aws

Guia do Desenvolvedor

AWS HealthLake

AWS HealthLake: Guia do Desenvolvedor

Table of Contents

O que é AWS HealthLake?

AWS HealthLake é um serviço HIPAA qualificado para ingestão, armazenamento e análise de dados clínicos utilizando a especificação Healthcare Interoperability FHIR (R4).

> **Note**
>
> Depois de 20 de fevereiro de 2023, HealthLake os armazenamentos de dados não usam processamento de linguagem natural integrado (NLP) por padrão. Se você estiver interessado em ativar esse recurso em seu armazenamento de dados, consulte o Como faço para ativar o recurso integrado HealthLake de processamento de linguagem natural do? capítulo Solução de problemas.

Os dados de saúde geralmente estão incompletos e inconsistentes. Geralmente também não é estruturado, com informações contidas em notas clínicas, relatórios de laboratório, pedidos de seguro, imagens médicas, conversas gravadas e dados de séries temporais (por exemplo, EEG traços cardíacos ECG ou cerebrais).

Os profissionais de saúde podem usar HealthLake para armazenar, transformar, consultar e analisar dados na AWS nuvem. Usando os recursos HealthLake integrados de processamento médico de linguagem natural (NLP), você pode analisar textos clínicos não estruturados de diversas fontes. HealthLake transforma dados não estruturados usando modelos de processamento de linguagem natural e fornece recursos avançados de consulta e pesquisa. Você pode usar HealthLake para organizar, indexar e estruturar as informações do paciente de forma segura, compatível e que possa ser auditada.

HealthLake também está integrado ao Amazon Athena e ao AWS Lake Formation. Você pode usar essa integração para consultar seu armazenamento de dados usandoSQL.

Benefícios do AWS HealthLake

Com AWS HealthLake, você pode:

- Ingira dados de saúde de forma rápida e fácil — Você pode importar em massa arquivos Fast Healthcare Interoperability Resources (FHIR) locais, incluindo notas clínicas, relatórios de laboratório, pedidos de seguro e muito mais, para um bucket do Amazon Simple Storage

Service (Amazon S3). Em seguida, você pode usar os dados em aplicativos ou fluxos de trabalho downstream.

- Use as FHIR REST API operações — HealthLake suporta o uso das FHIR REST API operações para realizar operações CRUD (Create/Read/Update/Delete) em seu armazenamento de dados. FHIRa pesquisa também é suportada.
- Armazene seus dados na AWS nuvem de uma forma segura e HIPAA elegível que possa ser auditada — Você pode armazenar dados no FHIR formato para que possam ser facilmente consultados. HealthLake cria uma visão cronológica completa do histórico médico de cada paciente e a estrutura no formato padrão R4FHIR.
- Integração com o Athena — HealthLake a integração com o Athena significa que você pode criar consultas poderosas SQL que podem ser usadas para criar e salvar critérios de filtro complexos. Em seguida, você pode usar esses dados em aplicativos posteriores, como SageMaker IA para treinar um modelo de aprendizado de máquina ou QuickSight a Amazon para criar painéis e visualizações de dados.
- Transforme dados não estruturados usando modelos especializados de aprendizado de máquina (ML) — HealthLake fornece processamento médico integrado de linguagem natural (NLP) usando o Amazon Comprehend Medical. Os dados brutos de texto médico são transformados usando modelos de ML especializados. Esses modelos foram treinados para entender e extrair informações significativas de dados de saúde não estruturados. Com a medicina integradaNLP, você pode extrair automaticamente dados de entidades (por exemplo, procedimentos médicos e medicamentos), relacionamentos entre entidades (por exemplo, um medicamento e sua dosagem) e características de entidades (por exemplo, resultado positivo ou negativo do teste ou hora do procedimento) do seu texto médico. HealthLake em seguida, cria novos recursos com base nas características, signo, sintoma e condição. Eles são adicionados como novos tipos de condição, observação e MedicationStatement recurso.

HealthLake casos de uso

Você pode usar HealthLake para os seguintes aplicativos de saúde:

- Gestão da saúde da população — HealthLake ajuda as organizações de saúde a analisar tendências, resultados e custos da saúde da população. Isso ajuda as organizações a identificar a intervenção mais adequada para uma população de pacientes e a escolher melhores opções de gerenciamento de cuidados.

- Melhorando a qualidade do atendimento — HealthLake ajuda hospitais, seguradoras de saúde e organizações de ciências biológicas a preencher lacunas no atendimento, melhorar a qualidade do atendimento e reduzir custos compilando uma visão completa do histórico médico do paciente.
- Otimizando a eficiência hospitalar — HealthLake oferece aos hospitais as principais ferramentas de análise e aprendizado de máquina para melhorar a eficiência e reduzir o desperdício hospitalar.

Acessando HealthLake

Você pode acessar HealthLake por meio do AWS Management Console, AWS Command Line Interface (AWS CLI) ou do AWS SDKs.

1. AWS Management Console — Fornece uma interface da web que você pode usar para acessar HealthLake.
2. AWS Command Line Interface (AWS CLI) — Fornece comandos para um amplo conjunto de AWS serviços, inclusive HealthLake, e é compatível com Windows, macOS e Linux. Para obter mais informações sobre a instalação do AWS CLI, consulte AWS Command Line Interface.
3. AWS SDKs— AWS fornece SDKs (kits de desenvolvimento de software) que consistem em bibliotecas e código de amostra para várias linguagens e plataformas de programação (Java, Python, Ruby,. NET, iOS, Android e assim por diante). Eles SDKs fornecem uma maneira conveniente de criar acesso programático a HealthLake e. AWS Para obter mais informações, consulte o AWSSDKpara Python.

HIPAAelegibilidade e segurança de dados

Este é um serviço HIPAA qualificado. Para obter mais informações sobre AWS a Lei de Portabilidade e Responsabilidade de Seguros de Saúde dos EUA de 1996 (HIPAA) e o uso de AWS serviços para processar, armazenar e transmitir informações de saúde protegidas (PHI), consulte HIPAA Visão geral.

As conexões que HealthLake contêm informações de identificação pessoal (PII) devem ser criptografadas. Por padrão, todas as conexões a serem HealthLake usadas HTTPSTLS. HealthLake armazena conteúdo criptografado do cliente e opera de acordo com o princípio da Responsabilidade AWS Compartilhada.

Preços

Para obter informações sobre HealthLake preços, consulte a página AWS HealthLake de preços. Para estimar melhor os custos potenciais associados ao HealthLake, você pode usar a calculadora HealthLake de preços.

Como AWS HealthLake funciona

AWS HealthLake cria um armazenamento de dados que armazena registros de saúde utilizando a especificação Healthcare Interoperability FHIR (R4). Com HealthLake, você pode realizar as seguintes tarefas.

> **Note**
>
> Depois de 20 de fevereiro de 2023, HealthLake os armazenamentos de dados não usam processamento de linguagem natural integrado (NLP) por padrão. Se você estiver interessado em ativar esse recurso em seu armazenamento de dados, consulte o Como faço para ativar o recurso integrado HealthLake de processamento de linguagem natural do? capítulo Solução de problemas.

- Crie, monitore e exclua um armazenamento de dados.
- Use `StartFHIRImportJob` para importar dados de saúde em massa de um bucket do Amazon Simple Storage Service (Amazon S3) para um armazenamento de dados.
- Use as operações Criar, Ler, Atualizar e Excluir (CRUD) para gerenciar dados armazenados em seu armazenamento de dados.
- Use SQL no Amazon Athena para consultar seu armazenamento de dados.
- Use um HTTP cliente nas FHIR REST API operações para pesquisar seu armazenamento de dados.
- Permita que as API operações do Amazon Comprehend Medical pesquisem informações médicas em seus dados usando o processamento de linguagem natural (). NLP

Criação e monitoramento de armazenamentos de dados

Com HealthLake, você pode criar e monitorar armazenamentos de dados que podem armazenar dados do Fast Healthcare Interoperability Resources (FHIR).

Para criar um novo armazenamento de dados, você pode usar o C reateFHIRDatastore ou o HealthLake console. Para ver o status de um armazenamento de dados, use escribeFHIRDatastoreD. Para ver o status de vários armazenamentos de dados ativos, use istFHIRDatastoresL. Para excluir um armazenamento de dados, use eleteFHIRDatastoreD.

FHIRRESTAPIoperações

Você pode usar as FHIR REST API operações para realizar as operações Criar, Ler, Atualizar, Excluir (CRUD) no seu armazenamento de HealthLake dados. Para saber mais sobre como HealthLake dá suporte às FHIR REST API operações, consulteUsando FHIR REST API interações com um armazenamento HealthLake de dados.

Geração automatizada de recursos a partir de extensões FHIR DocumentReference de recursos

> **Note**
>
> Ao criar um armazenamento de HealthLake dados e adicionar dados que contenham o`DocumentReference`, você incorrerá em cobranças em sua AWS conta. Para obter mais detalhes, consulte AWS HealthLake os preços.

HealthLake fornece NLP em documentos encontrados no tipo `DocumentReference` de recurso. Para analisar o texto, HealthLake usa as seguintes operações do Amazon Comprehend Medical. API

- `DetectEntitiesV2`: inspeciona o texto clínico de uma variedade de entidades médicas e retorna informações específicas sobre elas, como categoria da entidade, localização e pontuação de confiança.
- `InferICD10CM`: Inspeciona o texto clínico para detectar condições médicas como entidades listadas em um prontuário de paciente e vincula essas entidades a identificadores de conceitos normalizados na base de conhecimento ICD -10-CM dos Centros de Controle de Doenças.
- `InferRxNorm`: Inspeciona o texto clínico para detectar medicamentos como entidades listadas em um prontuário de paciente e links para os identificadores de conceitos normalizados no RxNorm banco de dados da National Library of Medicine.

HealthLake analisa automaticamente os dados encontrados no tipo de `DocumentReference` recurso quando ele é adicionado ao seu armazenamento de dados. Os arquivos de `DocumentReference` recursos originais permanecem inalterados. As informações médicas extraídas são anexadas automaticamente como extensões FHIR compatíveis. Para saber mais sobre como NLP funciona em HealthLake, consulteUsando a geração automatizada de recursos com base

no processamento de linguagem natural (NLP) do tipo de FHIR DocumentReference recurso em AWS HealthLake.

Pesquise com consultas SQL baseadas

> **Note**
>
> Para armazenamentos de dados criados antes de 14 de novembro de 2022, sua pesquisa está limitada às FHIR REST API operações. Para usar consultas SQL baseadas para dados em seu armazenamento de HealthLake dados, consulteConsulte armazenamentos AWS HealthLake de dados usando SQL no Amazon Athena.

O Amazon Athena é um serviço de consulta sem servidorSQL. HealthLake os armazenamentos de dados são inseridos no Athena como tabelas Apache Iceberg. Essas tabelas foram projetadas para oferecer suporte a grandes conjuntos de dados analíticos. No Athena, cada tipo de FHIR recurso é representado como uma tabela. Usando o Athena, você só pode fazer `READ` solicitações no seu armazenamento de dados. Para saber mais sobre a pesquisa SQL baseada, consulteConsulte seu armazenamento HealthLake de dados usando SQL.

Pesquise com FHIR REST API operações

Você pode pesquisar os registros de saúde armazenados em seu armazenamento de dados especificando um tipo de recurso com parâmetros de pesquisa compatíveis ou usando uma ID de recurso encontrada no servidor, sem especificar o tipo de recurso. Para saber mais sobre como pesquisar usando as FHIR REST API operações, consulteUsando FHIR REST API interações com um armazenamento HealthLake de dados.

Ações para importação de dados

Use AWS HealthLake para importar seus arquivos em massa de um bucket do Amazon S3. Use o console ou o S tartFHIRImport Job para iniciar um trabalho de importação. Depois de importar seus arquivos, você pode usar o D escribeFHIRImport Job para monitorar o status do trabalho. Depois que o trabalho de importação for concluído, os dados poderão ser adicionados ao Athena, transformados ou analisados e usados em aplicativos downstream.

Ações para exportação de dados

Use HealthLake para exportar seus arquivos em massa para um bucket do Amazon S3. Use o console ou o S tartFHIRExport Job para iniciar um trabalho de exportação. Depois de exportar seus arquivos, você pode usar o D escribeFHIRExport Job para monitorar o status do trabalho e visualizar suas propriedades. Depois que o trabalho de exportação for concluído, você poderá visualizar os dados usando a Amazon QuickSight ou acessá-los usando outros AWS serviços.

AWS HealthLake validações FHIR de perfil suportadas

HealthLake suporta a especificação FHIR R4 básica. Incluídos na especificação R4 estão os FHIR perfis. Os perfis são usados em um tipo de FHIR recurso para definir uma definição de tipo de recurso mais específica usando restrições e/ou extensões no tipo de recurso básico. Por exemplo, um FHIR perfil pode identificar campos obrigatórios, como extensões e conjuntos de valores. Um recurso pode oferecer suporte a vários perfis. Todos os armazenamentos HealthLake de dados oferecem suporte ao uso FHIR de perfis.

Não é necessário adicionar um FHIR perfil ao adicionar dados a um armazenamento HealthLake de dados. Se nenhum FHIR perfil for especificado quando um recurso for adicionado ou atualizado, o recurso só será validado em relação ao esquema FHIR R4 básico.

FHIROs perfis aos quais um recurso está em conformidade são incluídos no recurso antes de ele ser ingerido. HealthLake HealthLake valida os FHIR perfis especificados quando são adicionados ao seu armazenamento de HealthLake dados.

FHIROs perfis são especificados em um guia de implementação. HealthLake valida os FHIR perfis definidos nos guias de implementação a seguir.

FHIRPerfis suportados por HealthLake

Nome	Versã	Guia de implementação	Recurso
Núcleo dos EUA	3.1.1	http://hl7.org/fhir/us/core/STU3.1.1/	Padrão
Núcleo dos EUA	4.0.0	https://hl7.org/fhir/us/core/STU4/index.html	Compatível
CARINBotão azul	1.1.0	http://hl7.org/fhir/us/carin-bb/STU1.1/	Padrão
CARINBotão azul	1.0.0	https://hl7.org/fhir/us/carin-bb/STU1/	Compatível
Troca de Dados do Pagador Da Vinci	1.0.0	https://hl7.org/fhir/us/davinci-pdex/	Padrão
Bolsa de Registros de Saúde Da Vinci () HRex	0.2.0	https://hl7.org/fhir/us/davinci-hrex/2020Sep/	Padrão

Nome	Versã	Guia de implementação	Recurso
DaVinci PDEXPlano Net	1.1.0	https://hl7.org/fhir/us/davinci-pdex-plan-net/STU1.1/	Padrão
DaVinci PDEXPlano Net	1.0.0	https://hl7.org/fhir/us/davinci-pdex-plan-net/STU1/	Compatível
DaVinci Payer Data Exchange (PDex) Formulário de medicamentos dos EUA	1.1.0	https://hl7.org/fhir/us/davinci-drug-formulary/STU1.1/	Padrão
DaVinci Payer Data Exchange (PDex) Formulário de medicamentos dos EUA	1.0.1	https://hl7.org/fhir/us/davinci-drug-formulary/STU1.0.1/	Compatível
Missão digital Ayushman Bharat da Autoridade Nacional de Saúde () ABDM	2,0	https://www.nrces.in/ndhm/fhir/r4/index.html	Padrão

Validando FHIR perfis especificados em um recurso

Para que um FHIR perfil seja validado, adicione-o ao `profile` elemento de recursos individuais usando o perfil URL designado no guia de implementação.

FHIROs perfis são validados quando você adiciona um novo recurso ao seu armazenamento de dados. Para adicionar um novo recurso, você pode usar a API operação S tartFHIRImport Job, fazer uma P0ST solicitação para adicionar um novo recurso ou `PUT` atualizar um recurso existente.

Example — Para ver qual FHIR perfil é referenciado em um recurso

O perfil URL é adicionado ao `profile` elemento no par de `"meta" : "profile"` valores-chave. Esse recurso foi truncado para maior clareza.

```
            {
             "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 — Como referenciar um perfil não padrão suportado FHIR

Para validar em relação a um perfil não padrão compatível (por exemplo. CarinBB 1.0.0) - adicione o perfil URL com a versão (separado por '|') e o perfil base URL no elemento. `meta.profile` Este recurso de exemplo foi truncado para maior clareza.

```
            {
             "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 dados pré-carregados

HealthLake suporta somente SYNTHEA como um tipo de dados pré-carregado. O Synthea é um gerador sintético de pacientes que modela o histórico médico de pacientes gerados por modelos. É um repositório Git de código aberto que HealthLake permite FHIR gerar pacotes de recursos compatíveis com R4 para que os usuários possam testar modelos sem usar dados reais do paciente.

Os seguintes tipos de recursos estão disponíveis em armazenamentos de dados pré-carregados.

Tipos de recursos do Synthea suportados

AllergyIntolerance	Local
CarePlan	MedicationAdministration
CareTeam	MedicationRequest
Reivindicar	Observação
Condição	Organização
Dispositivo	Paciente
DiagnosticReport	Praticante
Encontro	PractitionerRole
ExplanationofBenefit	Procedimento
ImagingStudy	Proveniência
Imunização	

Configurando permissões para começar a usar AWS HealthLake

Neste capítulo, você usa o AWS Management Console para configurar as permissões necessárias para começar a usar AWS HealthLake e criar um armazenamento de dados. Para configurar permissões para criar um armazenamento de dados, você cria um IAM usuário ou uma função que é administrador e HealthLake administrador do data lake. Você torna esse usuário um administrador de data lake no AWS Lake Formation. O administrador do data lake concede ao Lake Formation acesso aos recursos necessários para usar o Amazon Athena para consultar um armazenamento de dados.

Depois de criar um armazenamento de dados no HealthLake, você pode configurar permissões para importar arquivos para o armazenamento de dados ou exportá-los. Para obter informações sobre como configurar permissões para importar arquivos, consulteConfigurando permissões para trabalhos de importação. Para obter informações sobre como configurar permissões para exportar arquivos, consulteConfigurando permissões para trabalhos de exportação.

Tópicos

- Inscreva-se para um Conta da AWS
- Criar um usuário com acesso administrativo
- Configurar um IAM usuário ou uma função para usar HealthLake (IAMAdministrador)
- Adicionar um usuário ou função como administrador do Data Lake em Lake Formation (IAMadministrador)

Inscreva-se para um Conta da AWS

Se você não tiver um Conta da AWS, conclua as etapas a seguir para criar um.

Para se inscrever em um Conta da AWS

1. Abra a https://portal.aws.amazon.com/billing/inscrição.
2. Siga as instruções online.

   Parte do procedimento de inscrição envolve receber uma chamada telefônica e inserir um código de verificação no teclado do telefone.

Quando você se inscreve em um Conta da AWS, um Usuário raiz da conta da AWSé criado. O usuário-raiz tem acesso a todos os Serviços da AWS e recursos na conta. Como prática recomendada de segurança, atribua o acesso administrativo a um usuário e use somente o usuário-raiz para executar tarefas que exigem acesso de usuário-raiz.

AWS envia um e-mail de confirmação após a conclusão do processo de inscrição. A qualquer momento, você pode visualizar a atividade atual da sua conta e gerenciar sua conta acessando https://aws.amazon.com/e escolhendo Minha conta.

Criar um usuário com acesso administrativo

Depois de se inscrever em um Conta da AWS, proteja seu Usuário raiz da conta da AWS AWS IAM Identity Center, habilite e crie um usuário administrativo para que você não use o usuário root nas tarefas diárias.

Proteja seu Usuário raiz da conta da AWS

1. Faça login AWS Management Consolecomo proprietário da conta escolhendo Usuário raiz e inserindo seu endereço de Conta da AWS e-mail. Na próxima página, insira sua senha.

