomendamos que proteja Cuenta de AWS las credenciales y configure los usuarios individuales con AWS IAM Identity Center o AWS Identity and Access Management (IAM). De esta manera, solo se otorgan a cada usuario los permisos necesarios para cumplir sus obligaciones laborales. También recomendamos proteger sus datos de la siguiente manera:

- Utilice la autenticación multifactorial (MFA) con cada cuenta.
- UseSSL/TLSpara comunicarse con AWS los recursos. Necesitamos TLS 1.2 y recomendamos TLS 1.3.
- Configure API y registre la actividad del usuario con AWS CloudTrail. Para obtener información sobre el uso de CloudTrail senderos para capturar AWS actividades, consulte Cómo trabajar con CloudTrail senderos en la Guía del AWS CloudTrail usuario.
- Utilice soluciones de AWS cifrado, junto con todos los controles de seguridad predeterminados Servicios de AWS.
- Utilice servicios de seguridad gestionados avanzados, como Amazon Macie, que lo ayuden a detectar y proteger los datos confidenciales almacenados en Amazon S3.
- Si necesita entre FIPS 140 y 3 módulos criptográficos validados para acceder a AWS través de una interfaz de línea de comandos o unaAPI, utilice un FIPS terminal. Para obtener más información sobre los FIPS puntos finales disponibles, consulte la Norma federal de procesamiento de información () FIPS 140-3.

Se recomienda encarecidamente no introducir nunca información confidencial o sensible, como, por ejemplo, direcciones de correo electrónico de clientes, en etiquetas o campos de formato libre, tales como el campo Nombre. Esto incluye cuando trabaja con HealthLake o Servicios de AWS utiliza la consola, API AWS CLI, o. AWS SDKs Cualquier dato que ingrese en etiquetas o campos de formato libre utilizados para nombres se puedes emplear para los registros de facturación o diagnóstico. Si

proporciona una URL a un servidor externo, le recomendamos encarecidamente que no incluya la información sobre las credenciales URL para validar la solicitud a ese servidor.

Cifrado en REST forma AWS HealthLake

HealthLake proporciona cifrado de forma predeterminada para proteger los datos confidenciales de los clientes en reposo mediante una AWS clave propiedad del servicio de administración de claves (AWSKMS). También se admiten KMS las claves administradas por el cliente, que son necesarias tanto para importar como para exportar archivos de un almacén de datos. Para obtener más información sobre la KMS clave gestionada por el cliente, consulte Amazon Key Management Service. Los clientes pueden elegir una clave propia o una KMS KMS clave AWS administrada por el cliente al crear un almacén de datos. La configuración de cifrado no se puede cambiar una vez creado un almacén de datos. Si un almacén de datos utiliza AWS una KMS clave propia, se indicará como AWS_OWNED_KMS_KEY y no verá la clave específica utilizada para el cifrado en reposo.

AWSclave propia KMS

HealthLake utiliza estas claves de forma predeterminada para cifrar automáticamente la información potencialmente confidencial, como los datos de identificación personal o los datos de Private Health Information (PHI) en reposo. AWSKMSlas claves propias no se almacenan en tu cuenta. Forman parte de una colección de KMS claves que AWS posee y administra para su uso en varias AWS cuentas. AWSlos servicios pueden usar KMS claves AWS propias para proteger sus datos. No puede ver, administrar, usar KMS las claves AWS propias ni auditar su uso. Sin embargo, no es necesario que realice ninguna acción ni que cambie programas para proteger las claves que cifran sus datos.

No se te cobrará una cuota mensual ni una cuota de uso si utilizas KMS claves AWS propias, y estas no se tienen en cuenta para AWS KMS las cuotas de tu cuenta. Para obtener más información, consulta las claves AWS propias.

Claves KMS administradas por el cliente

HealthLake admite el uso de una KMS clave simétrica administrada por el cliente que usted crea, posee y administra para agregar una segunda capa de cifrado sobre el cifrado que ya AWS posee. Como usted tiene el control total de esta capa de cifrado, puede realizar tareas como las siguientes:

- Establecer y mantener políticas, IAM políticas y subvenciones clave
- Rotar el material criptográfico
- Habilitar y deshabilitar políticas de claves

- Agregar etiquetas.
- Crear alias de clave
- Programar la eliminación de claves

También puedes utilizarla CloudTrail para hacer un seguimiento de las solicitudes que se HealthLake envían AWS KMS en tu nombre. Se aplican AWS KMS cargos adicionales. Para obtener más información, consulta las claves propiedad del cliente.

Crear una clave administrada por el cliente

Puede crear una clave simétrica gestionada por el cliente mediante la consola AWS de gestión o el. AWS KMS APIs

Siga los pasos para crear una clave simétrica administrada por el cliente que se indican en la Guía para desarrolladores del servicio de administración de AWS claves.

Las políticas de clave controlan el acceso a la clave administrada por el cliente. Cada clave administrada por el cliente debe tener exactamente una política de clave, que contiene instrucciones que determinan quién puede usar la clave y cómo puede utilizarla. Cuando crea la clave administrada por el cliente, puede especificar una política de clave. Para obtener más información, consulte Administración del acceso a las claves administradas por el cliente en la Guía para desarrolladores de AWS Key Management Service.

Para utilizar la clave gestionada por el cliente con sus HealthLake recursos, la política clave debe permitir CreateGrant las operaciones de kms:. Esto añade una concesión a una clave gestionada por el cliente que controla el acceso a una KMS clave específica, lo que permite al usuario acceder a los requisitos de kms:grant para las operaciones. HealthLake Consulte Uso de subvenciones para obtener más información.

Para utilizar la KMS clave gestionada por el cliente con HealthLake los recursos, la política clave debe permitir las siguientes API operaciones:

- kms: CreateGrant añade las concesiones a una KMS clave específica gestionada por el cliente que permite el acceso a las operaciones de concesión.
- kms: DescribeKey proporciona los detalles de la clave gestionada por el cliente necesarios para validar la clave. Esto es necesario para todas las operaciones.
- kms: GenerateDataKey proporciona acceso a los recursos de cifrado en reposo para todas las operaciones de escritura.

- KMS:Decrypt proporciona acceso a las operaciones de lectura o búsqueda de recursos cifrados.

A continuación se muestra un ejemplo de declaración de política que permite a un usuario crear un almacén de datos cifrado con esa clave e interactuar con él: AWS HealthLake

```
"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"
            }
        }
    }
]
```

IAMPermisos necesarios para usar una KMS clave administrada por el cliente

Al crear un almacén de datos con el AWS KMS cifrado habilitado mediante una KMS clave administrada por el cliente, se requieren permisos tanto para la política de claves como para la IAM política del usuario o rol que crea el almacén de HealthLake datos.

Puede usar la clave de ViaService condición kms: para limitar el uso de la KMS clave únicamente a las solicitudes que se originen en HealthLake.

Para obtener más información sobre las políticas clave, consulte Habilitar IAM políticas en la Guía para desarrolladores del Servicio de administración de AWS claves.

El IAM usuario, el IAM rol o la AWS cuenta que crea tus repositorios debe tener los kms: CreateGrant kms: DescribeKey permisos kms: yGenerateDataKey, además, los HealthLake permisos necesarios.

¿Cómo se HealthLake utilizan las subvenciones en AWS KMS

HealthLake requiere una concesión para utilizar la KMS clave gestionada por el cliente. Al crear un almacén de datos cifrado con una KMS clave gestionada por el cliente, HealthLake crea una concesión en tu nombre enviando una CreateGrantsolicitud a AWSKMS. Las concesiones AWS KMS se utilizan para dar HealthLake acceso a una KMS clave de la cuenta de un cliente.

Las subvenciones que se HealthLake crean en tu nombre no se deben revocar ni retirar. Si revoca o retira la concesión que HealthLake autoriza el uso de las AWS KMS claves de su cuenta, HealthLake no puede acceder a estos datos, cifra los nuevos FHIR recursos introducidos en el almacén de datos o los descifra cuando se extraen. Al revocar o retirar una subvención HealthLake, el cambio se produce inmediatamente. Para revocar los derechos de acceso, debe eliminar el almacén de datos en lugar de revocar la concesión. Cuando se elimina un almacén de datos, HealthLake retira las concesiones en tu nombre.

Supervisión de las claves de cifrado para HealthLake

Puedes utilizarla CloudTrail para hacer un seguimiento de las solicitudes que se HealthLake envían AWS KMS en tu nombre cuando utilizas una KMS clave gestionada por el cliente. Las entradas del CloudTrail registro muestran healthlake.amazonaws.com en el userAgent campo para distinguir claramente las solicitudes realizadas por. HealthLake

Los siguientes ejemplos son CloudTrail eventos para CreateGrant descifrar y supervisar las AWS KMS operaciones solicitadas para acceder DescribeKey a los datos cifrados por HealthLake la clave gestionada por el cliente. GenerateDataKey

A continuación, se muestra cómo se utiliza CreateGrant para permitir el acceso HealthLake a una KMS clave proporcionada por el cliente, lo que HealthLake permite utilizar esa KMS clave para cifrar todos los datos inactivos del cliente.

Los usuarios no están obligados a crear sus propias subvenciones. HealthLake crea una subvención en tu nombre enviando una CreateGrant solicitud a AWSKMS. Las subvenciones AWS KMS se utilizan para dar HealthLake acceso a una AWS KMS clave de la cuenta de un cliente.

```
          {
    "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"
}
```

Los siguientes ejemplos muestran cómo se utilizan GenerateDataKey para garantizar que el usuario tenga los permisos necesarios para cifrar los datos antes de almacenarlos.

```
          {
    "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"
}
```

El siguiente ejemplo muestra cómo se HealthLake llama a la operación de descifrado para utilizar la clave de datos cifrados almacenada para acceder a los datos cifrados.

```
         {
    "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"
}
```

En el siguiente ejemplo, se muestra cómo se HealthLake utiliza la DescribeKey operación para comprobar si la AWS KMS clave propiedad del AWS KMS cliente se encuentra en un estado utilizable y para ayudar al usuario a solucionar problemas si no funciona.

```
         {
    "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"
}
```

Más información

Los siguientes recursos proporcionan más información sobre el cifrado de datos en reposo.

Para obtener más información sobre los conceptos básicos del Servicio de administración de AWS claves, consulte la AWS KMS documentación.

Para obtener más información sobre las prácticas recomendadas de seguridad, AWS KMS consulte la documentación.

Cifrado en tránsito para AWS HealthLake

AWS HealthLake utiliza TLS 1.2 para cifrar los datos en tránsito a través del punto final público y a través de los servicios de backend.

Administración de identidad y acceso para AWS HealthLake

AWS Identity and Access Management (IAM) es una Servicio de AWS que ayuda al administrador a controlar de forma segura el acceso a AWS los recursos. IAMlos administradores controlan quién puede autenticarse (iniciar sesión) y quién puede autorizarse (tener permisos) para usar HealthLake los recursos. IAMes una Servicio de AWS que puede utilizar sin coste adicional.

Temas

- Público

- Autenticación con identidades
- Administración de acceso mediante políticas
- ¿Cómo AWS HealthLake funciona con IAM
- Ejemplos de políticas basadas en la identidad para AWS HealthLake
- AWS políticas gestionadas para AWS HealthLake
- Solución de problemas de AWS HealthLake identidad y acceso

Público

La forma de usar AWS Identity and Access Management (IAM) varía según el trabajo en el que se realice HealthLake.