   Para obter ajuda com o login usando o usuário-raiz, consulte Fazer login como usuário-raiz no Guia do usuário do Início de Sessão da AWS .
2. Ative a autenticação multifator (MFA) para seu usuário root.

   Para obter instruções, consulte Habilitar um MFA dispositivo virtual para seu usuário Conta da AWS root (console) no Guia IAM do usuário.

Criar um usuário com acesso administrativo

1. Ative o IAM Identity Center.

   Para obter instruções, consulte Habilitar o AWS IAM Identity Center no Guia do usuário do AWS IAM Identity Center .
2. No IAM Identity Center, conceda acesso administrativo a um usuário.

Para ver um tutorial sobre como usar o Diretório do Centro de Identidade do IAM como fonte de identidade, consulte Configurar o acesso do usuário com o padrão Diretório do Centro de Identidade do IAM no Guia AWS IAM Identity Center do usuário.

Iniciar sessão como o usuário com acesso administrativo

- Para entrar com seu usuário do IAM Identity Center, use o login URL que foi enviado ao seu endereço de e-mail quando você criou o usuário do IAM Identity Center.

  Para obter ajuda para fazer login usando um usuário do IAM Identity Center, consulte Como fazer login no portal de AWS acesso no Guia Início de Sessão da AWS do usuário.

Atribuir acesso a usuários adicionais

1. No IAM Identity Center, crie um conjunto de permissões que siga as melhores práticas de aplicação de permissões com privilégios mínimos.

   Para obter instruções, consulte Criar um conjunto de permissões no Guia do usuário do AWS IAM Identity Center .
2. Atribua usuários a um grupo e, em seguida, atribua o acesso de autenticação única ao grupo.

   Para obter instruções, consulte Adicionar grupos no Guia do usuário do AWS IAM Identity Center .

Configurar um IAM usuário ou uma função para usar HealthLake (IAMAdministrador)

> Pessoa: Administrador IAM
>
> Um usuário que pode criar IAM usuários e funções e adicionar administradores de data lake.

Essas etapas neste tópico devem ser executadas por um IAM administrador.

Para conectar seu armazenamento de HealthLake dados ao Athena, você precisa criar um IAM usuário ou uma função que seja administrador e administrador do data lake. HealthLake Esse novo

usuário ou função concede acesso aos recursos encontrados em um armazenamento de dados por meio do AWS Lake Formation e tem a política `AmazonHealthLakeFullAccess` AWS gerenciada adicionada ao usuário ou função.

> **Important**
>
> Um IAM usuário ou função que seja administrador de data lake não pode criar novos administradores de data lake. Para adicionar mais administrador de data lake, você deve usar um IAM usuário ou uma função que tenha recebido `AdministratorAccess` acesso.

Para criar um administrador

1. Adicione a política **`AmazonHealthlakeFullAccess`** IAM AWS gerenciada a um usuário ou função na sua organização.

   Se você não estiver familiarizado com a criação de um IAM usuário, consulte Criação de um IAM usuário e Visão geral das AWS IAM políticas no Guia do IAM usuário.
2. Conceda ao IAM usuário ou à função acesso ao AWS Lake Formation.

   - Adicione a seguinte política IAM AWS gerenciada a um usuário ou função em sua organização: **`AWSLakeFormationDataAdmin`**

     > **Note**
     >
     > A `AWSLakeFormationDataAdmin` política concede acesso a todos os recursos AWS do Lake Formation. Recomendamos que você sempre use as permissões mínimas necessárias para realizar sua tarefa. Para obter mais informações, consulte as IAM melhores práticas no Guia IAM do usuário.
3. Adicione a seguinte política embutida ao usuário ou à função. Para obter mais informações, consulte Políticas em linha no Guia do IAM usuário.

```
{
    "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 obter mais informações sobre a `AWSLakeFormationDataAdmin` política, consulte a Referência de IAM Permissões e Personalidades do Lake Formation no Guia do Desenvolvedor do AWS Lake Formation.

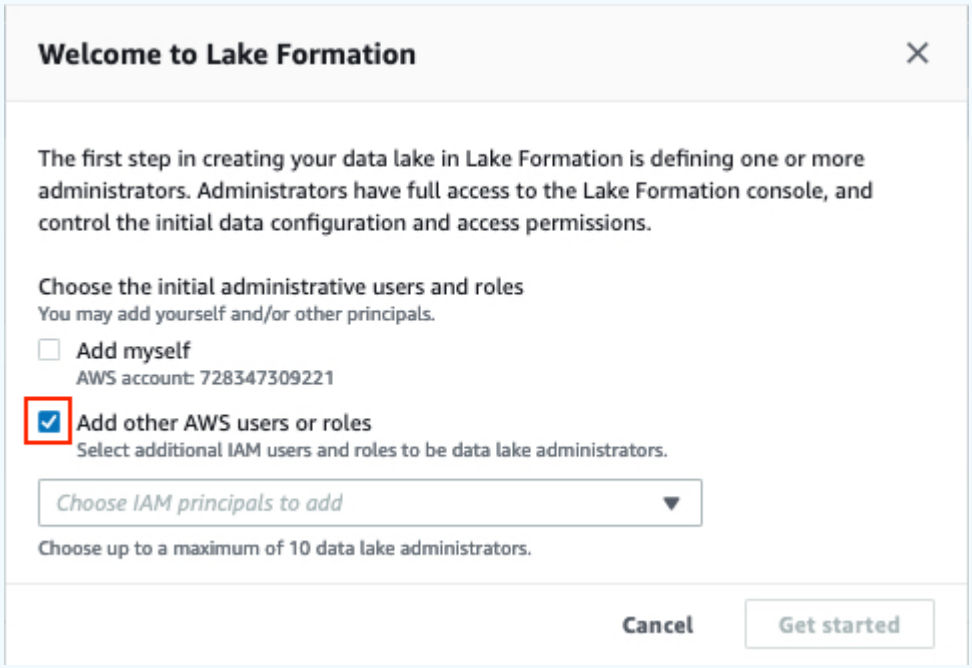
Adicionar um usuário ou função como administrador do Data Lake em Lake Formation (IAMadministrador)

Em seguida, o IAM administrador precisa adicionar o usuário ou a função criada na etapa 1 como administrador do data lake no Lake Formation.

Para adicionar um IAM usuário ou uma função como administrador do data lake

1. Abra o console do AWS Lake Formation: https://console.aws.amazon.com/lakeformation/

> **Note**
>
> Se esta é a primeira vez que você visita Lake Formation, uma caixa de diálogo Welcome to Lake Formation é exibida solicitando que você defina um administrador do Lake Formation.



2. Atribua ao novo usuário ou função um administrador de data lake do AWS Lake Formation.
   - Opção 1: Se você recebeu a caixa de diálogo Welcome to Lake Formation.
     1. Escolha Adicionar outros AWS usuários ou funções.
     2. Escolha a seta para baixo (▼).
     3. Escolha o HealthLake administrador que você gostaria que também fosse administrador do Lake Formation.
     4. Escolha Começar.
   - Opção 2: Use o painel de navegação (☰).
     1. Escolha o painel de navegação (☰).
     2. Em Permissões, escolha Funções e tarefas administrativas.
     3. Na seção Administradores do Data Lake, selecione Escolher administradores.
     4. Na caixa de diálogo Gerenciar administradores do data lake, escolha a seta para baixo (▼).
     5. Em seguida, selecione ou pesquise os HealthLake administradores, usuários ou funções que você também deseja que sejam administradores do Lake Formation.
     6. Escolha Salvar.
3. Altere as configurações de segurança padrão a serem gerenciadas pelo Lake Formation. Os recursos do armazenamento de HealthLake dados não precisam ser gerenciados pelo Lake

FormationIAM. Para atualizar, consulte Alterar o modelo de permissão padrão no AWS Lake Formation Developer Guide.

Criando um armazenamento de dados em AWS HealthLake

Depois de concluirConfigurando permissões para começar a usar AWS HealthLake, você estará pronto para criar um armazenamento de dados. Em AWS HealthLake, você usa um armazenamento de dados para armazenar dados no formato HL7 FHIR (R4). Os tópicos deste capítulo descrevem como criar um armazenamento de dados.

Para criar armazenamentos de dados habilitados para análise e conceder acesso a eles no Athena, adicione a política `AWSLakeFormationDataAdmin` gerenciada ao seu IAM usuário, grupo ou função. A `AWSLakeFormationDataAdmin` política permite que você crie administradores de data lake e conceda acesso aos armazenamentos de dados no Athena. Para obter informações sobre a configuração de permissões, consulteConfigurando permissões para começar a usar AWS HealthLake.

HealthLake também está integrado com AWS CloudTrail. Você pode usar CloudTrail para fornecer um registro das ações realizadas por um usuário, função ou AWS serviço em HealthLake. CloudTrail captura todas as API chamadas e ações do console HealthLake como eventos. Para saber mais, consulte Registro em log de chamadas do AWS HealthLake API com o AWS CloudTrail.

Para saber mais sobre os tipos de recursos Fast Healthcare Interoperability Resources (FHIR) que são suportados pelo HealthLake, consulte. Tipos FHIR de recursos compatíveis em AWS HealthLake

> ⚠ Compatibilidade com Amazon Athena
>
> HealthLake lojas de datas criadas antes de 14 de novembro de 2022 não podem realizar SQL consultas usando o Athena. Para usar os recursos de pesquisa do Athena em seu armazenamento de dados preexistente, primeiro migre os dados para um novo armazenamento de dados. Para saber mais sobre a migração de armazenamentos de dados preexistentes, consulte. Migração de um armazenamento de dados existente para usar o Amazon Athena

Depois de criar um armazenamento de dados, você pode obter suas propriedades, incluindo seu status, com as API operações API_DescribeFHIRDatastoreou API_ListFHIRDatastores.html. Ou você pode encontrar os status do armazenamento de dados e outros detalhes na página Armazenamentos de dados no HealthLake console.

Um armazenamento HealthLake de dados pode ter os seguintes status:

- Criação — Seu armazenamento de dados está sendo criado.
- Ativo — Seu armazenamento de dados está ativo. Você pode importar e exportar dados dele. Você também pode gerenciar e pesquisar os FHIR recursos armazenados no armazenamento de dados.
- Excluindo — Seu armazenamento de dados está sendo excluído.
- Excluído — Seu armazenamento de dados foi excluído.

Tópicos

Criando um armazenamento de dados (AWS Management Console)

> **HealthLake diferenças de console**
>
> O HealthLake console não oferece suporte à criação de um SMART armazenamento de dados FHIR ativado. Para criar um armazenamento SMART de dados FHIR ativado, você deve usar o AWS CLI ou um dos AWS suportadosSDKS. Para saber mais, consulte Integrando SMART com FHIR AWS HealthLake. Além disso, o console não diferencia entre os dois tipos de armazenamento de dados suportados HealthLake quando você visualiza a página de detalhes de um armazenamento de dados individual.

Para criar um armazenamento HealthLake de dados

1. Abra o HealthLake console em https://console.aws.amazon.com//healthlake/casa.
2. Abra o painel de navegação (≡).
3. Em seguida, escolha Armazenamentos de dados.
4. Em seguida, escolha Criar armazenamento de dados.
5. Na seção Configurações do Armazenamento de Dados, em Nome do Armazenamento de Dados, especifique um nome.
6. (Opcional) Na seção Configurações do Data Store, em Pré-carregar dados de amostra, marque a caixa de seleção para pré-carregar dados do Synthea.

- Os dados Synthea são um conjunto de dados de amostra pré-carregado. Para obter mais informações, consulte Tipos de dados pré-carregados.

7. Na seção Criptografia do Data Store, escolha Usar AWS chave própria (padrão) ou Escolher uma AWS KMS chave diferente (avançada).
8. Na seção Tags - opcional, você pode adicionar tags ao seu armazenamento de dados.

   - Para saber mais sobre como marcar seu armazenamento de dados, consulteAdicionando uma tag a um armazenamento de dados.

9. Em seguida, escolha Criar armazenamento de dados. O status dos seus armazenamentos de dados está disponível na página Armazenamentos de dados.

Criando um armazenamento de dados (AWS CLI e AWS SDKs)

Você pode usar os exemplos de código a seguir para criar um armazenamento HealthLake de dados.

AWS CLI

O exemplo a seguir demonstra como usar a operação `CreateFHIRDatastore` com o AWS CLI. Para executar o exemplo, é necessário instalar a AWS CLI. Quando você cria seu armazenamento de dados, a criptografia em repouso usa como padrão uma KMS chave AWS própria, a menos que especificado de outra forma. Para saber mais sobre criptografia, HealthLake consulte,Criptografia em REST para AWS HealthLake. REST

O exemplo é formatado para Unix, Linux e macOS. Para Windows, substitua o caractere de continuação Unix de barra invertida (\) no final de cada linha por um acento circunflexo (). ^

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

Em caso de sucesso, você receberá a seguinte JSON resposta. Quando seu armazenamento de dados estiver pronto para ingerir dados, o status mudará para`ACTIVE`. Para saber mais sobre a importação de dados para seu armazenamento HealthLake de dados, consulteImportação de arquivos para um armazenamento de HealthLake dados.

```
{
    "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 uma lista de todos os armazenamentos de dados/armazenamentos de dados, você pode usar a `ListFHIRDataStore` operação. Você também pode ver uma lista de armazenamentos de dados ativos no HealthLake console.

Python (boto3)

O exemplo a seguir demonstra como criar um armazenamento de HealthLake dados usando a `create_fhir_datastore` operação. Quando você cria seu armazenamento de dados, a criptografia em repouso assume como padrão uma AWS KMS chave de AWS propriedade, a menos que especificado de outra forma. Para saber mais sobre criptografia, HealthLake consulte,Criptografia em REST para AWS HealthLake. REST

```
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')
```

Um armazenamento de dados pode ter um dos quatro status. Use `list_fhir_datastores` para visualizar uma lista dos seus armazenamentos de HealthLake dados, independentemente do status. Este exemplo mostra como você pode filtrar com base no status de um armazenamento de dados.

```
import boto3

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

Para saber mais, consulte a `list_fhir_datastore`documentação do Boto3.

Importação de arquivos para um armazenamento de HealthLake dados

Depois de concluirCriando um armazenamento de dados em AWS HealthLake, você pode importar arquivos para o armazenamento de dados a partir de um bucket do Amazon Simple Storage Service (Amazon S3). Para importar arquivos, você inicia um trabalho de importação com o HealthLake console ou a `StartFHIRImportJob` API operação.

Ao criar um trabalho de importação, você especifica a localização dos seus dados de entrada no Amazon S3, um local de bucket do Amazon S3 para arquivos de log de saída, IAM uma função que HealthLake concede acesso aos seus buckets e uma chave de propriedade ou propriedade do cliente. AWS AWS Key Management Service HealthLake usa essa chave para criptografar seus dados no local de origem e será usada para descriptografá-los para permitir HealthLake a importação. Para obter informações sobre como configurar permissões para trabalhos de importação, consulteConfigurando permissões para trabalhos de importação. Para saber mais sobre como criar e usar AWS KMS chaves, consulte Criação de chaves no Guia do desenvolvedor do AWS Key Management Service.

HealthLake aceita arquivos de entrada no formato delimitado por nova linha JSON (`.ndjson`), em que cada linha consiste em um recurso válido. FHIR Você pode usar as API operações `DescribeFHIRImportJob` e `ListFHIRImportJobs` para descrever e listar os trabalhos de importação em andamento.

Para cada tarefa de importação, HealthLake gera um `manifest.json` arquivo. Esse registro descreve tanto os sucessos quanto os fracassos de um trabalho de importação. HealthLake envia o arquivo para o bucket do Amazon S3 que você especifica ao criar um trabalho de importação. Para obter mais informações, consulte JSONArquivo de manifesto.

Você pode enfileirar trabalhos de importação ou exportação. Esses trabalhos assíncronos de importação ou exportação são processados de forma FIFO (primeiro a entrar, primeiro a sair). Você pode criar, ler, atualizar ou excluir FHIR recursos enquanto um trabalho de importação ou exportação está em andamento.

Depois de preencher um datastore com dados pré-carregados ou importar dados, você pode começar a consultar seu datastore usando o Amazon SQL Athena. Para obter mais informações, consulte Consulte armazenamentos AWS HealthLake de dados usando SQL no Amazon Athena.

Tópicos

Configurando permissões para trabalhos de importação

Antes de importar arquivos para um armazenamento de dados, você deve conceder HealthLake permissão para acessar seus buckets de entrada e saída no Amazon S3. Para conceder HealthLake acesso, você cria uma função de IAM serviço para HealthLake, adiciona uma política de confiança à função para conceder permissões de HealthLake assumir função e anexa uma política de permissões à função que concede acesso aos seus buckets do Amazon S3.

Ao criar um trabalho de importação, você especifica o Amazon Resource Name (ARN) dessa função para `DataAccessRoleArn` o. Para obter mais informações sobre IAM funções e políticas de confiança, consulte IAMFunções.

Depois de configurar a permissão, você estará pronto para importar arquivos para o seu armazenamento de dados com uma tarefa de importação. Para obter mais informações, consulte Iniciando um trabalho de importação em HealthLake.

Para configurar permissões de importação

1. Caso ainda não tenha feito isso, crie um bucket Amazon S3 de destino para os arquivos de log de saída. O bucket do Amazon S3 deve estar na mesma AWS região do serviço, e o Block Public Access deve estar ativado para todas as opções. Para saber mais, consulte Como usar o Amazon S3 para bloquear o acesso público. Uma KMS chave de propriedade da Amazon ou do cliente também deve ser usada para criptografia. Para saber mais sobre o uso de KMS chaves, consulte Amazon Key Management Service.
2. Crie uma função de serviço de acesso a dados HealthLake e dê permissão ao HealthLake serviço para assumi-la com a seguinte política de confiança. HealthLake usa isso para gravar o bucket de saída do Amazon S3.

```
{
    "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. Adicione uma política de permissões à função de acesso a dados que permita que ela acesse o bucket do Amazon S3. `amzn-s3-demo-bucket`Substitua pelo nome do seu 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"
    }]
}
```

Iniciando um trabalho de importação em HealthLake

Depois de criar um armazenamento de dados e configurar as permissões para trabalhos de importação (Configurando permissões para trabalhos de importação), você pode começar a importar arquivos com um trabalho de importação. Você pode iniciar um trabalho de importação usando o AWS HealthLake console ou a AWS HealthLake importaçãoAPI, start-fhir-import-jobAPI.

Tópicos

- Importação de arquivos com operações API
- Iniciando um trabalho de importação (console)

Importação de arquivos com operações API

Pré-requisitos

Ao usar as AWS HealthLake API operações, você deve primeiro criar uma política AWS Identity and Access Management (IAM) e anexá-la a uma IAM função. Para saber mais sobre IAM funções e políticas de confiança, consulte IAMPolíticas e permissões. Os clientes também devem usar uma KMS chave para criptografia. Para saber mais sobre o uso de KMS chaves, consulte Amazon Key Management Service.

Para importar arquivos (API), use as etapas a seguir.

1. Carregue seus dados em um bucket do Amazon S3.
2. Use a `start-fhir-import-job API` API operação. Ao iniciar o trabalho, especifique o nome do bucket do Amazon S3 que contém os arquivos de entrada, a KMS chave que você deseja usar para criptografia e a configuração dos dados de saída.

3. Para saber mais sobre uma tarefa de FHIR importação, use a describe-fhir-import-joboperação para obter a ID, o nome, ARN a hora de início, a hora de término e o status atual da tarefa. Use list-fhir-import-jobpara mostrar todos os trabalhos de importação e seus status.

Iniciando um trabalho de importação (console)

Para importar arquivos com o console, você carrega seus dados em um bucket do Amazon S3,

Para importar arquivos, use as etapas a seguir.

1. Carregue seus dados em um bucket do Amazon S3.
2. Abra o HealthLake console em https://console.aws.amazon.com//healthlake/casa.
3. Acesse a página de detalhes do armazenamento de dados do seu armazenamento de dados e escolha Importar.
4. Especifique seu bucket do Amazon S3 e crie ou identifique a IAM função e a KMS chave que você deseja usar.
5. Escolha Importar dados.

JSONArquivo de manifesto

Para cada tarefa de importação, HealthLake gera um `manifest.json` arquivo. HealthLake envia o arquivo para o bucket do Amazon S3 que você especifica ao criar um trabalho de importação.

O `manifest.json` arquivo descreve tanto os sucessos quanto os fracassos de um trabalho de importação. Os arquivos de log são organizados em duas pastas, chamadas `SUCCESS` `FAILURE` e. Um arquivo de saída pode conter informações confidenciais, portanto, ao criar um trabalho de importação, você deve fornecer um bucket Amazon S3 de saída e uma AWS KMS chave para criptografia.

Veja a seguir um exemplo do `manifest.json` arquivo de saída. Recomendamos que você use esse arquivo como a primeira etapa para solucionar uma falha na tarefa de importação. Ele fornece detalhes sobre cada arquivo e o que causou a falha na tarefa de importação.