Usuario del servicio: si utiliza el HealthLake servicio para realizar su trabajo, el administrador le proporcionará las credenciales y los permisos que necesita. A medida que vaya utilizando más HealthLake funciones para realizar su trabajo, es posible que necesite permisos adicionales. Entender cómo se administra el acceso puede ayudarle a solicitar los permisos correctos al administrador. Si no puede acceder a una característica en HealthLake, consulte Solución de problemas de AWS HealthLake identidad y acceso.

Administrador de servicios: si estás a cargo de HealthLake los recursos de tu empresa, probablemente tengas acceso total a ellos HealthLake. Su trabajo consiste en determinar a qué HealthLake funciones y recursos deben acceder los usuarios del servicio. A continuación, debe enviar solicitudes a su administrador de IAM para cambiar los permisos de los usuarios de su servicio. Revise la información de esta página para conocer los conceptos básicos de IAM. Para obtener más información sobre cómo puede IAM utilizarlos su empresa HealthLake, consulte¿Cómo AWS HealthLake funciona con IAM.

IAMadministrador: si es IAM administrador, puede que desee obtener más información sobre cómo puede redactar políticas para administrar el acceso a ellas HealthLake. Para ver ejemplos de políticas HealthLake basadas en la identidad que puede utilizarIAM, consulte. Ejemplos de políticas basadas en la identidad para AWS HealthLake

Autenticación con identidades

La autenticación es la forma de iniciar sesión AWS con sus credenciales de identidad. Debe estar autenticado (con quien haya iniciado sesión AWS) como IAM usuario o asumiendo un IAM rol. Usuario raíz de la cuenta de AWS

Puede iniciar sesión AWS como una identidad federada mediante las credenciales proporcionadas a través de una fuente de identidad. AWS IAM Identity Center Los usuarios (IAMIdentity Center), la autenticación de inicio de sesión único de su empresa y sus credenciales de Google o Facebook son ejemplos de identidades federadas. Al iniciar sesión como identidad federada, su administrador habrá configurado previamente la federación de identidades mediante roles de IAM. Cuando accedes AWS mediante la federación, estás asumiendo un rol de forma indirecta.

Según el tipo de usuario que sea, puede iniciar sesión en el portal AWS Management Console o en el de AWS acceso. Para obtener más información sobre cómo iniciar sesión AWS, consulte Cómo iniciar sesión Cuenta de AWS en su Guía del AWS Sign-In usuario.

Si accede AWS mediante programación, AWS incluye un kit de desarrollo de software (SDK) y una interfaz de línea de comandos (CLI) para firmar criptográficamente sus solicitudes con sus credenciales. Si no utilizas AWS herramientas, debes firmar las solicitudes tú mismo. Para obtener más información sobre cómo usar el método recomendado para firmar las solicitudes usted mismo, consulte la versión 4 de la AWS firma para ver API las solicitudes en la Guía del IAM usuario.

Independientemente del método de autenticación que use, es posible que deba proporcionar información de seguridad adicional. Por ejemplo, le AWS recomienda que utilice la autenticación multifactorial (MFA) para aumentar la seguridad de su cuenta. Para obtener más información, consulte Autenticación multifactorial en la Guía del AWS IAM Identity Center usuario y Autenticación AWS multifactorial IAM en la Guía del IAM usuario.

Cuenta de AWS usuario root

Al crear una Cuenta de AWS, comienza con una identidad de inicio de sesión que tiene acceso completo a todos Servicios de AWS los recursos de la cuenta. Esta identidad se denomina usuario Cuenta de AWS raíz y se accede a ella iniciando sesión con la dirección de correo electrónico y la contraseña que utilizaste para crear la cuenta. Recomendamos encarecidamente que no utiliza el usuario raíz para sus tareas diarias. Proteja las credenciales del usuario raíz y utilícelas solo para las tareas que solo el usuario raíz pueda realizar. Para obtener la lista completa de las tareas que requieren que inicie sesión como usuario raíz, consulte Tareas que requieren credenciales de usuario raíz en la Guía del usuario de IAM.

Identidad federada

Como práctica recomendada, exija a los usuarios humanos, incluidos los que requieren acceso de administrador, que utilicen la federación con un proveedor de identidades para acceder Servicios de AWS mediante credenciales temporales.

Una identidad federada es un usuario del directorio de usuarios de su empresa, un proveedor de identidades web AWS Directory Service, el directorio del Centro de Identidad o cualquier usuario al que acceda Servicios de AWS mediante las credenciales proporcionadas a través de una fuente de identidad. Cuando las identidades federadas acceden Cuentas de AWS, asumen funciones y las funciones proporcionan credenciales temporales.

Para una administración de acceso centralizada, le recomendamos que utiliza AWS IAM Identity Center. Puede crear usuarios y grupos en IAM Identity Center, o puede conectarse y sincronizarse con un conjunto de usuarios y grupos de su propia fuente de identidad para usarlos en todas sus aplicaciones Cuentas de AWS . Para obtener información sobre IAM Identity Center, consulte ¿Qué es IAM Identity Center? en la Guía AWS IAM Identity Center del usuario.

Usuarios y grupos de IAM

Un IAMusuario es una identidad propia Cuenta de AWS que tiene permisos específicos para una sola persona o aplicación. Siempre que sea posible, recomendamos utilizar credenciales temporales en lugar de crear IAM usuarios con credenciales de larga duración, como contraseñas y claves de acceso. Sin embargo, si tiene casos de uso específicos que requieren credenciales a largo plazo con IAM los usuarios, le recomendamos que rote las claves de acceso. Para obtener más información, consulte Rotar las claves de acceso periódicamente para casos de uso que requieran credenciales de larga duración en la Guía del usuario de IAM.

Un grupo de IAM es una identidad que especifica un conjunto de usuarios de IAM. No puede iniciar sesión como grupo. Puede usar los grupos para especificar permisos para varios usuarios a la vez. Los grupos facilitan la administración de los permisos para grandes conjuntos de usuarios. Por ejemplo, puede asignar un nombre a un grupo IAMAdminsy concederle permisos para administrar IAM los recursos.

Los usuarios son diferentes de los roles. Un usuario se asocia exclusivamente a una persona o aplicación, pero la intención es que cualquier usuario pueda asumir un rol que necesite. Los usuarios tienen credenciales de larga duración permanentes; no obstante, los roles proporcionan credenciales temporales. Para obtener más información, consulte Casos de uso para IAM usuarios en la Guía del IAM usuario.

Roles de IAM

Un IAMrol es una identidad dentro de ti Cuenta de AWS que tiene permisos específicos. Es similar a un usuario de IAM, pero no está asociado a una determinada persona. Para asumir temporalmente un IAM rol en el AWS Management Console, puede cambiar de un IAM rol de usuario a uno

(consola). Puede asumir un rol llamando a una AWS API operación AWS CLI o o utilizando una operación personalizadaURL. Para obtener más información sobre los métodos de uso de roles, consulte Métodos para asumir un rol en la Guía del IAM usuario.

Los roles de IAM con credenciales temporales son útiles en las siguientes situaciones:

- Acceso de usuario federado: para asignar permisos a una identidad federada, puede crear un rol y definir los permisos para este. Cuando se autentica una identidad federada, se asocia la identidad al rol y se le conceden los permisos define el rol. Para obtener información sobre las funciones de la federación, consulte Crear una función para un proveedor de identidades externo (federación) en la Guía del IAM usuario. Si usa IAM Identity Center, configura un conjunto de permisos. Para controlar a qué pueden acceder sus identidades después de autenticarse, IAM Identity Center correlaciona el conjunto de permisos con un rol en. IAM Para obtener información acerca de los conjuntos de permisos, consulta Conjuntos de permisos en la Guía del usuario de AWS IAM Identity Center .
- Permisos IAM de usuario temporales: un IAM usuario o rol puede asumir un IAM rol para asumir temporalmente diferentes permisos para una tarea específica.
- Acceso entre cuentas: puede utilizar un rol de IAM para permitir que alguien (una entidad principal de confianza) de otra cuenta obtenga acceso a los recursos de su cuenta. Los roles son la forma principal de conceder acceso entre cuentas. Sin embargo, con algunos Servicios de AWS, puedes adjuntar una política directamente a un recurso (en lugar de usar un rol como proxy). Para conocer la diferencia entre las funciones y las políticas basadas en recursos para el acceso multicuenta, consulta el tema sobre el acceso a los recursos entre cuentas IAM en la Guía del IAM usuario.
- Acceso entre servicios: algunos Servicios de AWS utilizan funciones en otros. Servicios de AWS Por ejemplo, cuando realizas una llamada en un servicio, es habitual que ese servicio ejecute aplicaciones en Amazon EC2 o almacene objetos en Amazon S3. Es posible que un servicio haga esto usando los permisos de la entidad principal, usando un rol de servicio o usando un rol vinculado al servicio.
  - Sesiones de acceso directo (FAS): cuando utilizas un IAM usuario o un rol para realizar acciones en AWS ellas, se te considera director. Al utilizar algunos servicios, es posible que realice una acción que, a continuación, inicie otra acción en un servicio diferente. FASutiliza los permisos de la persona principal que llama a an Servicio de AWS, junto con los que solicitan, Servicio de AWS para realizar solicitudes a los servicios descendentes. FASlas solicitudes solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros recursos Servicios de AWS o para completarse. En este caso, debe tener permisos para realizar ambas acciones.

Para obtener detalles sobre la política a la hora de realizar FAS solicitudes, consulte Reenviar sesiones de acceso.

- Función de servicio: una función de servicio es una IAMfunción que un servicio asume para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener más información, consulte Crear un rol para delegar permisos a un Servicio de AWS en la Guía del IAM usuario.
- Función vinculada a un servicio: una función vinculada a un servicio es un tipo de función de servicio que está vinculada a un. Servicio de AWS El servicio puede asumir el rol para realizar una acción en su nombre. Los roles vinculados al servicio aparecen en usted Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.

- Aplicaciones que se ejecutan en Amazon EC2: puedes usar un IAM rol para administrar las credenciales temporales de las aplicaciones que se ejecutan en una EC2 instancia y que realizan AWS CLI o AWS API solicitan. Es preferible hacerlo de este modo a almacenar claves de acceso dentro de la instancia EC2. Para asignar un AWS rol a una EC2 instancia y ponerlo a disposición de todas sus aplicaciones, debe crear un perfil de instancia adjunto a la instancia. Un perfil de instancia contiene el rol y permite a los programas que se ejecutan en la instancia EC2 obtener credenciales temporales. Para obtener más información, consulta Usar un IAM rol para conceder permisos a las aplicaciones que se ejecutan en EC2 instancias de Amazon en la Guía del IAM usuario.

Administración de acceso mediante políticas

El acceso se controla AWS creando políticas y adjuntándolas a AWS identidades o recursos. Una política es un objeto AWS que, cuando se asocia a una identidad o un recurso, define sus permisos. AWS evalúa estas políticas cuando un director (usuario, usuario raíz o sesión de rol) realiza una solicitud. Los permisos en las políticas determinan si la solicitud se permite o se deniega. La mayoría de las políticas se almacenan AWS como JSON documentos. Para obtener más información sobre la estructura y el contenido de los documentos de JSON políticas, consulte Descripción general de JSON las políticas en la Guía del IAM usuario.

Los administradores pueden usar AWS JSON las políticas para especificar quién tiene acceso a qué. Es decir, qué entidad principal puedes realizar acciones en qué recursos y en qué condiciones.