```
{
"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
}
```

Exemplo: Iniciando e monitorando trabalhos de importação com o AWS CLI

O exemplo a seguir mostra como usar o AWS Command Line Interface para iniciar e monitorar um trabalho de importação. Você também pode usar o 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
```

Quando o trabalho de importação começar, você receberá a seguinte confirmação.

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

Para monitorar o status de uma tarefa de importação ou conhecer suas propriedades de configuração, use o AWS CLI comando describe-fhir-import-jobAPIor the, conforme mostrado no exemplo a seguir.

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

Você recebe as seguintes informações em resposta.

```
{
    "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 uma lista de todos os trabalhos de importação, use o AWS CLI comando list-fhir-import-jobsAPIou o, conforme mostrado no exemplo a seguir. Você pode adicionar um ou mais filtros para limitar os 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)
```

Você recebe as seguintes informações em resposta.

```
{
    "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
```

Exportação de arquivos de um armazenamento de HealthLake dados

Depois de criar um armazenamento de dados e importar dados (ou se você usar dados de amostra pré-carregados), você pode exportar os dados para um bucket do Amazon S3. Para exportar dados do seu armazenamento de HealthLake dados, use as operações a seguir.

- Faça uma solicitação de exportação usando a `StartFHIRExportJob` API operação usando AWS SDKs HealthLake e.
  - Essa operação só oferece suporte para fazer uma solicitação de exportação em todo o sistema.
- Faça uma solicitação de exportação usando a `export` sintaxe usando o. HealthLake FHIR REST API
  - Essa operação suporta a realização de solicitações de exportação para todo o sistema, pacientes e grupos. Você também pode aplicar parâmetros para filtrar ainda mais os dados na solicitação de exportação.

> **Important**
>
> HealthLake SDKas solicitações de exportação usando `StartFHIRExportJob` API a operação e as solicitações de FHIR REST API exportação usando `StartFHIRExportJobWithPost` API a operação têm IAM ações separadas. Cada IAM ação, SDK exportar com `StartFHIRExportJob` e FHIR REST API exportar com`StartFHIRExportJobWithPost`, pode ter permissões de permitir/negar tratadas separadamente. Se você quiser que ambas SDK e FHIR REST API as exportações sejam restritas, não se esqueça de negar as permissões para cada IAM ação.

Ambas as operações suportam apenas a exportação de seus arquivos para um bucket Amazon S3 (S3). Todos os arquivos do seu armazenamento de HealthLake dados são exportados como arquivos delimitados por nova linha JSON (`.ndjson`), onde cada linha consiste em um recurso válido. FHIR

Ambas as operações exigem uma função de serviço. Nele, HealthLake deve ser definido como o principal do serviço, e você deve definir um bucket do Amazon Simple Storage Service (S3) de

onde você deseja exportar seus arquivos. Para saber mais, consulte Configurando permissões para trabalhos de exportação.

Você pode enfileirar trabalhos de importação ou exportação. Esses trabalhos assíncronos de importação ou exportação são processados de forma FIFO (primeiro a entrar, primeiro a sair). Você pode criar, ler, atualizar ou excluir FHIR recursos enquanto um trabalho de importação ou exportação está em andamento.

Para exportar arquivos do seu armazenamento de HealthLake dados, consulte as seções a seguir.

- Configurando permissões para trabalhos de exportação
- Exportar arquivos do seu armazenamento de dados com o HealthLake console ou AWS SDKs
- Exportação de dados do seu armazenamento HealthLake de dados com operações FHIR REST API

Configurando permissões para trabalhos de exportação

Antes de exportar arquivos de um armazenamento de dados, você deve conceder HealthLake permissão para acessar seu bucket de saída no Amazon S3. Para conceder HealthLake acesso, você cria uma função de IAM serviço para HealthLake, adiciona uma política de confiança à função para conceder permissões de HealthLake assumir função e anexa uma política de permissões à função que concede acesso ao seu bucket do Amazon S3.

Se você já criou uma função para HealthLake inConfigurando permissões para trabalhos de importação, você pode reutilizá-la e conceder a ela as permissões adicionais para seu bucket de exportação do Amazon S3 listado neste tópico. Para saber mais sobre IAM funções e políticas de confiança, consulte IAMPolíticas e permissões.

> **Important**
>
> HealthLake SDKas solicitações de exportação usando `StartFHIRExportJob` API a operação e as solicitações de FHIR REST API exportação usando `StartFHIRExportJobWithPost` API a operação têm IAM ações separadas. Cada IAM ação, SDK exportar com `StartFHIRExportJob` e FHIR REST API exportar com`StartFHIRExportJobWithPost`, pode ter permissões de permitir/negar tratadas separadamente. Se você quiser que ambas SDK e FHIR REST API as exportações sejam restritas, não se esqueça de negar as permissões para cada IAM ação. Se você conceder

acesso total aos usuários HealthLake, nenhuma alteração nas permissões IAM do usuário será necessária.

O usuário ou função que configura as permissões deve ter permissão para criar funções, criar políticas e anexar políticas às funções. A IAM política a seguir concede essas permissões.

```
{
  "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 permissões de exportação

1. Caso ainda não tenha feito isso, crie um bucket Amazon S3 de destino para os dados que você exportará do seu armazenamento de dados. O bucket do Amazon S3 deve estar na mesma AWS região do serviço, e o Block Public Access deve estar ativado para todas as opções. Para saber mais, consulte Como usar o Amazon S3 para bloquear o acesso público. Uma KMS chave de propriedade da Amazon ou do cliente também deve ser usada para criptografia. Para saber mais sobre o uso de KMS chaves, consulte Amazon Key Management Service.
2. Se você ainda não o fez, crie uma função de serviço de acesso a dados HealthLake e dê permissão ao HealthLake serviço para assumi-la com a seguinte política de confiança. HealthLake usa isso para gravar o bucket de saída do Amazon S3. Se você já criou umConfigurando permissões para trabalhos de importação, pode reutilizá-lo e conceder permissões para seu bucket do Amazon S3 na próxima etapa.

   ```
   {
   ```

```
    "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. Adicione uma política de permissões à função de acesso a dados que permita que ela acesse seu bucket de saída do Amazon S3. `amzn-s3-demo-bucket`Substitua pelo nome do seu 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"
    }]
}
```

Exportar arquivos do seu armazenamento de dados com o HealthLake console ou AWS SDKs

Depois de concluirConfigurando permissões para trabalhos de exportação, você pode exportar arquivos do seu armazenamento de dados para um bucket do Amazon Simple Storage Service (Amazon S3). Para exportar arquivos de um armazenamento de dados, você inicia um trabalho de exportação em HealthLake. Um trabalho de exportação exporta arquivos do seu armazenamento de dados no formato delimitado por nova linha JSON (`.ndjson`), em que cada linha consiste em um recurso válido. FHIR Ao iniciar um trabalho de exportação, você deve especificar uma AWS KMS chave para criptografia. Para saber mais sobre como criar uma KMS chave, consulte Criação de chaves no Guia do desenvolvedor do AWS Key Management Service.

Os tópicos a seguir abordam como iniciar um trabalho de exportação com o AWS HealthLake console e AWS SDKs com a start-fhir-export-jobAPIoperação.

Tópicos

- Exportando arquivos do seu armazenamento de dados (console)
- Exportando arquivos do seu armazenamento de dados ()AWS SDKs

Exportando arquivos do seu armazenamento de dados (console)

Para exportar arquivos (console), use as etapas a seguir.

1. Crie um bucket S3 de saída na mesma HealthLake região de.
2. Para iniciar um novo trabalho de exportação, identifique o bucket de saída do Amazon S3 e crie ou identifique a IAM função que você deseja usar. Para saber mais sobre IAM funções e políticas de confiança, consulte IAMfunções. Use também uma criptografia de KMS chave. Para saber mais sobre o uso de KMS chaves, consulte Amazon Key Management Service.
3. Para ver o status do seu trabalho de exportação, use `ListFHIRExportJobs` API a operação.

Exportando arquivos do seu armazenamento de dados ()AWS SDKs

Para exportar arquivos do seu armazenamento de dados com o AWS SDKs, use a start-fhir-export-joboperação. O código a seguir mostra como iniciar um trabalho de exportação com o SDK for Python (Boto3).

```
import boto3

client = boto3.client('healthlake')

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

Para obter a ID, nomeARN, hora de início, hora de término e status atual de um trabalho de FHIR exportação, use describe-fhir-export-job. Use list-fhir-export-jobspara listar todos os trabalhos de exportação e seus status.

O código a seguir mostra como obter as propriedades de um trabalho de exportação específico com o SDK for Python (Boto3).

```
import boto3

client = boto3.client('healthlake')

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

Exportação de dados do seu armazenamento HealthLake de dados com operações FHIR REST API

Depois de concluirConfigurando permissões para trabalhos de exportação, você pode exportar dados do seu armazenamento de HealthLake dados com FHIR REST API operações. Para fazer uma solicitação de exportação usando o FHIR RESTAPI, você deve ter um IAM usuário, um grupo ou uma função com as permissões necessárias, especificar `$export` como parte da `POST` solicitação e incluir parâmetros de solicitação no corpo da solicitação. De acordo com a FHIR especificação, o FHIR servidor deve oferecer suporte `GET` às solicitações e pode oferecer suporte `POST` às solicitações. Para oferecer suporte a parâmetros adicionais, é necessário um corpo para iniciar a exportação e, portanto, HealthLake oferece suporte POST às solicitações.

> **Important**
>
> HealthLake os armazenamentos de dados criados antes de 1º de junho de 2023 oferecem suporte somente a solicitações de trabalho de exportação FHIR REST API baseadas em exportações em todo o sistema.
>
> HealthLake os armazenamentos de dados criados antes de 1º de junho de 2023 não oferecem suporte para obter o status de uma exportação usando uma `GET` solicitação no endpoint de um armazenamento de dados.

Todas as solicitações de exportação que você faz usando o FHIR REST API são devolvidas em `ndjson` formato e exportadas para um bucket do Amazon S3. Cada objeto do S3 conterá somente um único tipo FHIR de recurso.

Você pode enfileirar solicitações de exportação de acordo com as AWS cotas da conta. Para saber mais sobre as Cotas de Serviço associadas a HealthLake, consulte. AWS HealthLake endpoints e cotas

HealthLake suporta os três tipos de solicitações de endpoint de exportação em massa a seguir.

Tipo	Descrições	Sintaxe
Exportação do sistema	Exporte todos os dados do HealthLake FHIR servidor.	`POST https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/$export`
Todos os pacientes	Exporte todos os dados relacionados a todos os pacientes, incluindo os tipos de recursos associados ao tipo de recurso do paciente.	`POST https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/Patient/$export`
Grupo de pacientes	Exporte todos os dados relacionados a um grupo de pacientes especificado com uma ID de grupo.	`POST https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/Group/ID/$export`

Antes de começar

Atenda aos requisitos a seguir para fazer uma solicitação de exportação usando o FHIR REST API for HealthLake.

- Você deve ter configurado um usuário, grupo ou função que tenha as permissões necessárias para fazer a solicitação de exportação. Para saber mais, consulte Autorizando uma solicitação `export`.
- Você deve ter criado uma função de serviço que conceda HealthLake acesso ao bucket do Amazon S3 para o qual você deseja que seus dados sejam exportados. A função de serviço também deve ser especificada HealthLake como principal do serviço. Para obter mais informações sobre a configuração de permissões, consulteConfigurando permissões para trabalhos de exportação.

Autorizando uma solicitação **`export`**

Para fazer uma solicitação de exportação bem-sucedida usando FHIR REST API o., autorize seu usuário, grupo ou função usando um IAM ou OAuth2 .0. Você também deve ter uma função de serviço.

Autorizando uma solicitação usando IAM

Quando você faz uma `$export` solicitação, o usuário, o grupo ou a função devem ter `StartFHIRExportJobWithPost` e `CancelFHIRExportJobWithDelete` IAM as ações incluídas na política. `DescribeFHIRExportJobWithGet`

> **Important**
>
> HealthLake SDKas solicitações de exportação usando `StartFHIRExportJob` API a operação e as solicitações de FHIR REST API exportação usando `StartFHIRExportJobWithPost` API a operação têm IAM ações separadas. Cada IAM ação, SDK exportar com `StartFHIRExportJob` e FHIR REST API exportar com`StartFHIRExportJobWithPost`, pode ter permissões de permitir/negar tratadas separadamente. Se você quiser que ambas SDK e FHIR REST API as exportações sejam restritas, não se esqueça de negar as permissões para cada IAM ação.

Autorizando uma solicitação usando SMART on FHIR (OAuth2.0)

Ao fazer uma `$export` solicitação SMART em um armazenamento de HealthLake dados FHIR habilitado, você precisa ter os escopos apropriados atribuídos. Para saber mais sobre os escopos compatíveis, consulte[HealthLake escopos específicos de FHIR recursos de armazenamento de dados](#).

Fazendo uma **`export`** solicitação

Esta seção descreve as etapas necessárias que você deve seguir ao fazer uma solicitação de exportação usando FHIR REST API o.

Para evitar cobranças acidentais em sua AWS conta, recomendamos testar suas solicitações fazendo uma `POST` solicitação sem fornecer a `export` sintaxe.

Para fazer a solicitação, você deve fazer o seguinte:

1. Especifique `export` na `POST` solicitação URL de um endpoint compatível.
2. Especifique os parâmetros de cabeçalho necessários.
3. Especifique um corpo de solicitação que defina os parâmetros necessários.

Etapa 1: especificar **`export`** na **`POST`** solicitação URL um endpoint compatível

HealthLake oferece suporte a três tipos de solicitações de endpoint de exportação em massa. Para fazer uma solicitação de exportação em massa, você deve fazer uma solicitação `POST` com base em um dos três endpoints compatíveis. Os exemplos a seguir demonstram como especificar `export` na solicitaçãoURL.

- `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`

Nessa string de `POST` solicitação, você pode usar os seguintes parâmetros de pesquisa compatíveis.

Parâmetros de pesquisa compatíveis

HealthLake suporta os seguintes modificadores de pesquisa em solicitações de exportação em massa.

Esses exemplos incluem caracteres especiais que devem ser codificados antes de enviar sua solicitação.

Nome	Obrigatório?	Descrição	Exemplo
`_outputFormat`	Não	O formato dos arquivos de dados em massa solicitados a serem gerados. Os valores aceitos sãoapplication/fhir+n	

Nome	Obrigatório?	Descrição	Exemplo
		djson ,application/ndjson ,ndjson.	
_type	Não	Uma sequência de tipos de FHIR recursos delimitados por vírgula que você deseja incluir em seu trabalho de exportação. Recomendamos incluir _type porque isso pode ter uma implicação de custo quando todos os recursos são exportados.	&_type=MedicationStatement,Observation
_since	Não	Tipos de recursos modificados em ou após o carimbo de data e hora. Se um tipo de recurso não tiver um horário de última atualização, ele será incluído na sua resposta.	&_since=2024-05-09T00%3A00%3A00Z

Etapa 2: especificar os parâmetros de cabeçalho necessários

Para fazer uma solicitação de exportação usando o FHIR RESTAPI, você deve especificar os dois parâmetros de cabeçalho a seguir.

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

Em seguida, você deve especificar os elementos necessários no corpo da solicitação.

Etapa 3: especifique um corpo de solicitação que defina os parâmetros necessários.

A solicitação de exportação também exige um corpo em `JSON` formato. O corpo pode incluir os seguintes parâmetros.

Chave	Obrigatório?	Descrição	Valor
DataAccessRoleArn	Sim	E ARN de uma função HealthLake de serviço. A função de serviço usada deve ser especific ada HealthLake como principal do serviço.	`arn:aws:i am:: 444455556 666 :role/your- healthlake-se rvice-role`
JobName	Não	O nome da solicitação de exportação.	**your-export- job-name**
S3Uri	Sim	Parte de uma OutputDataConfig chave. O S3 URI do bucket de destino em que seus dados exportados serão baixados.	`s3://DOC- EXAMPLE-D ESTINATION- BUCKET/` **EXPORT- JOB /**
KmsKeyId	Sim	Parte de uma OutputDataConfig chave. A ARN AWS KMS chave usada para proteger o bucket do Amazon S3.	`arn:aws:k ms:` **region-of- bucket:123456789 012** `:key/`**1234abcd- 12ab-34cd -56ef-123 4567890ab**

Example — Corpo de uma solicitação de exportação feita usando o FHIR REST API

Para fazer uma solicitação de exportação usando o FHIR RESTAPI, você deve especificar um corpo, conforme mostrado a seguir.

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

Quando sua solicitação for bem-sucedida, você receberá a seguinte resposta.

cabeçalho de resposta

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

Corpo de resposta

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

Gerenciando sua solicitação de exportação

Depois de fazer uma solicitação de exportação bem-sucedida, você pode gerenciar essa solicitação usando `export` para descrever o status de uma solicitação de exportação atual e `export` cancelar uma solicitação de exportação atual.

Ao cancelar uma solicitação de exportação usando o RESTAPI, você só será cobrado pela parte dos dados que foram exportados até o momento em que você enviou a solicitação de cancelamento.

Os tópicos a seguir descrevem como você pode obter o status de uma solicitação de exportação atual ou cancelá-la.

Cancelamento de uma solicitação de exportação

Para cancelar uma solicitação de exportação, faça uma `DELETE` solicitação e forneça o ID do trabalho na solicitaçãoURL.

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

Quando sua solicitação for bem-sucedida, você receberá o seguinte.

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

Quando sua solicitação não for bem-sucedida, você receberá o seguinte.

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

Descrever uma solicitação de exportação

Para obter o status de uma solicitação de exportação, faça uma `GET` solicitação usando `export` e seu**`export-request-job-id`**.

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

A JSON resposta conterá um `ExportJobProperties` objeto. Ele pode conter os seguintes pares chave-valor.

Nome	Obrigatório?	Descrição	Valor
DataAccessRoleArn	Não	E ARN de uma função HealthLake de serviço. A função de serviço usada deve ser especific ada HealthLake como principal do serviço.	`arn:aws:iam:: 444455556666 :role/your-healthlake-service-role`
SubmitTime	Não	A data e hora em que um trabalho de exportação foi enviado.	`Apr 21, 2023 5:58:02`
EndTime	Não	A hora em que um trabalho de exportaçã o foi concluído.	`Apr 21, 2023 6:00:08 PM`
JobName	Não	O nome da solicitação de exportação.	**`your-export-job-name`**
JobStatus	Não		Os valores válidos são: `SUBMITTED | IN_PROGRESS | COMPLETED_WITH_ERRORS | COMPLETED | FAILED`
S3Uri	Sim	Parte de um OutputDataConfigobjeto. O Amazon S3 URI do	`s3://DOC-EXAMPLE-DESTINATION-`

Nome	Obrigatório?	Descrição	Valor
		bucket de destino em que seus dados exportados serão baixados.	BUCKET/ **EXPORT-JOB** /
KmsKeyId	Sim	Parte de um OutputDataConfigobjeto. A ARN AWS KMS chave usada para proteger o bucket do Amazon S3.	arn:aws:kms: **region-of-bucket:123456789012** :key/**1234abcd-12ab-34cd-56ef-1234567890ab**

Example : Corpo de uma solicitação de descrição de exportação feita usando o FHIR REST API

Quando for bem-sucedido, você receberá a seguinte JSON resposta.

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

Excluindo um armazenamento de dados no HealthLake

A exclusão de um armazenamento de dados é uma operação assíncrona. Depois de iniciado, o status muda para Excluindo. Um armazenamento de dados mantém o status de Exclusão até que todos os FHIR dados do armazenamento de dados e da infraestrutura subjacente necessária também sejam removidos.

Depois que os dados e a infraestrutura são removidos, o status do armazenamento de HealthLake dados muda para Excluído. Após a exclusão, os detalhes sobre seus armazenamentos de dados estarão disponíveis somente usando as `ListFHIRDataStores` operações `DescribeFHIRDataStore` e por sete dias. Depois de sete dias, o armazenamento de dados excluído não aparecerá nos resultados.

Para excluir com êxito um armazenamento de dados, o usuário, o grupo ou a função que faz a solicitação deve ter a IAM ação `glue:DeleteDatabase` adicionada à sua IAM política. Essa IAM ação não está incluída como parte da política AWS gerenciada,`AmazonHealthLakeFullAccess`.

Você pode excluir um armazenamento de dados com o AWS Management Console AWS SDKs, ou AWS CLI o.

Tópicos

Excluindo um armazenamento de dados (console)

Para excluir um armazenamento de dados com o console, escolha seu armazenamento de dados na página Armazenamentos de dados e escolha excluir.

Para excluir um armazenamento HealthLake de dados

1. Abra o HealthLake console em https://console.aws.amazon.com//healthlake/casa.
2. Abra o painel de navegação (≡).
3. Em seguida, escolha Armazenamentos de dados.
4. Na página Armazenamentos de dados, escolha a opção ao lado do armazenamento de dados que você deseja excluir.

5. Em seguida, escolha Excluir
6. Na caixa de diálogo, digite **`delete`** para confirmar que você deseja excluir o armazenamento de dados selecionado.
7. Em seguida, selecione Excluir. Em seguida, o status do seu armazenamento de dados mudará de Ativo para Excluído.

Excluindo um armazenamento de dados (AWS SDKse AWS CLI)

Você pode usar os exemplos de código abaixo para excluir um armazenamento HealthLake de dados.

AWS CLI

Os exemplos a seguir demonstram o uso da `DeleteFHIRDatastore` operação com o. AWS CLI Para executar o exemplo, é necessário instalar a AWS CLI.

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

Em caso de sucesso, você receberá a seguinte JSON resposta.