De forma predeterminada, los usuarios y los roles no tienen permisos. Un administrador de IAM puede crear políticas de IAM para conceder permisos a los usuarios para realizar acciones en los

recursos que necesitan. A continuación, el administrador puede agregar las políticas de IAM a los roles y los usuarios pueden asumirlos.

Las políticas de IAM definen permisos para una acción, independientemente del método que se utilice para realizar la operación. Por ejemplo, suponga que dispone de una política que permite la acción `iam:GetRole`. Un usuario con esa política puede obtener información sobre el rol de AWS Management Console AWS CLI, el o el AWS API.

Políticas basadas en identidad

Las políticas basadas en la identidad son documentos de política de JSON permisos que se pueden adjuntar a una identidad, como un IAM usuario, un grupo de usuarios o un rol. Estas políticas controlan qué acciones puedes realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener información sobre cómo crear una política basada en la identidad, consulte Definir IAM permisos personalizados con políticas administradas por el cliente en la Guía del usuario. IAM

Las políticas basadas en identidades pueden clasificarse además como políticas insertadas o políticas gestionadas. Las políticas insertadas se integran directamente en un único usuario, grupo o rol. Las políticas administradas son políticas independientes que puede adjuntar a varios usuarios, grupos y funciones de su empresa. Cuenta de AWS Las políticas administradas incluyen políticas AWS administradas y políticas administradas por el cliente. Para saber cómo elegir entre una política gestionada o una política integrada, consulte Elegir entre políticas gestionadas y políticas integradas en la Guía del IAM usuario.

Políticas basadas en recursos

Las políticas basadas en recursos son documentos de JSON política que se adjuntan a un recurso. Ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM y las políticas de buckets de Amazon S3. En los servicios que admiten políticas basadas en recursos, los gestionadores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puedes realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe especificar una entidad principal en una política en función de recursos. Los principales pueden incluir cuentas, usuarios, roles, usuarios federados o. Servicios de AWS

Las políticas basadas en recursos son políticas insertadas que se encuentran en ese servicio. No puede usar políticas AWS administradas desde una política IAM basada en recursos.

Listas de control de acceso () ACLs

Las listas de control de acceso (ACLs) controlan qué responsables (miembros de la cuenta, usuarios o roles) tienen permisos para acceder a un recurso. ACLsson similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de JSON políticas.

Amazon S3 AWS WAF y Amazon VPC son ejemplos de servicios compatiblesACLs. Para obtener más informaciónACLs, consulte la descripción general de la lista de control de acceso (ACL) en la Guía para desarrolladores de Amazon Simple Storage Service.

Otros tipos de políticas

AWS admite tipos de políticas adicionales y menos comunes. Estos tipos de políticas puedes establecer el máximo de permisos que los tipos de políticas más frecuentes le conceden.

- Límites de permisos: un límite de permisos es una función avanzada en la que se establecen los permisos máximos que una política basada en la identidad puede conceder a una IAM entidad (IAMusuario o rol). Puedes establecer un límite de permisos para una entidad. Los permisos resultantes son la intersección de las políticas basadas en la identidad de la entidad y los límites de permisos. Las políticas basadas en recursos que especifiquen el usuario o rol en el campo `Principal` no estarán restringidas por el límite de permisos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para obtener más información sobre los límites de los permisos, consulte Límites de permisos para las entidades de IAM en la Guía del usuario de IAM.
- Políticas de control de servicios (SCPs): SCPs son JSON políticas que especifican los permisos máximos para una organización o unidad organizativa (OU). AWS Organizations AWS Organizations es un servicio para agrupar y administrar de forma centralizada varios de los Cuentas de AWS que son propiedad de su empresa. Si habilitas todas las funciones de una organización, puedes aplicar políticas de control de servicios (SCPs) a una o a todas tus cuentas. SCPLimita los permisos de las entidades en las cuentas de los miembros, incluidas las de cada una Usuario raíz de la cuenta de AWS. Para obtener más información sobre OrganizationsSCPs, consulte las políticas de control de servicios en la Guía del AWS Organizations usuario.
- Políticas de control de recursos (RCPs): RCPs son JSON políticas que puedes usar para establecer los permisos máximos disponibles para los recursos de tus cuentas sin actualizar las IAM políticas asociadas a cada recurso que poseas. Esto RCP limita los permisos de los recursos en las cuentas de los miembros y puede afectar a los permisos efectivos de las identidades Usuario raíz de la cuenta de AWS, incluidos los permisos, independientemente de si pertenecen a su organización. Para obtener más información sobre Organizations e RCPs incluir una lista de

Servicios de AWS ese apoyoRCPs, consulte Políticas de control de recursos (RCPs) en la Guía del AWS Organizations usuario.

- Políticas de sesión: las políticas de sesión son políticas avanzadas que se pasan como parámetro cuando se crea una sesión temporal mediante programación para un rol o un usuario federado. Los permisos de la sesión resultantes son la intersección de las políticas basadas en identidades del rol y las políticas de la sesión. Los permisos también puedes proceder de una política en función de recursos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para obtener más información, consulte Políticas de sesión en la Guía del usuario de IAM.

Varios tipos de políticas

Cuando se aplican varios tipos de políticas a una solicitud, los permisos resultantes son más complicados de entender. Para saber cómo se AWS determina si se debe permitir una solicitud cuando se trata de varios tipos de políticas, consulte la lógica de evaluación de políticas en la Guía del IAM usuario.

¿Cómo AWS HealthLake funciona con IAM

Antes de administrar el IAM acceso a HealthLake, infórmese sobre IAM las funciones disponibles para su uso HealthLake.

IAMfunciones que puedes usar con AWS HealthLake

Característica de IAM	HealthLake apoyo
Políticas basadas en identidades	Sí
Políticas basadas en recursos	No
Acciones de políticas	Sí
Recursos de políticas	Sí
Claves de condición de política	Sí
ACLs	No
ABAC(etiquetas en las políticas)	Sí
Credenciales temporales	Sí

Característica de IAM	HealthLake apoyo
Permisos de entidades principales	Sí
Roles de servicio	Sí
Roles vinculados al servicio	No

Para obtener una visión general de cómo HealthLake funcionan otros AWS servicios con la mayoría de las IAM funciones, consulte AWS los servicios con los que funcionan IAM en la Guía del IAM usuario.

Políticas basadas en la identidad para AWS HealthLake

Compatibilidad con las políticas basadas en identidad: sí

Las políticas basadas en la identidad son documentos de política de JSON permisos que se pueden adjuntar a una identidad, como un IAM usuario, un grupo de usuarios o un rol. Estas políticas controlan qué acciones puedes realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener información sobre cómo crear una política basada en la identidad, consulte Definir IAM permisos personalizados con políticas administradas por el cliente en la Guía del usuario. IAM

Con las políticas basadas en identidades de IAM, puede especificar las acciones permitidas o denegadas, así como los recursos y las condiciones en las que se permiten o deniegan las acciones. No es posible especificar la entidad principal en una política basada en identidad porque se aplica al usuario o rol al que está adjunto. Para obtener más información sobre todos los elementos que puede usar en una JSON política, consulte la referencia sobre los elementos de la IAM JSON política en la Guía del IAMusuario.

Ejemplos de políticas basadas en la identidad para AWS HealthLake

Para ver ejemplos de políticas HealthLake basadas en la identidad, consulte. Ejemplos de políticas basadas en la identidad para AWS HealthLake

Políticas basadas en recursos incluidas AWS HealthLake

Admite políticas basadas en recursos: no

Las políticas basadas en recursos son documentos de JSON política que se adjuntan a un recurso. Ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM y las políticas de buckets de Amazon S3. En los servicios que admiten políticas basadas en recursos, los gestionadores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puedes realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe especificar una entidad principal en una política en función de recursos. Los principales pueden incluir cuentas, usuarios, roles, usuarios federados o. Servicios de AWS

Para hacer posible el acceso entre cuentas, puede especificar toda una cuenta o entidades de IAM de otra cuenta como entidad principal de la política basada en recursos. Añadir a una política en función de recursos una entidad principal entre cuentas es solo una parte del establecimiento de una relación de confianza. Cuando el principal y el recurso son diferentes Cuentas de AWS, el IAM administrador de la cuenta de confianza también debe conceder a la entidad principal (usuario o rol) el permiso para acceder al recurso. Para conceder el permiso, adjunte la entidad a una política basada en identidad. Sin embargo, si la política en función de recursos concede el acceso a una entidad principal de la misma cuenta, no es necesaria una política basada en identidad adicional. Para obtener más información, consulte Acceso a recursos entre cuentas IAM en la Guía del IAM usuario.

Acciones políticas para AWS HealthLake

Compatibilidad con las acciones de política: sí

Los administradores pueden usar AWS JSON políticas para especificar quién tiene acceso a qué. Es decir, qué entidad principal puedes realizar acciones en qué recursos y en qué condiciones.

El `Action` elemento de una JSON política describe las acciones que puede utilizar para permitir o denegar el acceso en una política. Las acciones de política suelen tener el mismo nombre que la AWS API operación asociada. Hay algunas excepciones, como las acciones que solo permiten permisos y que no tienen una operación coincidente. API También hay algunas operaciones que requieren varias acciones en una política. Estas acciones adicionales se denominan acciones dependientes.

Incluya acciones en una política para conceder permisos y así llevar a cabo la operación asociada.

Para ver una lista de HealthLake acciones, consulte las acciones definidas AWS HealthLake en la Referencia de autorización de servicios.

Las acciones políticas HealthLake utilizan el siguiente prefijo antes de la acción:

```
healthlake
```

Para especificar varias acciones en una única sentencia, separe cada acción con una coma.

```
"Action": [
      "healthlake:action1",
      "healthlake:action2"
         ]
```

Para ver ejemplos de políticas HealthLake basadas en la identidad, consulte. Ejemplos de políticas basadas en la identidad para AWS HealthLake

Recursos de políticas para AWS HealthLake

Compatibilidad con los recursos de políticas: sí

Los administradores pueden usar AWS JSON políticas para especificar quién tiene acceso a qué. Es decir, qué entidad principal puedes realizar acciones en qué recursos y en qué condiciones.

El elemento `Resource` JSON de política especifica el objeto o los objetos a los que se aplica la acción. Las instrucciones deben contener un elemento `Resource` o `NotResource`. Como práctica recomendada, especifique un recurso mediante su nombre de recurso de Amazon (ARN). Puedes hacerlo para acciones que admitan un tipo de recurso específico, conocido como permisos de nivel de recurso.

Para las acciones que no admiten permisos de nivel de recurso, como las operaciones de descripción, utiliza un carácter comodín (*) para indicar que la instrucción se aplica a todos los recursos.

```
"Resource": "*"
```

Para ver una lista de los tipos de HealthLake recursos y sus respectivos tiposARNs, consulte los recursos definidos AWS HealthLake en la Referencia de autorización de servicio. Para obtener información sobre las acciones con las que puede especificar cada recurso, consulte Acciones definidas por AWS HealthLake. ARN

Para ver ejemplos de políticas HealthLake basadas en la identidad, consulte. Ejemplos de políticas basadas en la identidad para AWS HealthLake

Claves de condición de la política para AWS HealthLake

Compatibilidad con claves de condición de políticas específicas del servicio: sí

Los administradores pueden usar AWS JSON políticas para especificar quién tiene acceso a qué. Es decir, qué entidad principal puedes realizar acciones en qué recursos y en qué condiciones.