```
{
    "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)

O AWS SDK for Python suporta o `describe_fhir_datastore` método que recebe um único parâmetro. `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)
```

Quando bem-sucedido, ele retorna um dicionário 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 retornar detalhes sobre mais de um armazenamento de dados por vez, use `ListFHIRDatastore`

use o `DeleteFHIRDataStore` comando usando o AWS CLI conforme mostrado no exemplo a seguir. Você também pode excluir um armazenamento de dados usando o console delete-fhir-datastore APIou o console. A exclusão de um armazenamento de dados remove todas as versões de FHIR recursos contidas no armazenamento de dados e na infraestrutura subjacente. Os registros

relacionados a um armazenamento de dados excluído são mantidos na conta de serviço de acordo com as HIPAA diretrizes.

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

Conforme mostrado no exemplo de JSON resposta a seguir, o status muda para `DELETING` "" para confirmar que o armazenamento de dados e seu conteúdo estão em processo de exclusão.

```
{
  "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)"
}
```

Usando FHIR REST API interações com um armazenamento HealthLake de dados

Em AWS HealthLake, você usa REST API interações do Fast Healthcare Interoperability Resources (FHIR) para gerenciar e pesquisar FHIR recursos em seu armazenamento de dados. FHIRRESTAPIas interações são usadas para realizar interações de Criar, Ler, Atualizar e Excluir (CRUD) em recursos em um armazenamento de dados. Você também pode formar cadeias de caracteres de pesquisa complexas usando uma `POST` HTTP solicitação `GET` ou, pois HealthLake oferece suporte a um subconjunto de operações FHIR de pesquisa suportadas.

Para fins de conformidade, os tipos de FHIR recursos são validados de acordo com o HL7 FHIR recurso R4. StructureDefinition Para encontrar os recursos FHIR relacionados de um armazenamento de HealthLake dados ativo, faça uma `GET` solicitação onde `metadata` está especificado noURL, da seguinte forma.

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

Quando for bem-sucedido, você receberá um código de `200` HTTP resposta e a Declaração de Capacidade do seu armazenamento HealthLake de dados. Para obter mais informações, consulte `CapabilityStatement`a documentação do HL7 FHIR R4.

A tabela a seguir lista FHIR as interações suportadas pelo AWS HealthLake.

FHIRinterações apoiadas por AWS HealthLake

FHIRinteração	Descrição
Interações de todo o sistema	
`capabilities`	Obtenha uma declaração de capacidade para o sistema
`batch/transaction`	Atualizar, criar ou excluir um conjunto de recursos em uma única interação
Interações em nível de tipo	
`create`	Crie um novo recurso com uma ID atribuída ao servidor
`search`	Pesquise um tipo de recurso com base em alguns critérios de filtro

FHIRinteração	Descrição
`history`	Recupere o histórico de alterações de um determinado tipo de recurso
Interações em nível de instância	
`read`	Leia o estado atual de um recurso
`history`	Leia o histórico de alterações de um recurso específico
`vread`	Leia o estado de uma versão específica do recurso
`update`	Atualize um recurso pelo ID (ou crie-o se for novo)
`delete`	Excluir um recurso

Tópicos

- Tipos FHIR de recursos compatíveis em AWS HealthLake
- Executando as operações Criar, Ler, Atualizar e Excluir (CRUD) em armazenamentos HealthLake de dados
- Pesquisando seu armazenamento de HealthLake dados usando as FHIR REST API operações
- Lendo o histórico FHIR de recursos
- Obtendo dados do paciente com a operação Patient $everything FHIR REST API
- Exportando dados do seu armazenamento de HealthLake dados usando $export

Tipos FHIR de recursos compatíveis em AWS HealthLake

A tabela a seguir lista os tipos de recursos FHIR R4 suportados pelo AWS HealthLake. Para obter mais informações, consulte o Índice de recursos na documentação do HL7 FHIR R4.

FHIRTipos de recursos R4 compatíveis com HealthLake

Conta	DetectedIssue	Fatura	Praticante
ActivityDefinition	Dispositivo	Ferramentas	PractitionerRole
AdverseEvent	DeviceDefinition	Ligação	Procedimento

AllergyIntolerance	DeviceMetric	Listar	Proveniência
Nomeação	DeviceUseStatement	Local	Questionário
AppointmentResponse	DeviceRequest	Medida	QuestionnaireResponse
AuditEvent- Veja a nota	DiagnosticReport	MeasureReport	RelatedPerson
Binário	DocumentManifest	Mídia	RequestGroup
BodyStructure	DocumentReference	Medicação	ResearchStudy
Pacote - Veja a nota	EffectEvidenceSynthesis	MedicationAdministration	ResearchSubject
CapabilityStatement	Encontro	MedicationDispense	RiskAssessment
CarePlan	Endpoint	MedicationKnowledge	RiskEvidenceSynthesis
CareTeam	EpisodeOfCare	MedicationRequest	Schedule
ChargeItem	EnrollmentRequest	MedicationStatement	ServiceRequest
ChargeItemDefinition	EnrollmentResponse	MessageHeader	Slot
Reivindicar	ExplanationOfBenefit	MolecularSequence	Espécime
ClaimResponse	FamilyMemberHistory	NutritionOrder	StructureDefinition
Comunicação	Sinalizador	Observação	StructureMap
CommunicationRequest	Objetivo	OperationOutcome	Substance
Composição	Grupo	Organização	SupplyDelivery
ConceptMap	GuidanceResponse	OrganizationAffiliation	SupplyRequest

Condição	HealthcareService	Parâmetros	Tarefa
Consentimento	ImagingStudy	Paciente	ValueSet
Contrato	Imunização	PaymentNotice	VisionPrescription
Cobertura	ImmunizationEvaluation	PaymentReconciliation	VerificationResult
CoverageEligibilityRequest	ImmunizationRecommendation	Pessoa	
CoverageEligibilityResponse	InsurancePlan	PlanDefinition	

⚠ FHIRespecificações e HealthLake

- Você não pode fazer `GET` `POST` solicitações com esses tipos de recursos: binário OperationOutcome, pacote e parâmetros.
- AuditEvent— Um AuditEvent recurso pode ser criado ou lido, mas não pode ser atualizado ou excluído.
- Bundle — Há várias maneiras de HealthLake gerenciar solicitações de Bundle. Para mais informações, consulte Gerenciando vários FHIR recursos usando o Bundle.
- VerificationResult— Esse tipo de recurso só é compatível com armazenamentos de dados criados após 09 de dezembro de 2023.

Executando as operações Criar, Ler, Atualizar e Excluir (CRUD) em armazenamentos HealthLake de dados

Embora você use AWS ações nativas ao gerenciar armazenamentos de dados, importar dados e exportar dados, você usa quatro FHIR HTTP operações principais para criar (`POST`), ler (`GET`), atualizar (`PUT`) e excluir (`DELETE`) FHIR recursos em um armazenamento de HealthLake dados. Os tópicos a seguir descrevem como realizar as operações Criar, Ler, Atualizar e Excluir (CRUD) em seu armazenamento de HealthLake dados usando os FHIR REST API serviços. Você deve usar um

processo de assinatura do Signature versão 4 para autenticar HealthLake API solicitações enviadas por meio de um HTTP cliente. Para saber mais, consulte Processo de assinatura do Signature versão 4 no Referência geral da AWS.

Tópicos

Criando um recurso com **POST**

Você usa uma `POST` solicitação para criar um novo recurso em um armazenamento HealthLake de dados. `POST`as solicitações não exigem que você forneça um `id` elemento. Os HealthLake servidores retornam um código de HTTP status `201` Criado quando um recurso é criado com sucesso.

> **Note**
>
> Quando você faz uma `POST` solicitação sobre o tipo de `DocumentReference` recurso, as extensões existentes não são modificadas. Em vez disso, AWS HealthLake adiciona as novas extensões com as existentes ao seu armazenamento de dados. Para obter mais detalhes sobre como HealthLake usa o processamento de linguagem natural (NLP) no tipo de `DocumentReference` recurso para extrair dados médicos valiosos, consulteUsando a geração automatizada de recursos com base no processamento de linguagem natural (NLP) do tipo de FHIR DocumentReference recurso em AWS HealthLake.

Example Criação de um **`Patient`** recurso usando uma **`POST`** solicitação.

Para criar uma `POST` solicitação de armazenamento de HealthLake dados, use o endpoint do seu armazenamento de dados e forneça um corpo de JSON solicitação. Para encontrar o endpoint de um armazenamento de dados, consulte o HealthLake console em Armazenamentos de dados ou use a escribeFHIRDatastore operação D na AWS HealthLake APIReferência.

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"
}
```

Resposta do JSON

Para confirmar a criação do recurso do paciente, você receberá um código de HTTP status 201 Criado e a seguinte JSON resposta.

```
{
  "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"
  }
}
```

Lendo um recurso com **GET**

Este exemplo mostra como ler um FHIR recurso do paciente usando uma GET solicitação.

Example Lendo um **Patient** recurso específico usando uma **GET** solicitação.

Para criar uma GET solicitação HealthLake de armazenamento de dados, use o endpoint do seu armazenamento de dados. Para encontrar o endpoint de um armazenamento de dados, consulte o HealthLake console em Armazenamentos de dados ou use a escribeFHIRDatastore operação D na AWS HealthLake APIReferência.

Você também deve incluir o tipo de recurso **Patient** e um 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
```

Resposta do JSON

Quando for bem-sucedido, você receberá um código de 200 HTTP status e a seguinte JSON resposta.

```
{
    "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"
}
```

Atualizando um recurso usando **PUT**

O exemplo a seguir mostra como usar PUT para atualizar detalhes sobre um paciente no tipo de FHIR recurso do paciente. Além disso, quando você faz uma PUT solicitação em um recurso ainda não criado, ele cria uma versão inicial.

Sua solicitação retornará um código de `200` HTTP status se o recurso tiver sido atualizado ou retornará um código de `201` HTTP status se um novo recurso for criado.

> **Note**
>
> Quando você faz uma PUT solicitação sobre o tipo de `DocumentReference` recurso, as extensões existentes não são modificadas. Em vez disso, AWS HealthLake adiciona as novas extensões com as existentes ao seu armazenamento de dados. Para obter mais detalhes sobre como HealthLake usa o processamento de linguagem natural (NLP) no tipo de `DocumentReference` recurso para extrair dados médicos valiosos, consulteUsando a geração automatizada de recursos com base no processamento de linguagem natural (NLP) do tipo de FHIR DocumentReference recurso em AWS HealthLake.

Example Atualização de um tipo de **`Patient`** recurso usando uma **PUT** solicitação

Ao fazer uma PUT solicitação, você precisará do endpoint do armazenamento de dados, do nome do tipo de recurso que deseja atualizar, de um identificador e do corpo da JSON solicitação.

Se você usa PUT para criar um novo recurso, ele usa o identificador fornecido para criar o novo recurso.

PUT Request

Exemplo de estrutura de uma PUT solicitação válida:

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

JSON Request Body

Um exemplo de JSON corpo usado para atualizar o recurso especificado do paciente.

```
{
  "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"
}
```

Resposta JSON

Você receberá o seguinte JSON em resposta para confirmar a alteração:

```
{
 "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"
 }
}
```

Atualização condicional

A atualização condicional permite atualizar um recurso existente com base em alguns critérios de pesquisa de identificação, em vez de por identificação lógica. Quando o servidor processa essa atualização, ele executa uma pesquisa usando seus recursos de pesquisa padrão para o tipo de recurso, com o objetivo de resolver uma única identificação lógica para essa solicitação.

A ação necessária depende de quantas correspondências são encontradas:

- Sem correspondências, sem identificação fornecida no corpo da solicitação: o servidor cria o recurso.
- Sem correspondências, id fornecido e o recurso ainda não existe com o id: o servidor trata a interação como uma interação de atualização como criação.
- Nenhuma correspondência, ID fornecida e já existe: o servidor rejeita a atualização com um `409 Conflict` erro.
- Uma correspondência, nenhum ID de recurso fornecido OU (ID do recurso fornecido e corresponde ao recurso encontrado): O servidor executa a atualização em relação ao recurso correspondente conforme descrito acima, onde, se o recurso foi atualizado, o servidor SHALL retornará um`200 OK`;

- One Match, ID do recurso fornecido, mas não corresponde ao recurso encontrado: o servidor retorna um `409 Conflict` erro indicando que a especificação do ID do cliente era um problema, preferencialmente com um OperationOutcome
- Várias correspondências: o servidor retorna um `412 Precondition Failed` erro indicando que os critérios do cliente não foram seletivos o suficiente, de preferência com um OperationOutcome

Example — Atualize um recurso para pacientes cujo nome é peter, data de nascimento é 1º de janeiro de 2000 e número de telefone 1234567890:

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

Excluindo um recurso usando **DELETE**

Para excluir um recurso em seu armazenamento de HealthLake dados, você deve fazer uma `DELETE` HTTP solicitação.

Example Excluindo um tipo de **`Patient`** recurso específico usando uma **`DELETE`** solicitação.

Para criar uma `DELETE` solicitação, use o endpoint do armazenamento de dados. Para encontrar o endpoint de um armazenamento de dados, consulte o HealthLake console em Armazenamentos de dados ou use a escribeFHIRDatastore operação D encontrada na AWS HealthLake APIReferência.

Você também deve incluir o tipo de recurso e um identificador válido.

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

Resposta HTTP

Quando for bem-sucedido, você receberá um código de `204` HTTP status confirmando que o recurso não está mais no armazenamento de dados. Quando uma solicitação de exclusão falhar, você receberá um código de HTTP status da série 400 indicando por que a `DELETE` solicitação falhou.

Gerenciando vários FHIR recursos usando o Bundle

Na especificação HL7 FHIR R4, os pacotes são simplesmente uma coleção de recursos. HealthLake suporta a criação de um tipo de recurso de pacote em uma FHIR REST API solicitação e o uso de uma transação de pacote para realizar várias CRUD operações em uma única FHIR REST API

solicitação. Em uma transação de pacote, você deve especificar o tipo de pacote como `batch` na FHIR REST API solicitação.

Todas as solicitações de pacotes são registradas por AWS CloudTrail. Para saber mais sobre como usar CloudTrail com HealthLake, consulteRegistro em log de chamadas do AWS HealthLake API com o AWS CloudTrail.

HL7FHIRRecursos R4 (externos)

- Para ler a especificação completa, consulte Tipo de recurso: pacote no Índice de FHIR documentação.
- Para ler sobre interações em lote usando o FHIR RESTAPI, consulte Interações em lote usando o FHIR REST API no Índice de FHIR documentação.

As seções abaixo descrevem como estruturar uma FHIR REST API solicitação para criar um novo recurso de pacote ou processar recursos individualmente usando transações de pacote.

> ⚠ Diferenças entre o HealthLake console AWS CLI, o e o AWS SDKs
>
> O HealthLake console só oferece suporte a operações do tipo Bundle em que o tipo de recurso Bundle é especificado na FHIR REST API solicitação. URL

Executando várias CRUD operações usando FHIR pacotes

Quando nenhum tipo de recurso é especificado em sua solicitaçãoURL, a FHIR REST API solicitação é analisada como transações individuais do armazenamento de dados. Cada CRUD operação fornecida no JSON corpo é avaliada e um código de HTTP status específico é retornado. HealthLake suporta o tipo `batch` de pacote.

Para realizar várias CRUD operações em uma única FHIR REST API solicitação, faça o seguinte:

A lista a seguir mostra partes truncadas de um corpo de solicitação usado na solicitação de pacote FHIR RESTAPI. Para ver o corpo completo da solicitação, consulte Criação de uma solicitação de pacote envolvendo várias CRUD operações.

1. Não especifique um tipo de recurso em sua POST solicitação:

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

2. No corpo da solicitação, especifique o tipo de pacote como `"type": "batch"`
3. No corpo da solicitação, especifique dados específicos do recurso para cada CRUD interação começando com a chave. `resource`
4. Cada CRUD operação é especificada como a `request` no corpo da solicitação da seguinte forma:

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

Na JSON resposta, você recebe um código de HTTP status para cada CRUD operação especificada na solicitação.

HealthLake limita as transações do pacote

- Para saber mais sobre os limites HealthLake impostos aos pacotes, consulteAWS HealthLake endpoints e cotas.

Veja a seguir um exemplo de uma operação de pacote contendo várias CRUD operações.

Example — Criar uma solicitação de pacote envolvendo várias CRUD operações.

Para fazer uma FHIR REST API solicitação que execute várias CRUD operações, você deve fazer uma `POST` solicitação usando o endpoint do armazenamento de dados e fornecer um corpo da JSON solicitação.

Você pode encontrar o endpoint do seu armazenamento de dados no HealthLake console em Armazenamentos de dados ou usando a escribeFHIRDatastore operação D na AWS HealthLake APIReferência.

POST Request

Faça uma `POST` solicitação usando o endpoint do seu armazenamento de dados. Use a próxima guia, Corpo da JSON solicitação, para ver os elementos necessários do corpo da solicitação.

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

JSON Request Body

No corpo da solicitação, você deve fornecer os seguintes pares FHIR chave-valor junto com quaisquer outros dados específicos do recurso sobre as solicitações individuais. CRUD O primeiro exemplo mostra um corpo de JSON solicitação truncado destacando os elementos necessários. O segundo exemplo mostra o corpo completo da JSON solicitação.

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

Aqui está um exemplo completo que mostra a criação de um novo `Patient` tipo de `Medication` 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"
      }
    }
  ]
}
```

Resposta do JSON

Para confirmar a criação dos recursos especificados no exemplo de transação do pacote, você recebe 201 o código de HTTP status Created para cada CRUD operação incluída. Quando uma CRUD operação falha, você obtém o HTTP status da série 400 indicando por que a solicitação individual falhou.

```
{
  "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"
      }
    }
```

```
  ]
}
```

Agrupando recursos como um tipo de recurso Bundle

Para criar um novo tipo de recurso do Bundle, você deve especificar `Bundle` na FHIR REST API solicitação e fornecer um JSON corpo válido contendo os recursos que você deseja agrupar.

Quando o pacote é especificado na solicitaçãoURL, o conteúdo do corpo da JSON solicitação é salvo no armazenamento de HealthLake dados como está. Portanto, nenhuma CRUD operação pode ser executada nos tipos de recursos individuais. Pacotes desse tipo recebem uma única nova ID de recurso. Como os recursos são salvos no estado em que se encontram, você não pode fazer `GET` `POST` solicitações sobre recursos individuais salvos no tipo de recurso Bundle.

> **Note**
>
> A especificação HL7 FHIR R4 também oferece suporte ao agrupamento de recursos usando Grupo, Composição e Lista. Quando você cria esses tipos de recursos, os recursos individuais não são contidos diretamente. Em vez disso, eles usam o `Reference` elemento para apontar para os recursos individuais. Portanto, o uso desses tipos de recursos permite que você modifique os recursos individuais contidos neles.

Para criar um tipo de `Bundle` recurso, você deve especificá-lo em sua `POST` solicitação e fornecer uma JSON enumeração dos recursos que você deseja incluir.

Example — Criação de um recurso de pacote usando uma solicitação **`POST`**

Para criar um `bundle` recurso, faça o seguinte:

1. Formate uma FHIR REST API solicitação da seguinte forma:

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

2. Forneça um JSON corpo que especifique os recursos que você deseja agrupar. Este exemplo agrupa dois 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"
          }
        ]
      }
    }
  ]
}
```

Pesquisando seu armazenamento de HealthLake dados usando as FHIR REST API operações

HealthLake suporta a pesquisa em seu armazenamento de dados usando as REST API operações fornecidas como parte do FHIR padrão. Nesta seção, você encontrará exemplos de como fazer `GET` `POST` solicitações em vários tipos de recursos diferentes.

> **Note**
>
> Para consultas que envolvam informações de identificação pessoal (PII) ou Informações de saúde protegidas (PHI), é recomendável usar solicitações. `POST` Em uma `POST` solicitação, PII ou PHI é adicionado como parte do corpo da solicitação e é criptografado em trânsito.

A FHIR especificação oferece suporte a vários tipos de parâmetros de pesquisa, mas HealthLake suporta somente um subconjunto. Para ter mais informações, consulte Tipos de parâmetros de pesquisa compatíveis e Parâmetros de pesquisa avançada suportados por HealthLake.

Pesquisando seu armazenamento de dados usando as FHIR REST API operações.

- Tipos de parâmetros de pesquisa compatíveis
- Parâmetros de pesquisa avançada suportados por HealthLake
  - _include
  - _revinclude
  - _summary
  - _elements

- _total
- _sort
- _count
- Chaining and Reverse Chaining(_has)
- Modificadores de pesquisa compatíveis
- Comparadores de pesquisa compatíveis
- Parâmetros de pesquisa não suportados pelo HealthLake
- Pesquise com POST exemplos
- Pesquise com GET exemplos

Tipos de parâmetros de pesquisa compatíveis

A tabela a seguir mostra os tipos de parâmetros de pesquisa suportados em HealthLake.

Tipos de parâmetros de pesquisa compatíveis

Parâmetro de pesquisa	Descrição
_id	ID do recurso (não completoURL)
_lastUpdated	Data da última atualização. O servidor tem poder discricionário quanto à precisão do limite.
_tag	Pesquise por uma tag de recurso.
_perfil	Pesquise todos os recursos marcados com um perfil.
_segurança	Pesquise as etiquetas de segurança aplicadas a esse recurso.
_fonte	Pesquise de onde vem o recurso.
_texto	Pesquise a narrativa do recurso.
createdAt	Pesquise na extensão personalizada -createdA t.

> **Note**
>
> Os seguintes parâmetros de pesquisa são compatíveis somente com datastores criados após 09 de dezembro de 2023: _security, _source, _text,. createdAt

A tabela a seguir mostra exemplos de como modificar cadeias de caracteres de consulta com base nos tipos de dados especificados para um determinado tipo de recurso. Para maior clareza, os caracteres especiais na coluna de exemplos não foram codificados. Para fazer uma consulta bem-sucedida, certifique-se de que a string de consulta tenha sido codificada corretamente.

Tipos de parâmetros de pesquisa	Detalhes	Exemplos

Número	Pesquisa um valor numérico em um recurso especificado. Números significativos são observados.
	
	O número de dígitos significativos é específico por valor do parâmetro de pesquisa, excluindo zeros à esquerda.
	
	Prefixos de comparação são permitidos.
		`[parameter]=100`
		
		`[parameter]=1e2`
		
		`[parameter]=lt100`

Data/ DateTime	Pesquisa uma data ou hora específica. O formato esperado é `yyyy-mm-ddThh:mm:ss[Z|(+|-)hh:mm]` , mas pode variar.
	
	Aceita os seguintes tipos de dados: `date` `dateTime` `instant`,`Period`,, `Timing` e. Para obter mais detalhes sobre o uso desses
		`[parameter]=eq2013-01-14`
		
		`[parameter]=gt2013-01-14T10:00`
		
		`[parameter]=ne2013-01-14`

Tipos de parâmetros de pesquisa	Detalhes	Exemplos
	tipos de dados em pesquisas , consulte a data no Índice de FHIR documentação. Prefixos de comparação são permitidos.	

String	Pesquisa uma sequência de caracteres com distinção entre maiúsculas e minúsculas.
	
	Suporta ambos `HumanName` os `Address` tipos. Para obter mais detalhes, consulte a entrada do tipo de HumanName `Address` dados e as entradas do tipo de dados no Índice de FHIR documentação.
	
	A pesquisa avançada é suportada usando `:text` modificadores.
		`[base]/Patient?given=eve`
		
		`[base]/Patient?given:contains=eve`

Tipos de parâmetros de pesquisa	Detalhes	Exemplos
Token	Pesquisa uma close-to-exact correspondência com uma sequência de caracteres, geralmente comparada a um par de valores de código médico. A distinção entre maiúsculas e minúsculas está vinculada ao sistema de código usado ao criar uma consulta. As consultas baseadas em subsunção podem ajudar a reduzir problemas relacionados à distinção entre maiúsculas e minúsculas. Para maior clareza, não `|` foi codificado.	`[parameter]=[system]|[code]` : Aqui `[system]` se refere a um sistema de codificação e `[code]` se refere ao valor do código encontrado nesse sistema específico. `[parameter]=[code]` : Aqui, sua entrada corresponderá a um código ou a um sistema. `[parameter]=|[code]` : aqui, sua entrada corresponderá a um código e a propriedade do sistema não terá identificador.

Composto	Pesquisa vários parâmetros em um único tipo de recurso, usando os modificadores `$` e a `,` operação.
	
	Prefixos de comparação são permitidos.
		`/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 parâmetros de pesquisa	Detalhes	Exemplos

Quantidade	Pesquisa um número, sistema e código como valores. É necessário um número, mas o sistema e o código são opcionais. Com base no tipo de dados Quantidade. Para obter mais detalhes, consulte Quantidade no Índice de FHIR Documentação.
	
	Usa a seguinte sintaxe presumida `[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`

Referência	Pesquisa referências a outros recursos.	`[base]/Observation?subject=Patient/23`
		
		`test`

URI	Pesquisa uma sequência de caracteres que identifica inequivocamente um recurso específico.	`[base]/ValueSet?url=http://acme.org/fhir/ValueSet/123`
Especial	Pesquisas baseadas em NLP extensões médicas integradas.	

Parâmetros de pesquisa avançada suportados por HealthLake

HealthLake suporta os seguintes parâmetros de pesquisa avançada.

Nome	Descrição	Exemplo	Recurso
`_include`	Usado para solicitar que recursos adicionais sejam retornados em uma solicitação de pesquisa. Ele retorna recursos que são referenciados pela instância do recurso de destino.	`Encounter?_include=Encounter:subject`	
`_revinclude`	Usado para solicitar que recursos adicionais sejam retornados em uma solicitação de pesquisa. Ele retorna recursos que fazem referência à instância do recurso primário.	`Patient?_id=` **`patient-identifier`** `&_revinclude=Encounter:patient`	
`_summary`	O resumo pode ser usado para solicitar um subconjunto do recurso.	`Patient?_summary=text`	Os seguintes parâmetros de resumo são suportados:`_summary=true` ,`_summary=false` ,`_summary=text` ,`_summary=data` .
`_element`	Solicite que um conjunto específico de elementos seja retornado como parte de um recurso nos resultados da pesquisa.	`Patient?_elements=identifier,active,link`	
`_total`	Retorna o número de recursos que correspondem aos parâmetros de pesquisa.	`Patient?_total=accurate`	Support`_total=accurate` ,`_total=none` .

Nome	Descrição	Exemplo	Recurso
`_sort`	Indique a ordem de classificação dos resultados da pesquisa retornados usando uma lista separada por vírgulas. O `-` prefixo pode ser usado para qualquer regra de classificação na lista separada por vírgulas para indicar a ordem decrescente.	`Observation?_sort=status,-date`	Support classificar por campos com tipos`Number, String, Quantity, Token, URI, Reference` . A classificação por só `Date` é compatível com datastores criados após 09 de dezembro de 2023. Support até 5 regras de classificação.
`_count`	Controle quantos recursos são retornados por página do pacote de pesquisa.	`Patient?_count=100`	O tamanho máximo da página é 100.
`chainin`	Elementos de pesquisa dos recursos referenciados. O `.` direciona a pesquisa em cadeia para o elemento dentro do recurso referenciado.	`DiagnosticReport?subject:Patient.name=peter`	
`reverse chainin (_has)`	Pesquise um recurso com base nos elementos dos recursos que se referem a ele.	`Patient?_has:Observation:patient:code=1234-5`	

`_include`

O uso `_include` em uma consulta de pesquisa permite que FHIR recursos adicionais especificados também sejam retornados. Use `_include` para incluir recursos que estão vinculados diretamente.

Example — Para usar **`_include`** para encontrar os pacientes ou o grupo de pacientes que foram diagnosticados com tosse

Você pesquisaria o tipo de `Condition` recurso especificando o código de diagnóstico para tosse e, em seguida, `_include` especificaria que deseja que o `subject` diagnóstico também seja retornado.

No tipo de `Condition` recurso, `subject` refere-se ao tipo de recurso do paciente ou ao tipo de recurso do grupo.

Para maior clareza, os caracteres especiais no exemplo não foram codificados. Para fazer uma consulta bem-sucedida, certifique-se de que a sequência de caracteres de consulta tenha sido codificada corretamente.

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

`_revinclude`

O uso `_revinclude` em uma consulta de pesquisa permite que FHIR recursos adicionais especificados também sejam retornados. Use `_revinclude` para incluir recursos vinculados de trás para frente.

Example — Usar **`_revinclude`** para incluir tipos de recursos relacionados de Encontro e Observação vinculados a um paciente específico

Para fazer essa pesquisa, primeiro defina o indivíduo `Patient` especificando seu identificador no parâmetro de `_id` pesquisa. Em seguida, você especificaria FHIR recursos adicionais usando a estrutura `Encounter:patient` `Observation:patient` e.

Para maior clareza, os caracteres especiais no exemplo não foram codificados. Para fazer uma consulta bem-sucedida, certifique-se de que a sequência de caracteres de consulta tenha sido codificada corretamente.

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

`_summary`

O uso `_summary` em uma consulta de pesquisa permite que o usuário solicite um subconjunto do FHIR recurso. Ele pode conter um dos seguintes valores:`true, text, data, false`. Quaisquer outros valores serão tratados como inválidos. Os recursos retornados serão marcados com `'SUBSETTED'` meta.tag, para indicar que os recursos estão incompletos.

- `true`: retorne todos os elementos suportados que estão marcados como 'resumo' na definição básica do (s) recurso (s).

- `text`: retorne somente os elementos 'text', 'id', 'meta' e somente os elementos obrigatórios de nível superior.
- `data`: Retorne todas as partes, exceto o elemento 'texto'.
- `false`: Retorne todas as partes do (s) recurso (s)

Em uma única solicitação de pesquisa, `_summary=text` não pode ser combinado com `_include` ou com parâmetros de `_revinclude` pesquisa.

Example — Obtenha o elemento “texto” dos recursos do paciente em um armazenamento de dados.

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

`_elements`

O uso `_elements` em uma consulta de pesquisa permite que elementos FHIR de recursos específicos sejam solicitados. Os recursos retornados serão marcados com `'SUBSETTED'` meta.tag, para indicar que os recursos estão incompletos.

O `_elements` parâmetro consiste em uma lista separada por vírgulas de nomes de elementos básicos, como elementos definidos no nível raiz do recurso. Somente os elementos listados devem ser retornados. Se os valores dos `_elements` parâmetros contiverem elementos inválidos, o servidor os ignorará e retornará elementos obrigatórios e válidos.

`_elements`não será aplicável aos recursos incluídos (recursos retornados cujo modo de pesquisa é`include`).

Em uma única solicitação de pesquisa, `_elements` não pode ser combinado com os parâmetros `_summary` de pesquisa.

Example — Obtenha elementos “identificadores”, “ativos” e “vinculados” dos recursos do paciente em seu HealthLake armazenamento de dados.

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

_total

O uso `_total` em uma consulta de pesquisa retornará o número de recursos que correspondem aos parâmetros de pesquisa solicitados. HealthLake retornará o número total de recursos correspondentes (recursos retornados cujo modo de pesquisa é`match`) na resposta `Bundle.total` da pesquisa.

`_total`suporta os `accurate` valores dos `none` parâmetros. `_total=estimate`não é suportado. Quaisquer outros valores serão tratados como inválidos. `_total`não é aplicável aos recursos incluídos (recursos retornados cujo modo de pesquisa é`include`).

Example — Obtenha o número total de recursos do paciente em um armazenamento de dados:

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

_sort

O uso `_sort` na consulta de pesquisa organiza os resultados em uma ordem específica. Os resultados são ordenados com base na lista separada por vírgulas das regras de classificação em ordem de prioridade. As regras de classificação devem ser parâmetros de pesquisa válidos. Quaisquer outros valores serão tratados como inválidos.

Em uma única solicitação de pesquisa, você pode usar até 5 parâmetros de pesquisa de classificação. Opcionalmente, você pode usar um `-` prefixo para indicar a ordem decrescente. O servidor classificará em ordem crescente por padrão.

Os tipos de parâmetros de pesquisa de classificação suportados são:`Number, String, Date, Quantity, Token, URI, Reference`. Se um parâmetro de pesquisa se referir a um elemento aninhado, esse parâmetro de pesquisa não terá suporte para classificação. Por exemplo, a pesquisa por “nome” do tipo de recurso Paciente se refere ao paciente. O elemento Nome com o tipo de HumanName dados é considerado aninhado. Portanto, não há suporte para classificar os recursos do paciente por 'nome'.

Example — Obtenha os recursos do paciente em um armazenamento de dados e classifique-os por data de nascimento em ordem crescente:

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

`_count`

O parâmetro `_count` é definido como uma instrução para o servidor sobre quantos recursos devem ser retornados em uma única página.

O tamanho máximo da página é 100. Qualquer valor maior que 100 é inválido. `_count=0`não é suportado.

Example — Pesquise o recurso do paciente e defina o tamanho da página de pesquisa para 25:

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

Chaining and Reverse Chaining(_has)

O encadeamento e o encadeamento reverso FHIR fornecem uma maneira mais eficiente e compacta de obter dados interconectados, reduzindo a necessidade de várias consultas separadas e tornando a recuperação de dados mais conveniente para desenvolvedores e usuários.

Se algum nível de recursão retornar mais de 100 resultados, HealthLake retornará 4xx para proteger o armazenamento de dados de ser sobrecarregado e causar várias paginações.

Example — Encadeamento - Obtém tudo DiagnosticReport o que se refere a um paciente em que o nome do paciente é peter.

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

Example — Encadeamento reverso - Obtenha recursos do paciente, em que o recurso do paciente é referido por pelo menos uma observação em que a observação tem um código de 1234 e onde a observação se refere ao recurso do paciente no parâmetro de pesquisa do paciente.

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

Modificadores de pesquisa compatíveis

Os modificadores de pesquisa são usados com campos baseados em strings. Todos os modificadores de pesquisa em HealthLake uso usam lógica baseada em booleanos. Por exemplo,

você pode `:contains` especificar que um campo de string maior inclua uma string pequena para que seja incluído nos resultados da pesquisa.

Modificadores de pesquisa compatíveis

Modificador de pesquisa	Tipo
:ausente	Todos os parâmetros, exceto `Composite`
:exato	String
:contém	String
:não	Token
:texto	Token
:identificador	Referência

Comparadores de pesquisa compatíveis

Você pode usar comparadores de pesquisa para controlar a natureza da correspondência em uma pesquisa. Você pode usar comparadores ao pesquisar nos campos de número, data e quantidade. A tabela a seguir lista os comparadores de pesquisa e suas definições que são suportadas pelo HealthLake.

Comparadores de pesquisa compatíveis

Comparador de pesquisa	Descrição
eq	O valor do parâmetro no recurso é igual ao valor fornecido.
um	O valor do parâmetro no recurso não é igual ao valor fornecido.
gt	O valor do parâmetro no recurso é maior que o valor fornecido.

Comparador de pesquisa	Descrição
lt	O valor do parâmetro no recurso é menor que o valor fornecido.
idade	O valor do parâmetro no recurso é maior ou igual ao valor fornecido.
le	O valor do parâmetro no recurso é menor ou igual ao valor fornecido.
mar	O valor do parâmetro no recurso começa após o valor fornecido.
eb	O valor do parâmetro no recurso termina antes do valor fornecido.

Parâmetros de pesquisa não suportados pelo HealthLake

Para obter uma lista completa dos parâmetros de pesquisa compatíveis, consulte o registro FHIR de parâmetros de pesquisa. HealthLake suporta todos os parâmetros de pesquisa, exceto aqueles listados na tabela.

Parâmetros de pesquisa não suportados

Composição do pacote	Localização próxima
Identificador de pacote	C onsent-source-reference
Mensagem do pacote	Paciente contratado
Tipo de pacote	Conteúdo do recurso
Carimbo de data/hora do pacote	Consulta de recursos

Pesquise com POST exemplos

Você pode pesquisar um armazenamento de HealthLake dados fazendo POST solicitações. Você pode fornecer parâmetros de consulta no corpo da solicitação URI ou em um, mas não pode usar os dois em uma única solicitação.

Os exemplos neste tópico seguem essa prática recomendada.

> **Note**
>
> Para consultas que envolvam informações de identificação pessoal (PII) ou Informações de saúde protegidas (PHI), é recomendável usar solicitações. `POST` Em uma `POST` solicitação, PII ou PHI é adicionado como parte do corpo da solicitação e é criptografado em trânsito.

Ao fazer uma POST solicitação com um parâmetro no corpo da solicitação, use `Content-Type: application/x-www-form-urlencoded` como parte do cabeçalho.

Este tópico fornece exemplos de como pesquisar POST usando os seguintes tipos de recursos.

- Idade: A idade não é um tipo de recurso definido emFHIR. Em vez disso, a idade é capturada como parte do tipo de recurso do paciente. Para pesquisar um grupo de pacientes com base em idade ou faixa etária específica, use the section called “Comparadores de pesquisa compatíveis” a. Para obter mais detalhes, consulte Tipo de recurso: Paciente no Índice de FHIR Documentação.
- Condição: Esse tipo de recurso armazena detalhes relacionados a conceitos clínicos, como diagnóstico, situações, condição clínica e problemas que atingiram um nível de preocupação. Para saber mais, consulte Tipo de recurso: Condição no Índice de FHIR documentação. HealthLake cria novas condições com base nos documentos encontrados no DocumentReference. Essas adições são excluídas por padrão ao fazer uma `POST` solicitação. Para incluí-los, você deve especificar um identificador válido para um recurso de condição em sua pesquisa.
- DocumentReference:Esse tipo de recurso é suportado pelo HealthLake. Esse tipo de recurso suporta documentos de referência de qualquer tipo. Para saber mais, consulte Tipo de recurso: DocumentReference no Índice de FHIR documentação. HealthLake também fornece processamento integrado de linguagem natural (NLP) de documentos encontrados no DocumentReference. Para saber mais, consulte Usando a geração automatizada de recursos com base no processamento de linguagem natural (NLP) do tipo de FHIR DocumentReference recurso em AWS HealthLake.

- Local: esse tipo de recurso inclui locais incidentais (um local usado para assistência médica sem designação ou autorização prévia) e locais dedicados e designados formalmente. Para obter mais detalhes, consulte Tipo de recurso: localização no Índice de FHIR documentação.
- Observação: medições e afirmações simples feitas sobre um paciente, dispositivo ou outro assunto. HealthLake cria novos recursos de observação com base nos documentos encontrados no DocumentReference recurso. Para saber mais sobre como HealthLake criar novos recursos, consulteUsando a geração automatizada de recursos com base no processamento de linguagem natural (NLP) do tipo de FHIR DocumentReference recurso em AWS HealthLake. Essas adições são excluídas por padrão ao fazer uma `POST` solicitação. Para incluí-los, você deve especificar um identificador válido para um recurso de observação em sua pesquisa. Para saber mais, consulte Tipo de recurso: Observação no Índice de FHIR Documentação.

Cada guia mostra exemplos de como pesquisar no tipo de recurso especificado. Ele inclui um exemplo de como especificar a solicitação no corpo da solicitação.

Age

Use o seguinte para fazer uma solicitação de pesquisa `POST` baseada no tipo de `Patient` recurso. Essa pesquisa usa o `eq` comparador de pesquisa para pesquisar indivíduos que nasceram em 1997.

Você precisa especificar uma solicitação URL e um corpo da solicitação. Aqui está um exemplo de solicitaçãoURL.

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

Para especificar o ano de 1997 na pesquisa, você adicionaria o seguinte elemento ao corpo da solicitação.