El elemento `Condition` (o bloque de `Condition`) permite especificar condiciones en las que entra en vigor una instrucción. El elemento `Condition` es opcional. Puedes crear expresiones condicionales que utilizan operadores de condición, tales como igual o menor que, para que la condición de la política coincida con los valores de la solicitud.

Si especifica varios elementos de `Condition` en una instrucción o varias claves en un único elemento de `Condition`, AWS las evalúa mediante una operación `AND` lógica. Si especifica varios valores para una única clave de condición, AWS evalúa la condición mediante una `OR` operación lógica. Se deben cumplir todas las condiciones antes de que se concedan los permisos de la instrucción.

También puedes utilizar variables de marcador de posición al especificar condiciones. Por ejemplo, puede conceder un permiso de usuario de IAM para acceder a un recurso solo si está etiquetado con su nombre de usuario de IAM. Para obtener más información, consulte Elementos de la política de IAM: variables y etiquetas en la Guía del usuario de IAM.

AWS admite claves de condición globales y claves de condición específicas del servicio. Para ver todas las claves de condición AWS globales, consulte las claves de contexto de condición AWS globales en la Guía del IAMusuario.

Para ver una lista de claves de HealthLake condición, consulte las claves de condición AWS HealthLake en la Referencia de autorización de servicio. Para obtener información sobre las acciones y los recursos con los que puede utilizar una clave de condición, consulte Acciones definidas por AWS HealthLake.

Para ver ejemplos de políticas HealthLake basadas en la identidad, consulte. Ejemplos de políticas basadas en la identidad para AWS HealthLake

Listas de control de acceso () ACLs en AWS HealthLake

SoportaACLs: No

Las listas de control de acceso (ACLs) controlan qué directores (miembros de la cuenta, usuarios o roles) tienen permisos para acceder a un recurso. ACLsson similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de JSON políticas.

Control de acceso basado en atributos () con ABAC AWS HealthLake

Soportes ABAC (etiquetas en las políticas): Sí

El control de acceso basado en atributos (ABAC) es una estrategia de autorización que define los permisos en función de los atributos. En AWS, estos atributos se denominan etiquetas. Puede adjuntar etiquetas a IAM entidades (usuarios o roles) y a muchos AWS recursos. Etiquetar entidades y recursos es el primer paso deABAC. Luego, diseñe ABAC políticas para permitir las operaciones cuando la etiqueta del principal coincida con la etiqueta del recurso al que está intentando acceder.

ABACes útil en entornos de rápido crecimiento y ayuda en situaciones en las que la administración de políticas se vuelve engorrosa.

Para controlar el acceso en función de etiquetas, debe proporcionar información de las etiquetas en el elemento de condición de una política utilizando las claves de condición `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`.

Si un servicio admite las tres claves de condición para cada tipo de recurso, el valor es Sí para el servicio. Si un servicio admite las tres claves de condición solo para algunos tipos de recursos, el valor es Parcial.

Para obtener más informaciónABAC, consulte Definir permisos con ABAC autorización en la Guía del IAM usuario. Para ver un tutorial con los pasos de configuraciónABAC, consulte Usar el control de acceso basado en atributos (ABAC) en la Guía del IAMusuario.

Usar credenciales temporales con AWS HealthLake

Compatibilidad con credenciales temporales: sí

Algunas Servicios de AWS no funcionan cuando inicias sesión con credenciales temporales. Para obtener información adicional, incluida la información sobre cuáles Servicios de AWS funcionan con credenciales temporales, consulta Servicios de AWS la guía del IAM usuario sobre cómo trabajar con IAM ellas.

Está utilizando credenciales temporales si inicia sesión en ellas AWS Management Console mediante cualquier método excepto un nombre de usuario y una contraseña. Por ejemplo, cuando accedes AWS mediante el enlace de inicio de sesión único (SSO) de tu empresa, ese proceso crea automáticamente credenciales temporales. También crea credenciales temporales de forma automática cuando inicia sesión en la consola como usuario y luego cambia de rol. Para obtener más información sobre el cambio de rol, consulte Cambiar de un rol de usuario a un IAM rol (consola) en la Guía del IAMusuario.

Puede crear credenciales temporales manualmente con la tecla AWS CLI o AWS API. A continuación, puede utilizar esas credenciales temporales para acceder AWS. AWS recomienda generar credenciales temporales de forma dinámica en lugar de utilizar claves de acceso a largo plazo. Para obtener más información, consulte Credenciales de seguridad temporales en IAM.

Permisos principales entre servicios para AWS HealthLake

Admite sesiones de acceso directo (FAS): Sí

Cuando utilizas un IAM usuario o un rol para realizar acciones en AWSél, se te considera director. Al utilizar algunos servicios, es posible que realice una acción que, a continuación, inicie otra acción en un servicio diferente. FASutiliza los permisos de la persona principal que llama a an Servicio de AWS, junto con los que solicitan, Servicio de AWS para realizar solicitudes a los servicios descendentes. FASlas solicitudes solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros recursos Servicios de AWS o para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener detalles sobre la política a la hora de realizar FAS solicitudes, consulte Reenviar sesiones de acceso.

Roles de servicio para AWS HealthLake

Compatibilidad con roles de servicio: sí

Un rol de servicio es un rol de IAM que asume un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener más información, consulte Crear un rol para delegar permisos a un Servicio de AWS en la Guía del IAM usuario.

Para obtener información sobre las funciones de servicio y la política interna necesaria para el acceso completo a ellas AWS HealthLake, consulteConfigurar los permisos para empezar a usar AWS HealthLake.

> **Warning**
>
> Cambiar los permisos de un rol de servicio podría afectar a la HealthLake funcionalidad. Edite las funciones de servicio solo cuando se HealthLake proporcionen instrucciones para hacerlo.

Funciones vinculadas al servicio para AWS HealthLake

Compatibilidad con roles vinculados al servicio: no

Un rol vinculado a un servicio es un tipo de rol de servicio que está vinculado a un. Servicio de AWS El servicio puede asumir el rol para realizar una acción en su nombre. Los roles vinculados al servicio aparecen en usted Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.

Para obtener más información acerca de cómo crear o administrar roles vinculados a servicios, consulte Servicios de AWS que funcionan con IAM. Busque un servicio en la tabla que incluya `Yes` en la columna Rol vinculado a un servicio. Seleccione el vínculo Sí para ver la documentación acerca del rol vinculado a servicios para ese servicio.

Ejemplos de políticas basadas en la identidad para AWS HealthLake

De forma predeterminada, los usuarios y roles no tienen permiso para crear, ver ni modificar recursos de HealthLake. Tampoco pueden realizar tareas con AWS Management Console, AWS Command Line Interface (AWS CLI) o. AWS API Un administrador de IAM puede crear políticas de IAM para conceder permisos a los usuarios para realizar acciones en los recursos que necesitan. A continuación, el administrador puede agregar las políticas de IAM a los roles y los usuarios pueden asumirlos.

Para obtener información sobre cómo crear una política IAM basada en la identidad mediante estos documentos de JSON política de ejemplo, consulte Crear IAM políticas (consola) en la Guía del IAMusuario.

Para obtener más información sobre las acciones y los tipos de recursos definidos HealthLake, incluido el formato de cada uno de los tipos de recursos, consulte las claves de condición, recursos y acciones de la Referencia AWS HealthLake de autorización de servicios. ARNs

Temas

- Prácticas recomendadas sobre las políticas
- Mediante la consola de AWS HealthLake
- Acceder a un almacén de AWS HealthLake datos en Amazon Athena
- Cómo permitir a los usuarios que vean sus propios permisos

Prácticas recomendadas sobre las políticas

Las políticas basadas en la identidad determinan si alguien puede crear HealthLake recursos de tu cuenta, acceder a ellos o eliminarlos. Estas acciones pueden generar costos adicionales para su Cuenta de AWS. Siga estas directrices y recomendaciones al crear o editar políticas basadas en identidades:

- Comience con las políticas AWS administradas y avance hacia los permisos con privilegios mínimos: para empezar a conceder permisos a sus usuarios y cargas de trabajo, utilice las políticas AWS administradas que otorgan permisos para muchos casos de uso comunes. Están disponibles en su. Cuenta de AWS Le recomendamos que reduzca aún más los permisos definiendo políticas administradas por el AWS cliente que sean específicas para sus casos de uso. Con el fin de obtener más información, consulte las políticas administradas por AWS o las políticas administradas por AWS para funciones de trabajo en la Guía de usuario de IAM.
- Aplique permisos de privilegio mínimo: cuando establezca permisos con políticas de IAM, conceda solo los permisos necesarios para realizar una tarea. Para ello, debe definir las acciones que se puedes llevar a cabo en determinados recursos en condiciones específicas, también conocidos como permisos de privilegios mínimos. Con el fin de obtener más información sobre el uso de IAM para aplicar permisos, consulte Políticas y permisos en IAM en la Guía de usuario de IAM.
- Use condiciones en las políticas de IAM para restringir aún más el acceso: puede agregar una condición a sus políticas para limitar el acceso a las acciones y los recursos. Por ejemplo, puede escribir una condición de política para especificar que todas las solicitudes deben enviarse medianteSSL. También puedes usar condiciones para conceder el acceso a las acciones del servicio si se utilizan a través de una acción específica Servicio de AWS, como AWS CloudFormation. Para obtener más información, consulte los elementos IAM JSON de la política: Condición en la Guía del IAM usuario.
- Utilice IAM Access Analyzer para validar sus IAM políticas y garantizar permisos seguros y funcionales: IAM Access Analyzer valida las políticas nuevas y existentes para que se ajusten al lenguaje de las políticas (JSON) y IAM a las IAM mejores prácticas. IAM Access Analyzer proporciona más de 100 comprobaciones de políticas y recomendaciones prácticas para ayudarlo

a crear políticas seguras y funcionales. Para obtener más información, consulte Validar políticas con IAM Access Analyzer en la Guía del IAM usuario.

- Requerir autenticación multifactorial (MFA): si se encuentra en una situación en la que se requieren IAM usuarios o un usuario raíz Cuenta de AWS, actívela MFA para aumentar la seguridad. Para solicitarlo MFA cuando se convoque a API las operaciones, añada MFA condiciones a sus políticas. Para obtener más información, consulte APIAcceso seguro con MFA en la Guía del IAM usuario.

Para obtener más información sobre las prácticas recomendadas de IAM, consulte las Prácticas recomendadas de seguridad en IAM en la Guía de usuario de IAM.

Mediante la consola de AWS HealthLake

Para acceder a la AWS HealthLake consola, debe tener un conjunto mínimo de permisos. Estos permisos deben permitirle enumerar y ver detalles sobre los HealthLake recursos de su cuenta Cuenta de AWS. Si crea una política basada en identidades que sea más restrictiva que el mínimo de permisos necesarios, la consola no funcionará del modo esperado para las entidades (usuarios o roles) que tengan esa política.

No es necesario que concedas permisos mínimos de consola a los usuarios que realicen llamadas únicamente al AWS CLI o al AWS API. En su lugar, permita el acceso únicamente a las acciones que coincidan con la API operación que están intentando realizar.