```
birthdate=eq1997
```

Resposta do JSON

Quando for bem-sucedido, você receberá um código de `200` HTTP resposta e uma JSON resposta semelhante.

Condition

Use o seguinte para fazer uma `POST` solicitação sobre o tipo de `Condition` recurso. Essa pesquisa encontra locais em seu armazenamento HealthLake de dados que contêm o código médico`72892002`.

Você precisa especificar uma solicitação URL e um corpo da solicitação. Aqui está um exemplo de solicitaçãoURL.

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

Para especificar o código médico que você deseja pesquisar, adicione esse JSON elemento ao corpo da solicitação.

```
code=72892002
```

Resposta do JSON

Quando for bem-sucedido, você receberá um código de `200` HTTP resposta. A JSON resposta a seguir foi truncada para maior clareza.

```
{
 "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 os resultados HealthLake do processamento de linguagem natural integrado (NLP) ao fazer uma POST solicitação sobre o tipo de `DocumentReference` recurso, formate uma solicitação da seguinte forma.

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

Para especificar o DocumentReference elemento que você deseja referenciar, consulteParâmetros de pesquisa. Você os especificará no corpo da solicitação comoJSON.

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

Essa sequência de caracteres de consulta usa vários parâmetros de pesquisa para pesquisar nas API operações do Amazon Comprehend Medical usadas para gerar os resultados médicos integrados. NLP

Location

Use o seguinte para fazer uma `POST` solicitação sobre o tipo de `Location` recurso. Essa pesquisa encontra locais em seu armazenamento de HealthLake dados que contêm o nome da cidade Boston como parte do endereço.

Você deve especificar uma solicitação URL e um corpo da solicitação. Aqui está um exemplo de solicitaçãoURL.

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

Para especificar Boston na pesquisa, adicione o seguinte elemento ao corpo da solicitação:

```
address=Boston
```

Resposta do JSON

Quando for bem-sucedido, você receberá um código de `200` HTTP resposta. A JSON resposta foi truncada para maior clareza.

```
{
 "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

Use o seguinte para fazer uma solicitação de pesquisa POST baseada no tipo de `Observation` recurso. Essa pesquisa usa o parâmetro de `value-concept` pesquisa para procurar o código médico,`266919005`. Esse status indica`Never smoker`.

Você precisa especificar uma solicitação URL e um corpo da solicitação. Aqui está um exemplo de solicitaçãoURL.

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

Para especificar o status`Never smoker`, defina `value-concept=266919005` no corpo doJSON.

```
value-concept=266919005
```

Resposta do JSON

Quando for bem-sucedido, você receberá um código de `200` HTTP resposta. A JSON resposta a seguir foi truncada para maior clareza.

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

Pesquise com GET exemplos

Você pode pesquisar um armazenamento de HealthLake dados fazendo GET solicitações. HealthLake só oferece suporte ao fornecimento de parâmetros de consulta como parte doURI, e não como parte do corpo da solicitação.

> **Note**
>
> Para consultas que envolvam informações de identificação pessoal (PII) ou Informações de saúde protegidas (PHI), é recomendável usar solicitações. `POST` Em uma `POST` solicitação, PII ou PHI é adicionado como parte do corpo da solicitação e é criptografado em trânsito.

O tópico fornece exemplos de como pesquisar `GET` usando tipos de recursos compatíveis em HealthLake.

- Idade: A idade não é um tipo de recurso definido emFHIR. Em vez disso, a idade é capturada como parte do tipo de recurso do paciente. Para pesquisar um grupo de pacientes com base em idade ou faixa etária específica, você precisa usar umthe section called “Comparadores de pesquisa compatíveis”. Para obter mais detalhes, consulte Tipo de recurso: Paciente no Índice de FHIR Documentação.
- Condição: Esse tipo de recurso armazena detalhes relacionados a conceitos clínicos, como diagnóstico, situações, condição clínica e problemas que atingiram um nível de preocupação. Para saber mais, consulte Tipo de recurso: Condição no Índice de FHIR documentação. HealthLake cria novas condições com base nos documentos encontrados no DocumentReference. Essas adições são excluídas por padrão ao fazer uma `POST` solicitação. Para incluí-los, você deve especificar um identificador válido para um recurso de condição em sua pesquisa.
- DocumentReference:Esse tipo de recurso é suportado pelo HealthLake. Esse tipo de recurso suporta documentos de referência de qualquer tipo. Para saber mais, consulte Tipo de recurso: DocumentReference no Índice de FHIR documentação. HealthLake também fornece

processamento integrado de linguagem natural (NLP) de documentos encontrados no DocumentReference. Para saber mais, consulte Usando a geração automatizada de recursos com base no processamento de linguagem natural (NLP) do tipo de FHIR DocumentReference recurso em AWS HealthLake.

- Local: Esse tipo de recurso inclui locais incidentais (um local usado para assistência médica sem designação ou autorização prévia) e locais dedicados e designados formalmente. Para obter mais detalhes, consulte Tipo de recurso: Local no FHIR Índice de Documentação.
- Observação: medições e afirmações simples feitas sobre um paciente, dispositivo ou outro assunto. HealthLake cria novos recursos de observação com base nos documentos encontrados no DocumentReference recurso. Para saber mais sobre como HealthLake criar novos recursos, consulteUsando a geração automatizada de recursos com base no processamento de linguagem natural (NLP) do tipo de FHIR DocumentReference recurso em AWS HealthLake. Essas adições são excluídas por padrão ao fazer uma `POST` solicitação. Para incluí-los, você deve especificar um identificador válido para um recurso de observação em sua pesquisa. Para saber mais, consulte Tipo de recurso: Observação no Índice de FHIR Documentação.

Cada guia mostra um exemplo de como pesquisar no tipo de recurso especificado. Inclui um exemplo de como especificar a solicitação no URI e a JSON resposta relacionada.

Age

Use o seguinte para fazer uma solicitação de pesquisa `GET` baseada no tipo de `Patient` recurso. Essa pesquisa usa o `eq` comparador de pesquisa para pesquisar indivíduos que nasceram em 1997.

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

Resposta do JSON

Quando for bem-sucedido, você receberá um código de `200` HTTP resposta.

Condition

Use o seguinte para fazer uma `GET` solicitação sobre o tipo de `Condition` recurso. Essa pesquisa encontra locais em seu armazenamento HealthLake de dados que contêm o código médico`72892002`.

Você precisa especificar uma solicitação URL e um corpo da solicitação. Aqui está um exemplo de solicitaçãoURL.

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

Resposta do JSON

Quando for bem-sucedido, você receberá um código de 200 HTTP resposta. A JSON resposta a seguir foi truncada para maior clareza.

```
{
 "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

Este exemplo mostra como criar uma solicitação de pesquisa sobre o tipo de `DocumentReference` recurso para pacientes com diagnóstico de estreptococos e que também receberam prescrição de 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
```

Quando for bem-sucedido, você receberá a seguinte JSON resposta.

```
{
    "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

Use o seguinte para fazer uma GET solicitação sobre o tipo de Location recurso. Essa pesquisa encontra locais em seu armazenamento de HealthLake dados que contêm o nome da cidade Boston como parte do endereço.