Para tener acceso completo a HealthLake, asocie las siguientes políticas a un IAM usuario o rol: `AmazonHealthLakeFullAccess` y`AWSLakeFormationDataAdmin`. También debe adjuntar la política HealthLake en línea, que es una función de servicio. Un rol de servicio es un rol de IAM que asume un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener más información, consulte Crear un rol para delegar permisos a un Servicio de AWS en la Guía del IAM usuario. Para obtener información sobre la política en línea que crea el rol de servicio requerido, consulteConfigurar los permisos para empezar a usar AWS HealthLake. También debe usar la AWS Lake Formation consola o asignar CLI a su HealthLake administrador como administrador de AWS Lake Formation Data Lake. Para obtener más información, consulte Configurar los permisos para empezar a usar AWS HealthLake.

Acceder a un almacén de AWS HealthLake datos en Amazon Athena

Si desea proporcionar a los usuarios y roles acceso a los almacenes de HealthLake datos Amazon Athena, adjunte las siguientes IAM políticas al rol o usuario: `AmazonAthenaFullAccess` y`AmazonS3FullAccess`. `Select`y también se requieren `Describe` permisos en las tablas administradas por AWS Lake Formation. AWS Lake Formation Los permisos de tabla los concede un AWS Lake Formation administrador en la AWS Lake Formation consola o medianteCLI. Para obtener más información, consulte Configurar los permisos para empezar a usar AWS HealthLake

Cómo permitir a los usuarios que vean sus propios permisos

En este ejemplo, se muestra cómo podría crear una política que permita a los usuarios de IAM ver las políticas administradas e insertadas que se asocian a la identidad de sus usuarios. Esta política incluye permisos para completar esta acción en la consola o mediante programación mediante la AWS CLI tecla o. 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 políticas gestionadas para AWS HealthLake

Una política AWS administrada es una política independiente creada y administrada por AWS. AWS Las políticas administradas están diseñadas para proporcionar permisos para muchos casos de uso comunes, de modo que pueda empezar a asignar permisos a usuarios, grupos y funciones.

Ten en cuenta que es posible que las políticas AWS administradas no otorguen permisos con privilegios mínimos para tus casos de uso específicos, ya que están disponibles para que los usen todos los AWS clientes. Se recomienda definir políticas administradas por el cliente específicas para sus casos de uso a fin de reducir aún más los permisos.

No puedes cambiar los permisos definidos en AWS las políticas administradas. Si AWS actualiza los permisos definidos en una política AWS administrada, la actualización afecta a todas las identidades principales (usuarios, grupos y roles) a las que está asociada la política. AWS es más probable que actualice una política AWS administrada cuando Servicio de AWS se lance una nueva o cuando haya nuevas API operaciones disponibles para los servicios existentes.

Para obtener más información, consulte Políticas administradas de AWS en la Guía del usuario de IAM.

AWS política gestionada: AmazonHealthLakeFullAccess

La `AmazonHealthLakeFullAccess` política proporciona acceso completo a HealthLake. Con esta política asociada a su usuario o función, los usuarios pueden utilizarla HealthLake para acceder a los datos, consultarlos, importarlos y exportarlos HealthLake. Para realizar muchas de las acciones habituales HealthLake, debe añadir políticas adicionales al usuario o al rol. Para obtener más información, consulte HealthLake operaciones Configurar los permisos para empezar a usar AWS HealthLake y permisos.

Puede adjuntar la política `AmazonHealthLakeFullAccess` a las identidades de IAM.

Esta política concede *`administrative and contributor`* permisos que permiten a los usuarios y roles consultar, buscar, importar y exportar HealthLake, y también permite HealthLake realizar acciones en nombre de los usuarios y roles que tienen estos permisos.

Detalles de los permisos

Esta política incluye la siguiente declaración.

```
{
 "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 política gestionada: AmazonHealthLakeReadOnlyAccess

`AmazonHealthLakeReadOnlyAccess`la política otorga acceso y permisos de solo lectura a otros servicios HealthLake y a los recursos relacionados en otros AWS servicios. Aplique esta política a

los usuarios a los que desee conceder la posibilidad de consultar y ver HealthLake los almacenes de datos, pero no la posibilidad de crearlos o realizar cambios en ellos.

Puede adjuntar la política `AmazonHealthLakeReadOnlyAccess` a las identidades de IAM.

Esta política otorga *`read-only`* permisos que permiten a los usuarios y a los roles realizar consultas HealthLake.

Detalles de los permisos

Esta política incluye la siguiente declaración.

```
{
    "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 operaciones y permisos

En la siguiente tabla se enumeran las operaciones típicas HealthLake y los permisos necesarios para realizarlas.

HealthLake operaciones	Permisos necesarios
Cree un almacén de datos en HealthLake	`AmazonHealthLakeFullAccess` `AmazonLakeFormationDataAdmin` , la política en línea y los permisos de AWS Lake Formation administrador gestionados por AWS Lake Formation
Eliminar un almacén de datos en HealthLake	`AmazonHealthLakeFullAccess` `AmazonLakeFormationDataAdmin` , la política en línea y los permisos de AWS Lake Formation administrador gestionados por AWS Lake Formation
Enumere, busque o consulte un almacén de datos en HealthLake	`AmazonHealthLakeReadOnlyAccess`
Consulte un banco de datos mediante Amazon Athena	`AmazonAthenaFullAccess` `AmazonS3FullAccess` , AWS Lake Formation `Select` y `Describe` permisos en las tablas administradas por AWS Lake Formation
Importar datos de HealthLake	Consulte Configuración de permisos para trabajos de importación.
Exportar datos desde HealthLake	Consulte Exportación de archivos del almacén de datos (AWS SDKs).

HealthLake actualizaciones de las políticas AWS gestionadas

Vea los detalles sobre las actualizaciones de las políticas AWS administradas HealthLake desde el momento en que este servicio comenzó a rastrear estos cambios. Para recibir alertas automáticas sobre los cambios en esta página, suscríbase al RSS feed de la página del historial del HealthLake documento.

Cambio	Descripción	Fecha
AmazonHealthLakeFullAccess	`AmazonHealthLakeFullAccess` política requerida para permitir el acceso completo a HealthLake.	14 de noviembre de 2022
AmazonHealthLakeReadOnlyAccess	`AmazonHealthLakeReadOnlyAccess` política requerida para el acceso de solo lectura a. HealthLake	14 de noviembre de 2022
HealthLake comenzó a rastrear los cambios	HealthLake comenzó a realizar un seguimiento de los cambios de sus políticas AWS gestionadas.	14 de noviembre de 2022

Solución de problemas de AWS HealthLake identidad y acceso

Utilice la siguiente información como ayuda para diagnosticar y solucionar los problemas más comunes que pueden surgir al trabajar con HealthLake yIAM.

Temas

- No estoy autorizado a realizar ninguna acción en AWS HealthLake
- No estoy autorizado a realizar iam: PassRole
- Quiero permitir que personas ajenas a mi AWS cuenta accedan a mis AWS HealthLake recursos

No estoy autorizado a realizar ninguna acción en AWS HealthLake

Si AWS Management Console le indica que no está autorizado a realizar una acción, debe ponerse en contacto con su administrador para obtener ayuda. Su administrador es la persona que le facilitó su nombre de usuario y contraseña.

El siguiente ejemplo de error se produce cuando el `mateojackson` IAM usuario intenta usar la consola para ver los detalles de un *`my-example-widget`* recurso ficticio, pero no tiene los `healthlake:`*`GetWidget`* permisos ficticios.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 healthlake:GetWidget on resource: my-example-widget
```

En este caso, Mateo pide a su administrador que actualice sus políticas de forma que pueda obtener acceso al recurso *my-example-widget* mediante la acción `healthlake:GetWidget`.

No estoy autorizado a realizar iam: PassRole

Si recibe un error que indica que no tiene autorización para realizar la acción `iam:PassRole`, las políticas deben actualizarse a fin de permitirle pasar un rol a HealthLake.

Algunas Servicios de AWS permiten transferir una función existente a ese servicio en lugar de crear una nueva función de servicio o una función vinculada a un servicio. Para ello, debe tener permisos para transferir el rol al servicio.

En el siguiente ejemplo, el error se produce cuando un usuario de IAM denominado `marymajor` intenta utilizar la consola para realizar una acción en HealthLake. Sin embargo, la acción requiere que el servicio cuente con permisos que otorguen un rol de servicio. Mary no tiene permisos para transferir el rol al servicio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

En este caso, las políticas de Mary se deben actualizar para permitirle realizar la acción `iam:PassRole`.

Si necesita ayuda, póngase en contacto con su administrador. AWS El gestionador es la persona que le proporcionó las credenciales de inicio de sesión.

Quiero permitir que personas ajenas a mi AWS cuenta accedan a mis AWS HealthLake recursos

Puedes crear un rol que los usuarios de otras cuentas o las personas externas a la organización puedan utilizar para acceder a sus recursos. Puedes especificar una persona de confianza para que asuma el rol. En el caso de los servicios que admiten políticas basadas en recursos o listas de control de acceso (ACLs), puedes usar esas políticas para permitir que las personas accedan a tus recursos.

Para obtener más información, consulte lo siguiente:

- Para saber si HealthLake es compatible con estas funciones, consulte. ¿Cómo AWS HealthLake funciona con IAM
- Para obtener información sobre cómo proporcionar acceso a los recursos de su propiedad, consulte Proporcionar acceso a un IAM usuario en otro Cuenta de AWS de su propiedad en la Guía del IAM usuario. Cuentas de AWS
- Para obtener información sobre cómo proporcionar acceso a tus recursos a terceros Cuentas de AWS, consulta Cómo permitir el acceso a recursos que Cuentas de AWS son propiedad de terceros en la Guía del IAM usuario.
- Para obtener información sobre cómo proporcionar acceso mediante una federación de identidades, consulte Proporcionar acceso a usuarios autenticadoPara que su aplicación pueda acceder a s externamente (federación de identidades) en la Guía del usuario de IAM.
- Para saber la diferencia entre usar roles y políticas basadas en recursos para el acceso entre cuentas, consulta el tema Acceso a recursos entre cuentas IAM en la Guía del IAM usuario.

Registro de llamadas a la API de AWS HealthLake API con AWS CloudTrail

AWS HealthLake está integrado con AWS CloudTrail un servicio que proporciona un registro de las acciones realizadas por un usuario, un rol o un AWS servicio en HealthLake. CloudTrail captura todas API las llamadas HealthLake como eventos. Las llamadas capturadas incluyen las llamadas desde la HealthLake consola y las llamadas en código a las HealthLake API operaciones. Si crea una ruta, puede habilitar la entrega continua de CloudTrail eventos a un bucket de Amazon S3, incluidos los eventos para HealthLake. Si no configura una ruta, podrá ver los eventos más recientes en la CloudTrail consola, en el historial de eventos. Con la información recopilada por usted CloudTrail, puede determinar a HealthLake qué dirección IP se realizó la solicitud, quién la realizó, cuándo se realizó y detalles adicionales.

Para obtener más información CloudTrail, consulte la Guía AWS CloudTrail del usuario.

AWS HealthLake Información en CloudTrail

CloudTrail está activado en tu AWS cuenta al crearla. Cuando se produce una actividad en HealthLake, esa actividad se registra en un CloudTrail evento junto con otros eventos de AWS servicio en el historial de eventos. Puede ver, buscar y descargar los últimos eventos de la cuenta de AWS . Para obtener más información, consulte Visualización de eventos con el historial de CloudTrail eventos.

Para tener un registro continuo de los eventos de tu AWS cuenta, incluidos los eventos de tu cuenta HealthLake, crea una ruta. Un rastro permite CloudTrail entregar archivos de registro a un bucket de Amazon S3. De manera predeterminada, cuando se crea un registro de seguimiento en la consola, el registro de seguimiento se aplica a todas las regiones de AWS. La ruta registra los eventos de todas las regiones de la AWS partición y envía los archivos de registro al bucket de Amazon S3 que especifique. Además, puede configurar otros AWS servicios para analizar más a fondo los datos de eventos recopilados en los CloudTrail registros y actuar en función de ellos. Para más información, consulte los siguientes temas:

- Introducción a la creación de registros de seguimiento
- CloudTrail Integraciones y servicios compatibles
- Configuración de Amazon SNS Notifications para CloudTrail
- Recibir archivos de CloudTrail registro de varias regiones y recibir archivos de CloudTrail registro de varias cuentas

Todas HealthLake las acciones se registran CloudTrail y se documentan en la HealthLake APIReferencia y en esta Guía para desarrolladores para las acciones realizadas con el FHIR RESTAPI. Por ejemplo, las llamadas a las siguientes acciones generan entradas en los archivos de CloudTrail registro:

- `DescribeFHIRImportJob`
- `DescribeFHIRExportJob`
- `StartFHIRImportJob`
- `ListFHIRImportJobs`
- `StartFHIRExportJob`
- `ListFHIRExportJobs`
- `CreateFHIRDatastore`
- `ListFHIRDatastores`
- `DeleteFHIRDatastore`
- `DescribeFHIRDatastore`
- `UpdateResource`
- `CreateResource`
- `DeleteResource`

- `ReadResource`
- `GetCapabilities`
- `SearchWithGet`
- `SearchWithPost`

Cada entrada de registro o evento contiene información sobre quién generó la solicitud. La información de identidad del usuario lo ayuda a determinar lo siguiente:

- Si la solicitud se realizó con credenciales de usuario root o AWS Identity and Access Management (IAM).
- Si la solicitud se realizó con credenciales de seguridad temporales de un rol o fue un usuario federado.
- Si la solicitud la realizó otro AWS servicio.

Para obtener más información, consulte el CloudTrail userIdentityElemento.

Descripción de las entradas de los archivos de AWS HealthLake registro

Un rastro es una configuración que permite la entrega de eventos como archivos de registro a un bucket de Amazon S3 que especifique. CloudTrail Los archivos de registro contienen una o más entradas de registro. Un evento representa una solicitud única de cualquier fuente e incluye información sobre la acción solicitada, la fecha y la hora de la acción, los parámetros de la solicitud, etc. CloudTrail Los archivos de registro no son un registro ordenado de las API llamadas públicas, por lo que no aparecen en ningún orden específico.

El siguiente ejemplo muestra una entrada de CloudTrail registro que demuestra la `CreateFHIRDatastore` acción.

```
{
    "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"
```

```
}
```

Validación de conformidad para AWS HealthLake

Los auditores externos evalúan la seguridad y el cumplimiento AWS HealthLake como parte de varios programas de AWS cumplimiento. Porque HealthLake esto incluyeHIPAA.

Para saber si un programa de cumplimiento Servicio de AWS está dentro del ámbito de aplicación de programas de cumplimiento específicos, consulte Servicios de AWS Alcance por programa de de cumplimiento y elija el programa de cumplimiento que le interese. Para obtener información general, consulte Programas de AWS cumplimiento > Programas AWS .

Puede descargar informes de auditoría de terceros utilizando AWS Artifact. Para obtener más información, consulte Descarga de informes en AWS Artifact .

Su responsabilidad de cumplimiento al Servicios de AWS utilizarlos viene determinada por la confidencialidad de sus datos, los objetivos de cumplimiento de su empresa y las leyes y reglamentos aplicables. AWS proporciona los siguientes recursos para ayudar con el cumplimiento:

- Cumplimiento de seguridad y gobernanza: en estas guías se explican las consideraciones de arquitectura y se proporcionan pasos para implementar las características de seguridad y cumplimiento.
- Diseñando una arquitectura basada en la HIPAA seguridad y el cumplimiento en Amazon Web Services: en este documento técnico se describe cómo pueden utilizar las empresas AWS para crear HIPAA aplicaciones aptas.

  > **Note**
  >
  > No todos son aptos. Servicios de AWS HIPAA Para obtener más información, consulta la Referencia de servicios HIPAA aptos.

- AWS Recursos de de cumplimiento: esta colección de libros de trabajo y guías puede aplicarse a su industria y ubicación.
- AWS Guías de cumplimiento para clientes: comprenda el modelo de responsabilidad compartida desde el punto de vista del cumplimiento. En las guías se resumen las mejores prácticas para garantizar la seguridad Servicios de AWS y se orientan a los controles de seguridad en varios marcos (incluidos el Instituto Nacional de Estándares y Tecnología (NIST), el Consejo de Normas

de Seguridad de la Industria de Tarjetas de Pago (PCI) y la Organización Internacional de Normalización (ISO)).

- Evaluación de los recursos con reglas en la guía para AWS Config desarrolladores: el AWS Config servicio evalúa en qué medida las configuraciones de los recursos cumplen con las prácticas internas, las directrices del sector y las normas.
- AWS Security Hub— Esto Servicio de AWS proporciona una visión completa del estado de su seguridad interior AWS. Security Hub utiliza controles de seguridad para evaluar sus recursos de AWS y comprobar su cumplimiento con los estándares y las prácticas recomendadas del sector de la seguridad. Para obtener una lista de los servicios y controles compatibles, consulta la Referencia de controles de Security Hub.
- Amazon GuardDuty: Servicio de AWS detecta posibles amenazas para sus cargas de trabajo Cuentas de AWS, contenedores y datos mediante la supervisión de su entorno para detectar actividades sospechosas y maliciosas. GuardDuty puede ayudarlo a cumplir con varios requisitos de conformidad, por ejemplo PCIDSS, cumpliendo con los requisitos de detección de intrusiones exigidos por ciertos marcos de cumplimiento.
- AWS Audit Manager— Esto le Servicio de AWS ayuda a auditar continuamente su AWS consumo para simplificar la gestión del riesgo y el cumplimiento de las normativas y los estándares del sector.

Resiliencia en AWS HealthLake

La infraestructura AWS global se basa en AWS regiones y zonas de disponibilidad. AWS Las regiones proporcionan varias zonas de disponibilidad aisladas y separadas físicamente, que están conectadas mediante redes de baja latencia, alto rendimiento y alta redundancia. Con las zonas de disponibilidad, puede diseñar y utilizar aplicaciones y bases de datos que realizan una conmutación por error automática entre las zonas sin interrupciones. Las zonas de disponibilidad tienen una mayor disponibilidad, tolerancia a errores y escalabilidad que las infraestructuras tradicionales de uno o varios centros de datos.

Para obtener más información sobre AWS las regiones y las zonas de disponibilidad, consulte Infraestructura global.AWS

Además de la infraestructura AWS global, HealthLake ofrece varias funciones para ayudarlo a satisfacer sus necesidades de respaldo y resiliencia de datos.

Seguridad de infraestructuras en AWS HealthLake

Como servicio gestionado, AWS HealthLake está protegido por los procedimientos de seguridad de la red AWS global que se describen en el documento técnico Amazon Web Services: Overview of Security Processes.

Utiliza las API llamadas AWS publicadas para acceder a HealthLake través de la red. Los clientes deben ser compatibles con Transport Layer Security (TLS) 1.0 o una versión posterior. Recomendamos la versión TLS 1.2 o una versión posterior. Los clientes también deben admitir conjuntos de cifrado con total confidencialidad (PFS), como Ephemeral Diffie-Hellman () o Elliptic Curve Ephemeral Diffie-Hellman (DHE). ECDHE La mayoría de los sistemas modernos como Java 7 y posteriores son compatibles con estos modos.

Además, las solicitudes deben estar firmadas mediante un ID de clave de acceso y una clave de acceso secreta que esté asociada a una entidad principal de IAM. También puedes utilizar AWS Security Token Service (AWS STS) para generar credenciales de seguridad temporales para firmar solicitudes.

Prácticas de seguridad recomendadas para AWS HealthLake

AWS HealthLake proporciona una serie de características de seguridad que debe tener en cuenta a la hora de desarrollar e implementar sus propias políticas de seguridad. Las siguientes prácticas recomendadas son directrices generales y no constituyen una solución de seguridad completa. Puesto que es posible que estas prácticas recomendadas no sean adecuadas o suficientes para el entorno, considérelas como consideraciones útiles en lugar de como normas.

- Implemente el acceso a los privilegios mínimos.
- Siempre que sea posible, utilice Customer-Managed-Keys (CMKs) para cifrar sus datos. Para obtener más informaciónCMKs, consulte Amazon Key Management Service.
- Utilice Buscar conPOST, no Buscar con, GET cuando realice consultas PII en su almacén de datos PHI o en él.
- Limite el acceso a las funciones de auditoría importantes y confidenciales.
- Al crear recursos mediante la actualización o la importación masivaAPIs, no utilice PHI ni PII incluya los nombres de los almacenes de datos y los trabajos en ningún campo visible ni en el FHIR identificador lógico (LID).
- Al enviar solicitudes de creación, lectura, actualización, eliminación o búsqueda, no las utilices PHI en el HTTP encabezado.

- AWS CloudTrail Actívela para auditar el AWS HealthLake uso y garantizar que no haya actividad inesperada.
- Revise las prácticas recomendadas para utilizar los buckets de Amazon S3 de forma segura. Para obtener más información, consulte las prácticas recomendadas de seguridad en la guía del usuario de Amazon S3.

AWS HealthLake puntos finales y cuotas

Las siguientes secciones contienen información sobre AWS HealthLake las cuotas y los puntos finales. Puede solicitar aumentos de cuota para cuotas ajustables mediante Service Quotas. Para obtener más información, consulte Solicitud de un aumento de cuota en la Guía del usuario de Service Quotas.

Puntos de conexión de servicio

La tabla muestra los puntos finales HealthLake de servicio disponibles en una región determinada.

Nombre de la región	Región	Punto de conexión	Protocolo
Este de EE. UU. (Ohio)	us-east-2	healthlake.us-east-2.amazonaws.com healthlake-fips.us-east-2.amazonaws.com	HTTPS HTTPS
Este de EE. UU. (Norte de Virginia)	us-east-1	healthlake.us-east-1.amazonaws.com healthlake-fips.us-east-1.amazonaws.com	HTTPS HTTPS
Oeste de EE. UU. (Oregón)	us-west-2	healthlake.us-west-2.amazonaws.com healthlake-fips.us-west-2.amazonaws.com	HTTPS HTTPS
Asia-Pacífico (Bombay)	ap-south-1	healthlake.ap-south-1.amazonaws.com	HTTPS
Asia-Pacífico (Sídney)	ap-southeast-2	healthlake.ap-southeast-2.amazonaws.com	HTTPS

Nombre de la región	Región	Punto de conexión	Protocolo	
Europa (Londres)	eu-west-2	healthlake.eu-west-2.amazonaws.com	HTTPS	

Cuotas de servicio para HealthLake

Las siguientes son las cuotas predeterminadas para HealthLake.

Nombre	Valor predeterminado	Ajusta	Descripción
Número de caracteres de una nota médica	Cada región admitida: 10 000	No	El número máximo de caracteres de una nota médica individual dentro del tipo de DocumentReference recurso (POST/PUTsolicitudes).
Número de trabajos simultáneos de StartFHIRImport Job	Cada región admitida: 1	No	El número máximo de trabajos StartFHIRImport Job simultáneos.
Número de trabajos concurrentStartFHIRExportJob	Cada región admitida: 1	No	El número máximo de trabajos StartFHIRExport Job simultáneos.
Número de almacenes de datos por cuenta	Cada región admitida: 10	Sí	El número máximo predeterminado de almacenes de datos activos por cuenta.
Número de archivos en un StartFHIRImport Job	Cada región admitida: 10 000	No	El número máximo de archivos en un StartFHIRImport Job.

Nombre	Valor predeterminado	Ajusta	Descripción
Número de recursos por paquete	Cada región admitida: 160	No	El número máximo de recursos permitido en una solicitud de paquete.
Tasa de solicitudes de paquetes por cuenta	Cada región admitida: 20	Sí	El número máximo de solicitudes de POST paquetes que puedes realizar por segundo por cuenta.
Tasa de solicitudes de paquetes por almacén de datos	Cada región admitida: 10	Sí	El número máximo de solicitudes de POST paquetes que puede realizar por segundo por almacén de datos. Los almacenes de datos creados antes del 21 de agosto de 2023 se limitarán a 1 solicitud por segundo.
Tasa de solicitudes de ancelFHIRExport trabajo C que se utilizan DELETE por cuenta	Cada región admitida: 1	No	El número máximo de solicitudes de C ancelFHIRExport Job DELETE que puede realizar por minuto por cuenta.
Tasa de reateFHIRDatastore solicitudes de C por cuenta	Cada región admitida: 1	No	El número máximo de reateFHIRDatastore solicitudes C que puedes realizar por minuto por cuenta.