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

Resposta do JSON

Quando for bem-sucedido, você receberá um código de 200 HTTP resposta. A JSON resposta foi truncada para maior clareza.


```
{
    "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

Use o seguinte para fazer uma solicitação de pesquisa GET baseada no tipo de `Observation` recurso. Essa pesquisa usa o parâmetro de `value-concept` pesquisa para procurar o código médico,`266919005`. Esse status indica`Never smoker`.

Você precisa especificar uma solicitação URL e uma sequência de caracteres de consulta. Aqui está um exemplo de solicitaçãoURL.

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

Para especificar o status`Never smoker`, defina `value-concept=266919005` como a string de consulta.

Resposta do JSON

Quando for bem-sucedido, você receberá um código de `200` HTTP resposta. A JSON resposta a seguir foi truncada para maior clareza.

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

Lendo o histórico FHIR de recursos

A FHIR `history` interação recupera o histórico de um FHIR recurso específico em um armazenamento de HealthLake dados. Usando essa interação, você pode determinar como o

conteúdo de um FHIR recurso mudou ao longo do tempo. Também é útil em coordenação com os registros de auditoria para ver o estado de um recurso antes e depois da modificação.

> **Note**
>
> FHIR`history`o recurso é ativado por padrão para todos os armazenamentos de HealthLake dados criados após 25/10/2024. Se seu armazenamento de dados foi criado antes dessa data, você pode enviar um ticket de suporte para que a FHIR `history` interação seja ativada. Crie um caso usando AWS Support Center Consoleo. Para criar seu caso, faça login no seu Conta da AWS e escolha Criar caso.

A `history` interação é realizada usando o `HTTP GET` comando. As FHIR interações `create`, `update`, e `delete` resultam em uma versão histórica do recurso a ser salva. HealthLake suporta os seguintes parâmetros de pesquisa para a FHIR `history` interação.

HealthLake parâmetros de pesquisa suportados para FHIR **`history`** interação

Parâmetro de pesquisa	Descrição
`_count : integer`	O número máximo de resultados de pesquisa em uma página. O servidor retornará o número solicitado ou o número máximo de resultado s de pesquisa permitidos por padrão para o armazenamento de dados, o que for menor.
`_since : instant`	Inclua somente versões de recursos que foram criadas em ou após um determinado instante no tempo.
`_at : date(Time)`	Inclua somente versões de recursos que estavam atualizadas em algum momento durante o período especificado no valor de data e hora. Para obter mais informações, consulte `date`a HL7FHIRRESTfulAPIdocumentação.

O exemplo a seguir retorna 100 resultados históricos de pesquisa por página para um FHIR `Patient` recurso em HealthLake. Para ver o URL caminho inteiro, role até o botão Copiar. O URL é do formato:

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

O conteúdo de retorno de uma interação histórica está contido em um FHIR recurso `Bundle`, com o tipo definido como`history`. Ele contém o histórico de versões especificado, classificado com as versões mais antigas por último, e inclui recursos excluídos. Para obter informações adicionais sobre a `history` interação, consulte `history`a HL7 FHIR RESTful API documentação.

> **Note**
>
> Você pode optar por não participar de `history` tipos FHIR de recursos específicos. Para optar por não participar, crie um caso usando AWS Support Center Console. Para criar seu caso, faça login no seu Conta da AWS e escolha Criar caso.

Lendo o histórico de recursos específico FHIR da versão

A FHIR `vread` interação realiza uma leitura específica da versão de um recurso em um armazenamento de HealthLake dados. Usando essa interação, você pode ver o conteúdo de um FHIR recurso como estava em um determinado momento no passado.

HealthLake declara que oferece suporte ao controle de versão `CapabilityStatement.rest.resource.versioning`para cada recurso suportado. Todos os armazenamentos de HealthLake dados incluem `Resource.meta.versionId` (`vid`) em todos os recursos.

Quando a FHIR `history` interação está ativada (por padrão para armazenamentos de dados criados após 25/10/2024 ou por solicitação para armazenamentos de dados mais antigos), a `Bundle` resposta inclui o `vid` como parte do. `location` No exemplo a seguir, o é `vid` exibido como o número1. Para ver o exemplo completo, consulte Exemplo de bundle/bundle-response (). JSON

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

A `vread` interação é realizada usando o `HTTP GET` comando. A `vread` interação a seguir retorna uma única instância com o conteúdo especificado para o FHIR `Patient` recurso para a versão dos metadados do recurso especificada pelo`vid`. Para ver o URL caminho inteiro no exemplo a seguir, role até o botão Copiar. O URL é do formato:

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

> **Note**
>
> Se você usar a `history` interação sem `vread` ao ler um FHIR recurso, HealthLake sempre retornará a versão mais recente dos metadados do recurso.

Para obter informações adicionais sobre a `vread` interação, consulte `vread`a APIdocumentação do HL7 FHIR Restful.

Obtendo dados do paciente com a operação Patient $everything FHIR REST API

A operação Patient $everything é usada para consultar um recurso FHIR do Paciente junto com quaisquer outros recursos relacionados a esse paciente. Essa operação pode ser usada para fornecer ao paciente acesso a todo o prontuário ou para que um provedor realize um download em massa de dados relacionados a um paciente. HealthLake suporta $everything para um ID de paciente específico.

> **Note**
>
> Atualmente, a operação Patient $everything é suportada em armazenamentos de dados criados após 27 de fevereiro de 2024.

Obtenha todos os recursos relacionados a um paciente

Patient $everything é uma REST API operação que pode ser invocada conforme mostrado nos exemplos abaixo.

GET Request

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

> **Note**
>
> Os recursos em resposta são classificados por tipo de recurso e ID do recurso.
> A resposta é sempre preenchida com Bundle.total.

Parâmetros do paciente $everything

HealthLake suporta os seguintes parâmetros de consulta

Parâmetro	Detalhes
rápido	Obtenha todos os dados do paciente após uma data de início especificada.
end	Obtenha todos os dados do paciente antes de uma data de término especificada.
since	Atualize todos os dados do paciente após uma data especificada.
_tipo	Obtenha dados de pacientes para tipos de recursos específicos.
_contar	Obtenha dados do paciente e especifique o tamanho da página.

Example - Obtenha todos os dados do paciente após uma data de início especificada

O paciente $everything pode usar o `start` filtro para consultar dados somente após uma data 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 - Obtenha todos os dados do paciente antes de uma data de término especificada

O paciente $everything pode usar o `end` filtro para consultar apenas dados antes de uma data 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 - Atualize todos os dados do paciente após uma data especificada

O paciente $everything pode usar o `since` filtro para consultar somente dados atualizados após uma data 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 - Obtenha dados de pacientes para tipos de recursos específicos

O paciente $everything pode usar o `_type` filtro para especificar tipos de recursos específicos a serem incluídos na resposta. Vários tipos de recursos podem ser especificados em uma lista separada por vírgulas.

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

Example - Obtenha dados do paciente e especifique o tamanho da página

O paciente $everything pode usar o `_count` para definir o tamanho da página.

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

Paciente $tudo **`start`** e atributos **`end`**

HealthLake suporta os seguintes atributos de recursos para os parâmetros de início e término da consulta.

Recurso	Elemento de recursos
Conta	Conta. servicePeriod.iniciar
AdverseEvent	AdverseEvent.data
AllergyIntolerance	AllergyIntolerance.recordedDate
Nomeação	Marcação. Início
AppointmentResponse	AppointmentResponse.iniciar
AuditEvent	AuditEvent.periodo.start
Basic	Básico. Criado
BodyStructure	NÃO_ DATE
CarePlan	CarePlan.periodo.start
CareTeam	CareTeam.periodo.start
ChargeItem	ChargeItem. occurrenceDateTime, ChargeItem. occurrencePeriod.começar, ChargeItem. occurrenceTiming.evento
Reivindicar	Reclamação. billablePeriod.iniciar
ClaimResponse	ClaimResponse.criado
ClinicalImpression	ClinicalImpression.data
Comunicação	Comunicação. Enviada

Recurso	Elemento de recursos
CommunicationRequest	CommunicationRequest. occurrenceDateTime, CommunicationRequest. occurrencePeriod.iniciar
Composição	Composição.Data
Condição	Condição. recordedDate
Consentimento	Consentimento. dateTime
Cobertura	Cobertura. Período inicial
CoverageEligibilityRequest	CoverageEligibilityRequest.criado
CoverageEligibilityResponse	CoverageEligibilityResponse.criado
DetectedIssue	DetectedIssue.identificado
DeviceRequest	DeviceRequest.authoredOn
DeviceUseStatement	DeviceUseStatement.recordedOn
DiagnosticReport	DiagnosticReport.eficaz
DocumentManifest	DocumentManifest.criado

Recurso	Elemento de recursos
DocumentReference	DocumentReference.contexto.period.start
Encontro	Encounter.period.start
EnrollmentRequest	EnrollmentRequest.criado
EpisodeOfCare	EpisodeOfCare.periodo.start
ExplanationOfBenefit	ExplanationOfBenefit. billablePeriod.iniciar
FamilyMemberHistory	NÃO_ DATE
Sinalizador	FLAG.PERIOD.START
Objetivo	Objetivo. statusDate
Grupo	NÃO_ DATE
ImagingStudy	ImagingStudy.iniciado
Imunização	Imunização. Registrada
ImmunizationEvaluation	ImmunizationEvaluation.data
ImmunizationRecommendation	ImmunizationRecommendation.data
Fatura	Data da fatura

Recurso	Elemento de recursos
Listar	Data da lista
MeasureReport	MeasureReport.periodo.start
Mídia	Mídia. Emitido
MedicationAdministration	MedicationAdministration.eficaz
MedicationDispense	MedicationDispense.whenPrepared
MedicationRequest	MedicationRequest.authoredOn
MedicationStatement	MedicationStatement.dateAsserted
Molecular Sequence	NÃO_ DATE
Nutrition Order	NutritionOrder.dateTime
Observação	Observação. Eficaz
Paciente	NÃO_ DATE
Pessoa	NÃO_ DATE
Procedimento	Procedimento. Executado
Proveniência	Proveniência. occurredPeriod.start, Proveniência. occurredDateTime

Recurso	Elemento de recursos
QuestionnaireResponse	QuestionnaireResponse.de autoria
RelatedPerson	NÃO_ DATE
RequestGroup	RequestGroup.authoredOn
ResearchSubject	ResearchSubject.período
RiskAssessment	RiskAssessment. occurrenceDateTime, RiskAssessment. occurrencePeriod.iniciar
Schedule	Cronograma. planningHorizon
ServiceRequest	ServiceRequest.authoredOn
Espécime	Espécime. receivedTime
SupplyDelivery	SupplyDelivery. occurrenceDateTime, SupplyDelivery. occurrencePeriod.começar, SupplyDelivery. occurrenceTiming.evento
SupplyRequest	SupplyRequest.authoredOn
VisionPrescription	VisionPrescription.dateWritten

Exportando dados do seu armazenamento de HealthLake dados usando $export

Para fazer uma solicitação de exportação usando a FHIR REST API especificação `$export` como parte da `POST` solicitação e incluir parâmetros de solicitação no corpo da solicitação. De acordo com a FHIR especificação, o FHIR servidor deve oferecer suporte `GET` às solicitações e pode oferecer suporte `POST` às solicitações. Para oferecer suporte a parâmetros adicionais, é necessário um corpo para iniciar a exportação e, portanto, HealthLake oferece suporte POST às solicitações.

> **Important**
>
> HealthLake os armazenamentos de dados criados antes de 1º de junho de 2023 suportam somente solicitações de trabalho de exportação FHIR REST API baseadas em exportações em todo o sistema.
> HealthLake os armazenamentos de dados criados antes de 1º de junho de 2023 não oferecem suporte para obter o status de uma exportação usando uma `GET` solicitação no endpoint de um armazenamento de dados.

Todas as solicitações de exportação que você faz usando o FHIR REST API são devolvidas em `ndjson` formato e exportadas para um bucket do Amazon S3. Cada objeto do S3 conterá somente um único tipo FHIR de recurso.

Você pode fazer uma única solicitação de exportação para cada AWS conta por vez. Para saber mais sobre as Cotas de Serviço associadas a HealthLake, consulte. AWS HealthLake endpoints e cotas

Para saber mais sobre como fazer uma solicitação de exportação usando o. FHIR REST API Exportação de dados do seu armazenamento HealthLake de dados com operações FHIR REST API

Consulte armazenamentos AWS HealthLake de dados usando SQL no Amazon Athena

Quando você cria um armazenamento de HealthLake dados, a estrutura de FHIR dados altamente aninhada é ingerida no Amazon Athena e automaticamente transformada em tabelas Iceberg que podem ser consultadas com. SQL A concessão de acesso a esse novo recurso é gerenciada usando o AWS Lake Formation. Cada tipo de FHIR recurso é representado como uma tabela individual no Athena.

Important

Para armazenamentos de dados criados antes de 14 de novembro de 2022, você deve migrar seu armazenamento de dados existente para um novo para consultá-lo usandoSQL. Para obter ajuda, consulte Migração de um armazenamento de dados existente para usar o Amazon Athena.

Note

Depois de 20 de fevereiro de 2023, HealthLake os armazenamentos de dados não usam processamento de linguagem natural integrado (NLP) por padrão. Se você estiver interessado em ativar esse recurso em seu armazenamento de dados, consulte o Como faço para ativar o recurso integrado HealthLake de processamento de linguagem natural do? capítulo Solução de problemas.

Para criar um armazenamento de HealthLake dados, você deve adicionar IAM políticas adicionais e uma função de serviço ao seu IAM usuário ou função de HealthLake administrador. Para obter mais informações sobre a configuração de permissões, consulteConfigurando permissões para começar a usar AWS HealthLake.

HealthLake os armazenamentos de dados são inseridos no Athena como tabelas Iceberg. Para saber mais sobre como as tabelas Iceberg funcionam no Athena, consulte Como usar tabelas Iceberg no Guia do usuário do Athena.

HealthLake suporta `READ` as operações de seus armazenamentos de HealthLake dados e armazenamentos de dados no Athena. Para saber mais sobre as operações Criar, Ler, Atualizar e

Excluir (CRUD) usando as FHIR REST API operações, consulte Usando FHIR REST API interações com um armazenamento HealthLake de dados para saber mais sobre como CRUD as operações afetam seus dados no Athena.

Os tópicos deste capítulo descrevem como conectar seu armazenamento de HealthLake dados ao Athena, como consultá-lo usando SQL e como conectar os resultados a outros AWS serviços para análise posterior.

Sumário

Conectando seu armazenamento de dados ao Amazon Athena

> **Important**
>
> Depois de 14 de novembro de 2022, os IAM requisitos de acesso HealthLake foram alterados. Para criar armazenamentos de dados e conceder acesso a eles no Athena, você deve ter a política `AWSLakeFormationDataAdmin` gerenciada adicionada ao seu IAM usuário, grupo ou função. Você pode usar a `AWSLakeFormationDataAdmin` política para criar administradores de data lake e conceder acesso aos armazenamentos de dados no Athena.

Este tópico descreve as etapas necessárias para criar um usuário, grupo ou função do Athena e conceder a eles acesso FHIR aos recursos encontrados em HealthLake um armazenamento de dados.

- Conceder a um usuário, grupo ou função acesso a um armazenamento de HealthLake dados (AWS Lake Formation Console)
- Configurando uma conta Athena

Conceder a um usuário, grupo ou função acesso a um armazenamento de HealthLake dados (AWS Lake Formation Console)

> Pessoa: administrador HealthLake
>
> A pessoa HealthLake administradora é uma administradora de data lake em AWS Lake Formation. Eles concedem acesso aos armazenamentos de HealthLake dados em Lake Formation.

Para cada armazenamento de dados criado, há duas entradas visíveis no console do AWS Lake Formation. Uma entrada é um link de recurso. Os nomes dos links de recursos são sempre exibidos em itálico. Cada link de recurso é exibido com o nome e o proprietário do recurso compartilhado vinculado. Para todos os armazenamentos de HealthLake dados, o proprietário do recurso compartilhado é a conta HealthLake de serviço. A outra entrada é o armazenamento HealthLake de dados na conta HealthLake de serviço. As etapas desse procedimento usam o armazenamento de dados que é o link do recurso.

Para saber mais sobre links de recursos, consulte Como os links de recursos funcionam no Lake Formation no AWS Lake Formation Developer Guide.

Para que um usuário, grupo ou função possa consultar dados no Athena, você deve conceder a permissão Descreve no banco de dados de recursos. Em seguida, você deve conceder Selecionar e Descrever nas tabelas.

STEP1: Para conceder DESCRIBEpermissões em um banco de HealthLake dados de links de recursos do armazenamento de dados

1. Abra o console do AWS Lake Formation: https://console.aws.amazon.com/lakeformation/
2. Na barra de navegação principal, escolha Bancos de dados.
3. Na página Bancos de dados, escolha o botão de rádio ao lado do nome do armazenamento de dados que está em itálico.
4. Escolha Ações (▾).
5. Selecione Conceder.
6. Na página Conceder permissões de dados, em Diretores, escolha IAMusuários ou funções.
7. Em IAMusuários ou funções, use a seta para baixo (▾) ou pesquise o IAM usuário, a função ou o grupo sobre o qual você deseja poder fazer consultas no Athena.

8. Em LF-Tags ou cartão de recursos do catálogo, escolha a opção Recursos do catálogo de dados nomeados.
9. Em Bancos de dados, use a seta para baixo (▼) para escolher o banco de HealthLake dados do armazenamento de dados ao qual você deseja compartilhar o acesso.
10. No cartão Permissões do link do recurso, em Permissões do link do recurso, escolha Descrever.

Quando a concessão é bem-sucedida, o banner Conceder permissão de sucesso é exibido. Para ver a permissão que você acabou de conceder, escolha Permissões do Data lake. Encontre o usuário, o grupo e a função na tabela. Na coluna Permissões, você verá a lista Descrever.

Agora você deve usar Grant on target para conceder Selecionar e Descrever em todas as tabelas no banco de dados.

STEP2: Conceder acesso a todas as tabelas em um link de recurso do armazenamento de HealthLake dados

1. Abra o console do AWS Lake Formation: https://console.aws.amazon.com/lakeformation/
2. Na barra de navegação principal, escolha Bancos de dados.
3. Na página Bancos de dados, escolha o botão de rádio ao lado do nome do armazenamento de dados que está em itálico.
4. Escolha Ações (▼).
5. Escolha Grant no alvo.
6. Na página Conceder permissões de dados, em Diretores, escolha IAMusuários ou funções.
7. Em IAMusuários ou funções, use a seta para baixo (▼) ou pesquise o IAM usuário, grupo ou função sobre o qual você deseja poder fazer consultas no Athena.
8. Em LF-Tags ou cartão de recursos do catálogo, escolha a opção Recursos do catálogo de dados nomeados.
9. Em Bancos de dados, use a seta para baixo (▼) para escolher o banco de HealthLake dados do armazenamento de dados ao qual você deseja conceder acesso.
10. Em Tabelas, escolha Todas as tabelas para compartilhar todas as tabelas com um HealthLake usuário.
11. No cartão Permissões da tabela, em Permissões da tabela, escolha Descrever e selecionar.
12. Selecione Conceder.

Depois de escolher conceder, um banner de sucesso de permissões de concessão é exibido. O usuário especificado agora pode fazer consultas em um armazenamento de HealthLake dados no Athena.

Começando com Athena

> **HealthLake usuário**
>
> O HealthLake usuário usará o console do Athena ou AWS SDKs para consultar um armazenamento de HealthLake dados compartilhado com ele pelo HealthLake administrador. AWS CLI

Para consultar um armazenamento de dados usando o Athena, você deve fazer as três coisas a seguir.

- Conceda ao IAM usuário ou à função acesso ao armazenamento de HealthLake dados por meio do Lake Formation. Para saber mais, consulte Conceder a um usuário, grupo ou função acesso a um armazenamento de HealthLake dados (AWS Lake Formation Console).
- Crie um grupo de trabalho para seu armazenamento HealthLake de dados.
- Designe um bucket do Amazon S3 para armazenar os resultados da consulta.

Para começar a usar o Athena, adicione as políticas FullAccess AWS gerenciadas AmazonAthenaFullAccesse o AmazonS3 ao seu usuário, grupo ou função. Usar uma política AWS gerenciada é uma ótima maneira de começar a usar um novo serviço. Lembre-se de que as políticas gerenciadas pela AWS podem não conceder permissões de privilégio mínimo para seus casos de uso específicos porque estão disponíveis para uso por todos os clientes da AWS. Ao definir permissões com IAM políticas, conceda somente as permissões necessárias para realizar uma tarefa. Para saber mais IAM e aplicar o privilégio mínimo, consulte Aplicar permissões de privilégios mínimos no Guia do usuário. IAM

> **Important**
>
> Para consultar um armazenamento HealthLake de dados no Athena, você deve usar o mecanismo Athena versão 3.

Grupos de trabalho são recursos e, portanto, você pode usar políticas IAM baseadas para controlar o acesso a grupos de trabalho específicos. Para saber mais, consulte Como usar grupos de trabalho para controlar o acesso e os custos das consultas no Guia do usuário do Athena.

Para saber mais sobre como configurar grupos de trabalho, consulte o Guia do https://docs.aws.amazon.com/athena/latest/ug/workgroups-procedure.html usuário do Athena.

> **Note**
>
> A região em que seu bucket do Amazon S3 está e o console do Athena devem corresponder.

Antes de executar uma consulta, um local de bucket de resultados de consultas do Amazon S3 precisa ser especificado, ou você deve usar um grupo de trabalho que especificou um bucket e com uma configuração que substitui as configurações do cliente. Os arquivos de saída são salvos automaticamente para cada consulta executada.

Para obter mais detalhes sobre a especificação dos locais dos resultados da consulta no console do Athena, consulte Especificação de um local do resultado da consulta usando o console do Athena no Guia do usuário do Amazon Athena.

Para ver exemplos de como consultar seu armazenamento de HealthLake dados no Athena, consulte. Consulte seu armazenamento HealthLake de dados usando SQL

Consulte seu armazenamento HealthLake de dados usando SQL

> **Note**
>
> Depois de 20 de fevereiro de 2023, HealthLake os armazenamentos de dados não usam processamento de linguagem natural integrado (NLP) por padrão. Se você estiver interessado em ativar esse recurso em seu armazenamento de dados, consulte o Como faço para ativar o recurso integrado HealthLake de processamento de linguagem natural do? capítulo Solução de problemas.
>
> Todos os exemplos neste tópico usam dados fictícios criados usando o Synthea. Para saber mais sobre como criar um armazenamento de dados pré-carregado com dados Synthea, consulte. Criando um armazenamento de dados em AWS HealthLake

Quando você importa seu armazenamento de HealthLake dados para o Athena, cada tipo de recurso do seu armazenamento de HealthLake dados é convertido em uma tabela. Essas tabelas podem ser consultadas individualmente ou em grupo usando consultas SQL baseadas. Devido à estrutura dos armazenamentos de dados, seus dados são importados para o Athena como vários tipos de dados diferentes. Para saber mais sobre a criação de SQL consultas que podem acessar esses tipos de dados, consulte Como consultar matrizes com tipos complexos e estruturas aninhadas no Guia do usuário do Amazon Athena.

Para cada elemento em um tipo de recurso, a FHIR especificação define uma cardinalidade. A cardinalidade de um elemento define os limites inferior e superior de quantas vezes esse elemento pode aparecer. Ao criar uma SQL consulta, você deve levar isso em consideração. Por exemplo, vamos ver alguns elementos em Tipo de recurso: Paciente.

- Elemento: Nome A FHIR especificação define a cardinalidade como. `0..*`

  O elemento é capturado como uma 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
}]
```

  No Athena, para ver como um tipo de recurso foi ingerido, procure-o em Tabelas e visualizações. Para acessar elementos nessa matriz, você pode usar a notação de pontos. Aqui está um exemplo simples que acessaria os valores de `given` `family` e.

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

```
    name[1].family as LastName
FROM Patient
```

- Elemento: MaritalStatus A FHIR especificação define a cardinalidade como. `0..1`

  Esse elemento é capturado 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
}
```

  No Athena, para ver como um tipo de recurso foi ingerido, procure-o em Tabelas e visualizações. Para acessar pares de valores-chave noJSON, você pode usar a notação de pontos. Como não é uma matriz, nenhum índice de matriz é necessário. Aqui está um exemplo simples que acessaria o valor `detext`.

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

Para saber mais sobre como acessar e pesquisarJSON, consulte Consultar JSON no Guia do usuário do Athena.

As instruções de consulta da Athena Data Manipulation Language (DML) são baseadas no Trino. O Athena não suporta todos os recursos do Trino e há diferenças significativas. Para saber mais, consulte DMLconsultas, funções e operadores no Guia do usuário do Amazon Athena.

Além disso, o Athena oferece suporte a vários tipos de dados que você pode encontrar ao criar consultas em seu HealthLake armazenamento de dados. Para saber mais sobre os tipos de dados no Athena, consulte Tipos de dados no Amazon Athena no Guia do usuário do Amazon Athena.

Para saber mais sobre como SQL as consultas funcionam no Athena, SQLconsulte a referência do Amazon Athena no Guia do usuário do Amazon Athena.

Cada guia mostra exemplos de como pesquisar os tipos de recursos especificados e os elementos associados usando o Athena.

Element: Extension

O elemento `extension` é usado para criar campos personalizados em um armazenamento de dados.

Este exemplo mostra como acessar os recursos do `extension` elemento encontrado no tipo de `Patient` recurso.

Quando seu armazenamento HealthLake de dados é importado para o Athena, os elementos de um tipo de recurso são analisados de forma diferente. Como a estrutura do `element` é variável, ela não pode ser totalmente especificada no esquema. Para lidar com essa variabilidade, os elementos dentro da matriz são passados como strings.

Na descrição da tabela de`Patient`, você pode ver o elemento `extension` descrito como`array<string>`, o que significa que você pode acessar os elementos da matriz usando um valor de índice. Para acessar os elementos da string, no entanto, você deve usar`json_extract`.

Aqui está uma única entrada do `extension` elemento encontrado na tabela 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"
 }
]
```

Mesmo que isso seja válidoJSON, Athena o trata como uma string.

Este exemplo de SQL consulta demonstra como você pode criar uma tabela que contém os `patient-birthPlace` elementos `patient-mothersMaidenName` e. Para acessar esses elementos, você precisa usar diferentes índices de matriz e `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 saber mais sobre as consultas que envolvemJSON, consulte Extração de dados JSON no Guia do usuário do Amazon Athena.

Element: birthDate (Age)

A idade não é um elemento do tipo de recurso do paciente emFHIR. Aqui estão dois exemplos de pesquisas que filtram com base na idade.

Como a idade não é um elemento, usamos o `birthDate` para SQL as consultas. Para ver como um elemento foi ingeridoFHIR, pesquise o nome da tabela em Tabelas e visualizações. Você pode ver que é do tipo string.

Exemplo 1: Calculando um valor para a idade

Neste exemplo de SQL consulta, usamos uma SQL ferramenta integrada `year` para extrair esses componentes. `current_date` Em seguida, nós os subtraímos para retornar a idade real do paciente como uma coluna chamada`age`.

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

Exemplo 2: Filtragem para pacientes que nasceram antes `2019-01-01` e são`male`.

A SQL consulta mostra como usar a `CAST` função para converter o `birthDate` elemento como tipo `DATE` e como filtrar com base em dois critérios na `WHERE` cláusula. Como o elemento é ingerido como string de tipo por padrão, devemos `CAST` usá-lo como tipo`DATE`. Em seguida, você pode usar o < operador para compará-lo com uma data diferente,`2019-01-01`. Ao usar`AND`, você pode adicionar um segundo critério à `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

Este exemplo mostra pesquisas por locais dentro do tipo de recurso Localização em que o nome da cidade é 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

A condição do tipo de recurso armazena dados de diagnóstico relacionados a problemas que atingiram um nível de preocupação. HealthLakeO processamento de linguagem natural médica integrado (NLP) gera novos `Condition` recursos com base nos detalhes encontrados no tipo de DocumentReference recurso. Quando um novo recurso é gerado, HealthLake anexa a tag `SYSTEM_GENERATED` ao `meta` elemento. Esse exemplo de SQL consulta demonstra como você pode pesquisar a tabela de condições e retornar os resultados em que os `SYSTEM_GENERATED` resultados foram removidos.

Para saber mais sobre HealthLake o processamento integrado de linguagem natural (NLP), consulteUsando a geração automatizada de recursos com base no processamento de linguagem natural (NLP) do tipo de FHIR DocumentReference recurso em AWS HealthLake.

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

Você também pode pesquisar em um elemento de string especificado para filtrar ainda mais sua consulta. O `modifierextension` elemento contém detalhes sobre qual `DocumentReference` recurso foi usado para gerar um conjunto de condições. Novamente, você deve usar `json_extract` para acessar os JSON elementos aninhados que são trazidos para o Athena como uma string.

Esse exemplo de SQL consulta demonstra como você pode pesquisar tudo o `Condition` que foi gerado com base em um específico`DocumentReference`. Use `CAST` para definir o JSON elemento como uma string para que você possa usá-lo `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

O tipo de recurso, Observação, armazena medidas e afirmações simples feitas sobre um paciente, dispositivo ou outro assunto. HealthLakeO processamento de linguagem natural integrado (NLP) gera novos `Observation` recursos com base nos detalhes encontrados em um `DocumentReference` recurso. Esse exemplo de SQL consulta inclui `WHERE meta.tag[1] is NULL` comentários, o que significa que os `SYSTEM_GENERATED` resultados estão incluídos.