Nombre	Valor predeterminado	Ajusta	Descripción
Tasa de solicitudes DELETE por cuenta	Cada región admitida: 2000	Sí	El número máximo de DELETE solicitudes que puedes realizar por segundo por cuenta.
Tasa de DELETE solicitudes por almacén de datos	Cada región admitida: 1000	Sí	El número máximo de DELETE solicitudes que puede realizar por segundo por almacén de datos. Los almacenes de datos creados antes del 21 de agosto de 2023 se limitarán a 100 solicitudes por segundo.
Tasa de D eleteFHIRDatastore solicitudes por cuenta	Cada región admitida: 1	No	El número máximo de eleteFHIRDatastore solicitudes D que puedes realizar por minuto por cuenta.
Tasa de D escribeFHIRDatastore solicitudes por cuenta	Cada región admitida: 10	No	El número máximo de escribeFHIRDatastore solicitudes D que puedes realizar por segundo por cuenta.
Tasa de solicitudes de escribeFHIRExport trabajo tipo D por cuenta	Cada región admitida: 10	No	El número máximo de solicitudes de escribeFHIRExport trabajo D que puede realizar por segundo por cuenta.

Nombre	Valor predeterminado	Ajusta	Descripción
Tasa de solicitudes de escribeFHIRExport trabajo D que se utilizan GET por cuenta	Cada región admitida: 10	No	El número máximo de solicitudes de escribeFHIRExport trabajo en D GET que puede realizar por segundo por cuenta.
Tasa de solicitudes de escribeFHIRImport trabajo tipo D por cuenta	Cada región admitida: 10	No	El número máximo de solicitudes de escribeFHIRImport trabajo D que puede realizar por segundo por cuenta.
Tasa de solicitudes de descubrimiento por cuenta	Cada región admitida: 10	No	El número máximo de solicitudes de descubrimiento que puede realizar por minuto por cuenta.
Tasa de solicitudes GET por cuenta	Cada región admitida: 6000	Sí	El número máximo de GET solicitudes que puede realizar por segundo por cuenta.
Tasa de GET solicitudes por almacén de datos	Cada región admitida: 3000	Sí	El número máximo de GET solicitudes que puede realizar por segundo por almacén de datos. Los almacenes de datos creados antes del 21 de agosto de 2023 se limitarán a 100 solicitudes por segundo.

Nombre	Valor predeterminado	Ajusta	Descripción
Tasa de GetCapabilities solicitudes por cuenta	Cada región admitida: 10	No	El número máximo de GetCapabilities solicitudes que puedes realizar por segundo por cuenta.
Tasa de L istFHIRDatastores solicitudes por cuenta	Cada región admitida: 10	No	El número máximo de istFHIRDatastores solicitudes L que puedes realizar por segundo por cuenta.
Tasa de solicitudes de L istFHIRExport Jobs por cuenta	Cada región admitida: 10	No	El número máximo de solicitudes de L istFHIRExport Jobs que puede realizar por segundo por cuenta.
Tasa de solicitudes de L istFHIRImport Jobs por cuenta	Cada región admitida: 10	No	El número máximo de solicitudes de L istFHIRImport Jobs que puede realizar por segundo por cuenta.
Tasa de ListTagsforResource solicitudes por cuenta	Cada región admitida: 10	No	El número máximo de ListTagsforResource solicitudes que puedes realizar por segundo por cuenta.
Tasa de solicitudes POST por cuenta	Cada región admitida: 2000	Sí	El número máximo de POST solicitudes que puedes realizar por segundo por cuenta.

Nombre	Valor predeterminado	Ajusta	Descripción
Tasa de POST solicitudes por almacén de datos	Cada región admitida: 1000	Sí	El número máximo de POST solicitudes que puede realizar por segundo por almacén de datos. Los almacenes de datos creados antes del 21 de agosto de 2023 se limitarán a 100 solicitudes por segundo.
Tasa de solicitudes PUT por cuenta	Cada región admitida: 2000	Sí	El número máximo de PUT solicitudes que puede realizar por segundo por cuenta.
Tasa de PUT solicitudes por almacén de datos	Cada región admitida: 1000	Sí	El número máximo de PUT solicitudes que puede realizar por segundo por almacén de datos. Los almacenes de datos creados antes del 21 de agosto de 2023 se limitarán a 100 solicitudes por segundo.
Tasa de solicitudes de tartFHIRExport trabajo S por cuenta	Cada región admitida: 1	No	El número máximo de solicitudes de S tartFHIRExport Job que puede realizar por minuto por cuenta.

Nombre	Valor predeterminado	Ajusta	Descripción
Tasa de solicitudes de tartFHIRExport trabajo S que se utilizan POST por cuenta	Cada región admitida: 1	No	El número máximo de solicitudes de S tartFHIRExport Job POST que puede realizar por minuto por cuenta.
Tasa de solicitudes de tartFHIRImport trabajo S por cuenta	Cada región admitida: 1	No	El número máximo de solicitudes de S tartFHIRImport Job que puede realizar por minuto por cuenta.
Tasa de TagResource solicitudes por cuenta	Cada región admitida: 10	No	El número máximo de TagResource solicitudes que puedes realizar por segundo.
Tasa de UntagResource solicitudes por cuenta	Cada región admitida: 10	No	El número máximo de UntagResource solicitudes que puedes realizar por segundo por cuenta.
Porcentaje de solicitudes de búsqueda que se utilizan GET por cuenta	Cada región admitida: 200	Sí	El número máximo de solicitudes de búsqueda GET que puedes realizar por segundo por cuenta.
Tasa de solicitudes de búsqueda que se utilizan GET por almacén de datos	Cada región admitida: 100	Sí	El número máximo de solicitudes de búsqueda GET que puede realizar por segundo por almacén de datos.

Nombre	Valor predeterminado	Ajusta	Descripción
Tasa de solicitudes de búsqueda POST por cuenta	Cada región admitida: 200	Sí	El número máximo de solicitudes de búsqueda POST que puede realizar por segundo.
Tasa de solicitudes de búsqueda que se utilizan POST por almacén de datos	Cada región admitida: 100	Sí	El número máximo de solicitudes de búsqueda POST que puede realizar por segundo por almacén de datos.
Tamaño del archivo importado individual	Cada región admitida: 5 gigabytes	No	El tamaño máximo (en GB) de un archivo individual incluido en un StartFHIRImport Job.
Tamaño total del trabajo de importación	Cada región admitida: 500 gigabytes	No	El tamaño máximo (en GB) de todos los archivos incluidos en el trabajo de importación.

Resolución de problemas

La siguiente documentación puede ayudarle a solucionar los problemas que pueda tener con el uso. AWS HealthLake

Temas

¿Por qué no puedo crear un almacén HealthLake de datos?

El 14 de noviembre de 2022, HealthLake actualizé los IAM permisos necesarios para crear un nuevo banco de datos. Si no has actualizado las políticas asociadas al usuario o rol al que HealthLake accede, aparece el siguiente error.

```
AccessDeniedException: Insufficient Lake Formation permission(s): Required Database on
 Catalog
```

Para ver los requisitos IAM de política actualizados para crear un banco de datos, consulte la política AWS administrada: AmazonHealthLakeFullAccess. Para obtener step-by-step instrucciones sobre cómo agregar estas políticas a su IAM usuario o rol, consulteConfigurar los permisos para empezar a usar AWS HealthLake.

Para crear un almacén de datos, también necesitas usar una clave simétrica propiedad del cliente o de Amazon. KMS Asegúrate de tener los permisos correctos en tu política. IAM Para obtener más información AWS KMS, consulta AWS Key Management Servicela Guía para AWS Key Management Service desarrolladores.

Se ha superado el número de almacenes de datos permitidos por cuenta

HealthLake tiene una cuota de 10 almacenes de datos por cuenta. Para obtener información sobre cómo solicitar un aumento de cuota, visita AWS Support Center.

¿Cómo creo una autorización para el FHIR RESTfulAPIs?

Los usuarios deben utilizar un proceso de firma de la versión 4 de Signature para añadir la autenticación a HealthLake API las solicitudes enviadas a través de un HTTP cliente. Para obtener más información, consulte el proceso de firma de la versión 4 de Signature.

Para crear la AWS SDK autorización sigv4 mediante Python, cree un script similar al siguiente ejemplo.

```
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())
```

Puede encontrar información adicional sobre el uso de la autorización sigv4 AWS SDK para Python en el tema de credenciales de Boto3.

Mis datos no están en formato FHIR R4, ¿puedo seguir utilizándolos? HealthLake

Solo los datos con formato FHIR R4 se pueden importar a un HealthLake almacén de datos. Para ver una lista de socios que ofrecen productos para ayudar a los usuarios a transformar sus datos, consulte AWS HealthLake Socios.

¿Por qué recibo AccessDenied errores cuando utilizo un almacén FHIR RESTful APIs de datos cifrado con una KMS clave gestionada por el cliente?

Se requieren permisos tanto para la clave como para las IAM políticas administradas por el cliente para que un usuario o rol pueda acceder a un almacén de datos. Un usuario debe tener los IAM permisos necesarios para usar una clave administrada por el cliente. Si un usuario ha revocado o retirado una concesión que daba HealthLake permiso para usar la KMS clave gestionada por el cliente, HealthLake devolverá un AccessDenied error.

HealthLake debe tener el permiso para acceder a los datos de los clientes, cifrar FHIR los nuevos recursos importados a un almacén de datos y descifrar los FHIR recursos cuando se soliciten.

Para obtener más información, consulte Solución de problemas de acceso a claves.

¿Por qué falló mi importación?

Si la importación se realiza correctamente, se generará una carpeta con los inputFileName archivos.ndjson de salida; sin embargo, es posible que no se puedan importar registros individuales. Cuando esto suceda, se generará una segunda FAILURE carpeta con un manifiesto de los registros que no se pudieron importar. La ubicación de salida del trabajo para acceder al archivo de manifiesto es JobProperties. JobOutputDataConfig.Configuración S3.S3Uri.

Este archivo de manifiesto contiene detalles sobre el resultado del trabajo, como la ubicación de todas las respuestas correctas (. successOutput successOutputS3Uri), la ubicación de todas las respuestas fallidas (. failureOutput failureOutputS3Uri) y métricas de trabajo adicionales. El contenido del archivo de manifiesto se puede analizar mediante programación. En el siguiente ejemplo de archivo de manifiesto se enumeran los buckets de entrada y salida de Amazon S3, así como información sobre el número de recursos escaneados y cuántos se importaron correctamente.

```
            {
            "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
            }
```

Para analizar por qué falló un trabajo de importación, utilice el escribeFHIRImport trabajo D API para analizar el JobProperties. Se recomienda lo siguiente:

- Si el estado es FAILED y aparece un mensaje, los errores están relacionados con parámetros del trabajo, como el tamaño de los datos de entrada o el número de archivos de entrada, que superan HealthLake las cuotas.
- Si el estado del trabajo de importación es COMPLETED _ WITH _ERRORS, consulte el archivo de manifiesto, Manifest.json, para obtener información sobre los archivos que no se importaron correctamente.
- Si el estado del trabajo de importación es FAILED y no hay ningún mensaje, vaya a la ubicación de salida del trabajo para acceder al archivo de manifiesto, Manifest.json.

Para cada archivo de entrada, hay un archivo de salida con un nombre de archivo de entrada para cualquier recurso que no se pueda importar. Las respuestas contienen el número de línea (lineId) correspondiente a la ubicación de los datos de entrada, el objeto de FHIR respuesta (UpdateResourceResponse) y el código de estado (statusCode) de la respuesta.

Un ejemplo de archivo de salida tendría el siguiente aspecto:

```
{"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}
```

El ejemplo muestra que hubo errores en las líneas 3, 4, 7, 9 y 15 de las líneas de entrada correspondientes del archivo de entrada. Para cada una de esas líneas, las explicaciones son las siguientes:

- En la línea 3, la respuesta explica lo que resourceType se proporciona en la línea 3 del archivo de entrada no es válido.
- En la línea 5, la respuesta explica que hay un error de FHIR validación en la línea 5 del archivo de entrada.
- En la línea 7, la respuesta explica que hay un problema de validación y resourceId se proporciona como entrada.
- En la línea 9, la respuesta explica que el archivo de entrada debe contener un identificador de recurso válido.
- En la línea 15, la respuesta del archivo de entrada es que el archivo no tiene un JSON formato válido.

¿Cómo puedo encontrar DocumentReference los recursos que no se han podido procesar?

Si un DocumentReference recurso no era válido, HealthLake proporcionará una extensión que indique un error de validación en lugar del NLP resultado médico integrado. Para encontrar DocumentReference los recursos que provocaron un error de validación durante el NLP procesamiento, los clientes pueden utilizar HealthLake la función de búsqueda con la clave de búsqueda cm-decoration-statusy el valor de búsqueda VALIDATION_ ERROR. Esta búsqueda mostrará una lista de todos DocumentReference los recursos que han provocado errores de

validación, junto con un mensaje de error que describe la naturaleza del error. La estructura del campo de extensión en los DocumentReference recursos con errores de validación se parecerá a la del siguiente ejemplo.

```
"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/"
          }
    ]
```

También ERROR puede aparecer un VALIDATION_ si la NLP decoración crea más de 10 000 objetos anidados. Cuando esto sucede, el documento debe dividirse en documentos más pequeños antes de procesarlo.

Migración de un almacén de datos existente para usar Amazon Athena

Los almacenes de datos creados antes del 14 de noviembre de 2022 funcionan, pero no se pueden consultar en Athena mediante ellos. SQL Para consultar un banco de datos preexistente con Athena, primero debe migrarlo a un banco de datos nuevo.

Para migrar los datos a un nuevo banco de datos

1. Cree un nuevo banco de datos.
2. Exporte los datos del depósito preexistente a un bucket de Amazon S3.
3. Importe los datos al nuevo almacén de datos desde el bucket de Amazon S3.

La exportación de datos a un bucket de Amazon S3 conlleva un coste adicional. El cargo adicional depende del tamaño de los datos que exporte.

Conectar los resultados de búsqueda en Athena a otros servicios AWS

Puede tener problemas al compartir los resultados de búsqueda de Athena con otros AWS servicios.

El problema puede producirse cuando se utiliza `json_extract[1]` como parte de una consulta de SQL búsqueda.

Para solucionar este problema, debes actualizar a`CATVAR`.

Este problema puede producirse al intentar crear resultados guardados, una tabla (estática) o una vista (dinámica).

La consola Athena no funciona después de importar datos a un nuevo almacén de datos

Después de importar los datos a un nuevo banco de datos, es posible que los datos no estén disponibles para su uso inmediato. Esto es para dar tiempo a que los datos se integren en las tablas de iceberg. Inténtelo de nuevo más tarde.

¿Por qué aparece un error de permisos de Lake Formation: lakeformation: PutDataLakeSettings al añadir un nuevo administrador de lago de datos?

Si su IAM usuario o rol contiene la política `AWSLakeFormationDataAdmin` AWS administrada, no puede agregar nuevos administradores de lagos de datos. Aparecerá un mensaje de error que contiene lo siguiente:

```
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 política AWS gestionada `AdministratorAccess` es necesaria para añadir un IAM usuario o un rol como administrador AWS del lago de datos de Lake Formation. Si su IAM usuario o rol también lo contiene, `AWSLakeFormationDataAdmin` la acción fallará. La política `AWSLakeFormationDataAdmin` AWS gestionada contiene una denegación explícita de la API operación de AWS Lake Formation,`PutDataLakeSetting`.

Incluso los administradores con acceso total al AWS uso de la política `AdministratorAccess` AWS administrada pueden estar limitados por la `AWSLakeFormationDataAdmin` política.

¿Cómo activo la función HealthLake de procesamiento de lenguaje natural integrada?

A partir del 20 de febrero de 2023, el comportamiento predeterminado de HealthLake los almacenes de datos cambió.

Almacenes de datos actuales: todos los almacenes de HealthLake datos actuales dejarán de utilizar el procesamiento de lenguaje natural (NLP) en los recursos codificados en base64`DocumentReference`. Esto significa que no se analizarán `DocumentReference` los nuevos recursos ni se generarán nuevos recursos a partir del texto del tipo de recurso. NLP `DocumentReference` En el caso de `DocumentReference` los recursos existentes, los datos y los recursos generados mediante ellos NLP permanecen, pero no se actualizarán después del 20 de febrero de 2023.

Nuevos almacenes de datos: HealthLake los almacenes de datos creados después del 20 de febrero de 2023 no procesarán el lenguaje natural (NLP) en recursos codificados en base64`DocumentReference`.

Para activar esta función, debe crear un caso utilizando. AWS Support Center Console Para crear tu caso, inicia sesión en tu y Cuenta de AWS, a continuación, selecciona Crear caso. Para obtener más información sobre la creación de un caso y la gestión de casos, consulte Creación de casos de soporte y gestión de casos en la Guía del Soporte usuario.

El estado de mi almacén de datos no cambia desde el de Creación

Si intenta crear un nuevo banco de HealthLake datos y su estado no cambia desde Crear, debe actualizar Athena para usar el. AWS Glue Data Catalog

Para obtener más información, consulte Actualización al catálogo de datos de AWS Glue step-by-step en la Guía del usuario de Amazon Athena.

Después de actualizar correctamente el AWS Glue Data Catalog, ahora puede crear un almacén de datos.

Para eliminar el antiguo almacén de datos, comience por crear un caso utilizando AWS Support Center Console. Para crear su caso, inicie sesión en su y Cuenta de AWS, a continuación, seleccione Crear caso. Para obtener más información, consulte Creación de casos de soporte y administración de casos en la Guía del Soporte usuario.

El estado de creación de mi almacén de SDK datos devuelve un estado de excepción o desconocido

Actualícelo SDK a la última versión si las API llamadas al almacén de datos de su lista o a las de descripción muestran una excepción o un estado desconocido del almacén de datos.

Al realizar una FHIR POST API operación con un documento de 10 MB, aparece el HealthLake error 413Request Entity Too Large.

AWS HealthLake tiene un API límite de creación y actualización sincrónicas de 5 MB para evitar el aumento de las latencias y los tiempos de espera.

Puede ingerir documentos grandes, de hasta 164 MB, utilizando el formato binario mediante la importación masiva. ResourceType API

Historial de documentos de la guía para AWS HealthLake desarrolladores

En la siguiente tabla se describen los cambios en la documentación de las AWS HealthLake versiones.

- APIversión: más reciente
- Última actualización de la documentación: 25 de octubre de 2024

Cambio	Descripción	Fecha
HealthLake ahora los apoyos FHIR `history` y las interacciones `vread`	HealthLake ahora admite la FHIR `history` interacción para recuperar el historial de un recurso en particular y la `vread` interacción para realizar una lectura específica de una versión de un recurso.	25 de octubre de 2024
HealthLake ahora admite nuevos parámetros de FHIR búsqueda, extensiones y tipos de recursos.	HealthLake ahora admite nuevos parámetros FHIR de búsqueda, extensiones y tipos de recursos.	9 de diciembre de 2023
HealthLake ahora es compatible con el FHIR marco SMART on	HealthLake ahora admite la creación SMART en almacenes de HealthLake datos FHIR habilitados.	31 de mayo de 2023
HealthLake ahora admite la validación de perfiles	HealthLake ahora admite la validación FHIR de perfiles.	31 de mayo de 2023
HealthLake ahora admite `export`	HealthLake ahora admite la exportación de archivos mediante la FHIR REST API operación`export`.	31 de mayo de 2023

Región Asia-Pacífico (Bombay)	AWS HealthLake ya está disponible en la región de Asia Pacífico (Bombay).	4 de abril de 2023
El procesamiento integrado del lenguaje natural está desactivado	HealthLake desactivó el procesamiento de lenguaje natural integrado (NLP) en todos los almacenes de datos a partir del 20 de febrero de 2023.	20 de febrero de 2023
HealthLake se integra con Amazon Athena	Ahora puede usar Athena para consultar los almacenes de datos creados después del 14 de noviembre de 2022.	14 de noviembre de 2022
El tamaño total del trabajo de importación aumentó	El tamaño total máximo de todos los archivos de una solicitud de S tartFHIRImport Job es ahora de 500 GB.	3 de octubre de 2022
Soporte de paquetes	HealthLake ahora admite el tipo de recurso Bundle para ingerir varios recursos.	5 de agosto de 2022
Cuotas actualizadas para las CRUD operaciones en HealthLake	HealthLake ahora admite límites más altos para CRUD las solicitudes.	14 de julio de 2022
Incluye soporte	HealthLake ahora admite consultas `_include` en el almacén de datos.	14 de julio de 2022
AWS HealthLake ahora está disponible de forma general	HealthLake ahora está disponible de forma general.	30 de julio de 2020

AWS Glosario

Para obtener la AWS terminología más reciente, consulte el AWS glosario de la Glosario de AWS Referencia.

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