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

Essa coluna foi importada como uma `struct`. Portanto, você pode acessar elementos dentro dele usando a notação de pontos.

Resource type: MedicationStatement

MedicationStatement é um tipo de FHIR recurso que você pode usar para armazenar detalhes sobre medicamentos que um paciente tomou, está tomando ou tomará no futuro. HealthLakeO processamento de linguagem natural médica integrado (NLP) gera novos MedicationStatement recursos com base em documentos encontrados no tipo de DocumentReference recurso. Quando novos recursos são gerados, HealthLake anexa a tag `SYSTEM_GENERATED` ao `meta` elemento. Esse exemplo de SQL consulta demonstra como criar uma consulta que filtra com base em um único paciente usando seu identificador e encontra recursos que foram adicionados pelo HealthLake's integratedNLP.

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

Para saber mais sobre HealthLake a medicina integradaNLP, consulteUsando a geração automatizada de recursos com base no processamento de linguagem natural (NLP) do tipo de FHIR DocumentReference recurso em AWS HealthLake.

Exemplos de SQL consultas com filtragem complexa

> **Note**
>
> Depois de 20 de fevereiro de 2023, HealthLake os armazenamentos de dados não usam processamento de linguagem natural integrado (NLP) por padrão. Se você estiver interessado em ativar esse recurso em seu armazenamento de dados, consulte o Como faço para ativar o recurso integrado HealthLake de processamento de linguagem natural do? capítulo Solução de problemas.

Os exemplos neste tópico incluem SQL consultas de integração com o HealthLake Athena que usam filtragem complexa.

Example Criação de critérios de filtragem baseados em dados demográficos

Identificar os dados demográficos corretos do paciente é importante ao criar uma coorte de pacientes. Esse exemplo de consulta demonstra como você pode usar a notação de pontos Trino e `json_extract` filtrar dados em seu HealthLake armazenamento de dados.

```
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;
```

Com o console Athena, você pode classificar e baixar ainda mais os resultados.

Example Criação de filtros para um paciente e suas condições relacionadas

Esse exemplo de consulta demonstra como você pode encontrar e classificar todas as condições relacionadas aos pacientes encontrados em um armazenamento de HealthLake dados.

```
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;
```

Você pode usar o console do Athena para classificar melhor esses resultados ou baixá-los para análise posterior.

Example Criação de filtros para pacientes e suas observações relacionadas

Esse exemplo de consulta demonstra como você pode encontrar e classificar todas as observações relacionadas aos pacientes encontrados em um armazenamento de HealthLake dados.

```
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 Criação de condições de filtragem para um paciente e seus procedimentos relacionados

Conectar procedimentos aos pacientes é um aspecto importante da assistência médica. Essa SQL consulta demonstra como você pode usar os tipos de recursos do paciente e do procedimento para fazer isso no Athena. Essa SQL consulta retornará todos os pacientes e seus procedimentos relacionados encontrados em seu armazenamento HealthLake de dados.

```
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;
```

Agora você pode usar o console do Athena para baixar os resultados para análise posterior ou classificá-los para entender melhor os resultados.

Example Criação de condições de filtragem para um paciente e suas prescrições relacionadas

É importante ver uma lista atual de medicamentos que os pacientes estão tomando. Usando o Athena, você pode escrever uma SQL consulta que usa os tipos de paciente e de MedicationRequest recurso encontrados em seu armazenamento de HealthLake dados.

Essa SQL consulta une o paciente e as MedicationRequest tabelas importadas para o Athena. Ele também organiza as prescrições em suas entradas individuais usando a notação de pontos.

```
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
```

Você pode usar o console do Athena para classificar os resultados ou baixá-los para análise posterior.

Example Ver medicamentos encontrados no tipo MedicationStatement de recurso

O exemplo de consulta mostra como organizar o aninhado JSON importado para o SQL Athena usando. A consulta usa o `meta` elemento para indicar quando um medicamento foi adicionado pelo processamento integrado HealthLake de linguagem natural (NLP). Para saber mais sobre a integração com HealthLake o Amazon Comprehend Medical, consulte. Usando a geração automatizada de recursos com base no processamento de linguagem natural (NLP) do tipo de FHIR DocumentReference recurso em AWS HealthLake Ele também é usado `json_extract` para pesquisar dados dentro da matriz de JSON strings.

```
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;
```

Você pode usar o console do Athena para baixar esses resultados ou classificá-los.

Example Filtro para um tipo específico de doença

O exemplo mostra como você pode encontrar um grupo de pacientes, com idades entre 18 e 75 anos, que foram diagnosticados com 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%');
```

Agora você pode usar o console do Athena para classificar os resultados ou baixá-los para análise posterior.

AWS HealthLake e VPC endpoints de interface ()AWS PrivateLink

Você pode estabelecer uma conexão privada entre você VPC e AWS HealthLake criando um VPCendpoint de interface. Os VPC endpoints de interface são alimentados por AWS PrivateLinkuma tecnologia que você pode usar para acessar de forma privada HealthLake; APIs sem um gateway de internet, NAT dispositivo, VPN conexão ou AWS Direct Connect conexão. As instâncias em seu VPC não precisam de endereços IP públicos para se comunicar com HealthLake;APIs. O tráfego entre você VPC e HealthLake; não sai da rede Amazon.

Cada endpoint de interface é representado por uma ou mais Interfaces de Rede Elástica nas sub-redes.

Para obter mais informações, consulte VPCEndpoints de interface (AWS PrivateLink) no Guia do VPC usuário da Amazon.

Considerações sobre endpoints HealthLake VPC

Antes de configurar um VPC endpoint de interface para HealthLake, certifique-se de revisar as propriedades e limitações do endpoint de interface no Guia VPC do usuário da Amazon.

HealthLake suporta a realização de chamadas para todas as suas API ações a partir do seuVPC.

Criação de um VPC endpoint de interface para HealthLake;

Você pode criar um VPC endpoint para o serviço HealthLake; usando o VPC console da Amazon ou o AWS Command Line Interface (AWS CLI). Para obter mais informações, consulte Criação de um endpoint de interface no Guia do VPC usuário da Amazon.

Crie um VPC endpoint para HealthLake; usando o seguinte nome de serviço:

- com.amazonaws. *region*.lago de saúde

Se você ativar o modo privado DNS para o endpoint, poderá fazer API solicitações para HealthLake usar seu DNS nome padrão para a região. Por exemplo, *healthlake.us-east-1.amazonaws.com*.

Para obter mais informações, consulte Acessando um serviço por meio de um endpoint de interface no Guia do VPC usuário da Amazon.

Criação de uma política VPC de endpoint para HealthLake

Você pode anexar uma política de endpoint ao seu VPC endpoint que controla o acesso a. HealthLake Essa política especifica as seguintes informações:

- A entidade principal que pode realizar ações.
- As ações que podem ser realizadas.
- Os recursos aos quais as ações podem ser aplicadas.

Para obter mais informações, consulte Controle de acesso a serviços com VPC endpoints no Guia do VPC usuário da Amazon.

Exemplo: política VPC de endpoint para HealthLake ações

Veja a seguir um exemplo de uma política de endpoint para HealthLake. Quando anexada a um endpoint, essa política concede acesso à HealthLake `CreateFHIRDatastore` ação para todos os diretores em todos os recursos.

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

Marcando recursos em AWS HealthLake

Você pode atribuir metadados aos seus recursos da AWS na forma de tags. Cada tag é um rótulo que consiste em um valor e uma chave definida pelo usuário. As tags podem ajudar você a gerenciar, identificar, organizar, pesquisar e filtrar recursos.

Este tópico descreve as categorias de marcação normalmente usadas e as estratégias para ajudar você a implementar uma estratégia de marcação eficiente e consistente. As seções a seguir pressupõem conhecimento básico de AWS recursos, marcação, faturamento detalhado e AWS Identity and Access Management ()IAM.

Cada tag tem duas partes:

- Uma chave de tag (por exemplo CostCenter, Ambiente ou Projeto). Chaves de tag fazem distinção entre maiúsculas e minúsculas.
- Um valor de tag (por exemplo, 111122223333 ou Produção). Como chaves de tag, os valores das tags diferenciam maiúsculas de minúsculas.

Você pode usar tags para categorizar recursos por finalidade, proprietário, ambiente ou outros critérios. Para obter mais informações, consulte Estratégias de marcação da AWS.

Você pode adicionar, alterar ou remover tags, um recurso por vez, do console de serviço, do serviço API ou do AWSCLI.

Para ativar a marcação, certifique-se de que TagResources estão autorizados. Você pode autorizar TagResources anexando uma IAM política como no exemplo a seguir.

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

```
        }
    ]
}
```

Aviso importante

AWS HealthLake protege os dados do cliente de acordo com as políticas do Modelo de Responsabilidade AWS Compartilhada. Isso significa que todos os dados do cliente são criptografados em transição e em repouso. No entanto, nem todos os nomes inseridos pelo cliente para armazenamentos de dados ou operações baseadas em tarefas são criptografados. Eles nunca devem conter informações de identificação pessoal ou informações de saúde protegidas. Para obter mais informações, consulte o capítulo AWS HealthLake Segurança.

Práticas recomendadas

À medida que você cria uma estratégia de marcação para recursos da AWS, siga as melhores práticas:

- Não armazene Informações de Identificação Pessoal (PII), Informações Pessoais de Saúde (PHI) ou outras informações confidenciais em tags.
- Use um formato padronizado que diferencia maiúsculas de minúsculas para tags e aplique-o de forma consistente a todos os tipos de recursos.
- Considere as diretrizes de tags que oferecem suporte a diversas finalidades, como gerenciar o controle de acesso a recursos, o rastreamento de custos, a automação e a organização.
- Use ferramentas automatizadas para ajudar a gerenciar as tags de recursos. AWS Resource Groups e Resource Groups Tagging API permitem o controle programático de tags, possibilitando gerenciar, pesquisar e filtrar automaticamente tags e recursos.
- A marcação é mais eficaz quando você usa mais tags.
- As tags podem ser editadas ou modificadas conforme as necessidades do usuário mudam. No entanto, para atualizar as tags de controle de acesso, você também deve atualizar as políticas que fazem referência a essas tags para controlar o acesso aos seus recursos.

Requisitos de marcação

As tags têm os seguintes requisitos:

- As chaves não podem ser prefixadas com aws:.
- As chaves devem ser exclusivas por conjunto de tags.
- Uma chave deve ter entre 1 e 128 caracteres permitidos.
- Um valor deve ter entre 0 e 256 caracteres permitidos.
- Os valores não precisam ser exclusivos por conjunto de tags.
- Os caracteres permitidos para chaves e valores são letras Unicode, dígitos, espaço em branco e qualquer um dos seguintes símbolos: _ . : / = + - @.
- As chaves e os valores diferenciam letras maiúsculas de minúsculas.

Adicionando uma tag a um armazenamento de dados

Adicionar tags a um armazenamento de dados pode ajudá-lo a identificar e organizar seus AWS recursos e gerenciar o acesso a eles. Primeiro, você adiciona uma ou mais tags (pares de valores-chave) a um armazenamento de dados. Você pode usar até cinquenta tags por usuário. Também há restrições quanto aos caracteres que você pode usar nos campos de chave e valor.

Depois de ter as tags, você pode criar IAM políticas para gerenciar o acesso ao armazenamento de dados com base nessas tags. Você pode usar o HealthLake console ou o AWS CLI para adicionar tags a um armazenamento de dados. Adicionar tags a um repositório pode afetar o acesso ele. Antes de adicionar uma tag a um armazenamento de dados, revise todas IAM as políticas que possam usar tags para controlar o acesso a recursos, como armazenamentos de dados.

Siga estas etapas para usar o AWS CLI para adicionar uma tag a um armazenamento HealthLake de dados. Para adicionar uma tag a um armazenamento de dados ao criá-lo, consulteCriando um armazenamento de dados em AWS HealthLake.

No terminal ou na linha de comando, execute o comando tag-resource, especificando o Amazon Resource Name (ARN) do armazenamento de dados ao qual você deseja adicionar tags e a chave e o valor da tag que você deseja adicionar. Você pode adicionar mais de uma tag a um armazenamento de dados. Também há restrições quanto aos caracteres que você pode usar nos campos de chave e valor, conforme listado em Requisitos de marcação Por exemplo, para adicionar tags a um armazenamento de dados enquanto ele está sendo criado, você usaria o seguinte comando no AWS CLI. O nome do armazenamento de dados é Test_Data_Store, e as duas tags adicionadas com chaves são key1 e key2 com valores como valor1 e 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 adicionar tags a um armazenamento de dados existente, você executaria o seguinte exemplo de comando:

```
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
```

Se for bem-sucedido, esse comando não retornará nenhuma resposta.

Listando tags para um armazenamento de dados

Siga estas etapas para usar o AWS CLI para visualizar uma lista das AWS tags de um armazenamento HealthLake de dados. Se não foram adicionadas tags, a lista retornará vazia.

No terminal ou na linha de comando, execute o list-tags-for-resource comando conforme mostrado no exemplo a seguir.

```
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"
    }
}
```

Removendo tags de um armazenamento de dados

Você pode remover uma ou mais tags associadas a um armazenamento de dados. A exclusão de uma tag não exclui a tag de outros recursos da AWS associados a essa tag.

No terminal ou na linha de comando, execute o comando untag-resource, especificando o Amazon Resource Name ARN () do armazenamento de dados do qual você deseja remover as tags e a chave da tag que você deseja remover.

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

Se for bem-sucedido, esse comando não retornará uma resposta. Para verificar as tags associadas ao armazenamento de dados, execute o list-tags-for-resource comando.

Monitoramento HealthLake

O monitoramento é uma parte importante da manutenção da confiabilidade, disponibilidade e desempenho de HealthLake suas outras AWS soluções. AWSfornece as seguintes ferramentas de monitoramento para observar HealthLake, relatar quando algo está errado e realizar ações automáticas quando apropriado:

- A Amazon CloudWatch monitora seus AWS recursos e os aplicativos em que você executa AWS em tempo real. Você pode coletar e monitorar métricas, criar painéis personalizados e definir alarmes que o notificam ou tomam medidas quando uma métrica específica atinge um limite específico. Por exemplo, você pode CloudWatch rastrear o CPU uso ou outras métricas de suas EC2 instâncias da Amazon e iniciar automaticamente novas instâncias quando necessário. Para obter mais informações, consulte o Guia CloudWatch do usuário da Amazon.
- AWS CloudTrailcaptura API chamadas e eventos relacionados feitos por ou em nome de sua AWS conta. Desse modo, ele fornece os arquivos de log para um bucket do Amazon S3 especificado por você. Você pode identificar quais usuários e contas ligaram AWS, o endereço IP de origem dessas chamadas e quando elas ocorreram. Para obter mais informações, consulte o Guia do usuário do AWS CloudTrail.

Tópicos

- Monitoramento HealthLake com a Amazon CloudWatch

Monitoramento HealthLake com a Amazon CloudWatch

Você pode monitorar HealthLake o uso CloudWatch, que coleta dados brutos e os processa em métricas legíveis e quase em tempo real. Essas estatísticas são mantidas por 15 meses, de maneira que você possa usar informações históricas e ter uma perspectiva melhor de como sua aplicação web ou o serviço está se saindo. Você também pode definir alarmes que observam determinados limites e enviam notificações ou realizam ações quando esses limites são atingidos. Para obter mais informações, consulte o Guia CloudWatch do usuário da Amazon.

As métricas são relatadas para todos HealthLake APIs, incluindo as seguintes.

- Gerenciamento do armazenamento de dados APIs — CreateFHIRDatastore, DeleteFHIRDatastore, DescribeFHIRDatastore, L istFHIRDatastores

- Importar e exportar APIs — S tartFHIRImport Job, L istFHIRImport Jobs, D escribeFHIRImport Job, S tartFHIRExport Job, L istFHIRExport Jobs, D escribeFHIRExport Job
- HTTPRESTGerenciamento de clientes e recursos APIs — CreateResource, DeleteResource, GetCapabilities, ReadResource, SearchAll, SearchWithGet, SearchWithPost, UpdateResource.
- Marcação APIs — ListTagsForResource,, TagResource UntagResource

A tabela a seguir lista as métricas e as dimensões do HealthLake.

As métricas a seguir são relatadas. Cada um é apresentado como uma contagem de frequência para um intervalo de dados especificado pelo usuário.

Metrics

Metrics	Descrição
Contagem de chamadas	O número de chamadas paraAPIs. Isso pode ser relatado para a conta ou para um datastore especificado. Unidades: contagem Estatísticas válidas: Sum, Count Dimensões: operação, ID do datastore, tipo de datastore
Solicitações bem-sucedidas	O número de API solicitações bem-sucedidas. Unidades: contagem Estatísticas válidas: soma, média Dimensões: operação, armazenamento de dados, tipo de armazenamento de dados
Erros do usuário	O número de solicitações que falharam devido a um erro do usuário. Unidades: contagem Estatísticas válidas: soma, média

Metrics	Descrição
	Dimensões: operação, ID do datastore, tipo de datastore
Erros de servidor	O número de solicitações que falharam devido a um erro do servidor. Unidades: contagem Estatísticas válidas: soma, média Dimensões: operação, ID do datastore, tipo de datastore
Solicitações limitadas	O número de solicitações que foram limitadas. Essa métrica não está incluída nas contagens de erros do usuário ou do servidor. Unidades: contagem Estatísticas válidas: soma, média Dimensões: operação, ID do datastore, tipo de datastore
Latência	O tempo gasto em milissegundos para processar a solicitação do usuário. Unidade: milissegundos Estatísticas válidas: mínimo, máximo, média Dimensões: operação, ID do datastore, tipo de datastore

As seguintes dimensões são relatadas.

Dimensões

Dimensões	Descrição
Operação	Qual API operação foi usada
DataStoreID	O armazenamento de dados incluído na API solicitação
DataStoreType	O tipo de armazenamento de dados (atualmente, somente o FHIR R4 é suportado)

Você pode obter métricas HealthLake com o AWS Management Console AWS CLI, o ou CloudWatch API o. Você pode usá-lo CloudWatch API por meio de um dos kits de desenvolvimento de AWS software da Amazon (SDKs) ou das CloudWatch API ferramentas. O HealthLake console exibe gráficos com base nos dados brutos do CloudWatch API.

Você deve ter as CloudWatch permissões apropriadas para monitorar HealthLake CloudWatch. Para obter mais informações, consulte Autenticação e controle de acesso para a Amazon CloudWatch no Guia CloudWatch do usuário da Amazon.

Visualizando HealthLake métricas

Para visualizar métricas (CloudWatch console)

1. Faça login no Console de gerenciamento da AWS e abra o console do CloudWatch.
2. Escolha Métricas, escolha Todas as métricas e, em seguida, escolha AWS/HealthLake.
3. Escolha a dimensão, informe um nome de métrica e selecione Adicionar ao gráfico.
4. Escolha um valor para o intervalo de datas. A contagem da métrica para o intervalo de datas selecionado é exibida no gráfico.

Criando um alarme usando CloudWatch

Um CloudWatch alarme monitora uma única métrica durante um período de tempo especificado e executa uma ou mais ações: enviar uma notificação para um tópico do Amazon Simple Notification Service (AmazonSNS) ou política de Auto Scaling. A ação ou ações são baseadas no valor da métrica em relação a um determinado limite em vários períodos que você especifica. CloudWatch também pode enviar uma SNS mensagem da Amazon quando o alarme muda de estado.

CloudWatch os alarmes invocam ações somente quando o estado muda e persiste durante o período especificado.

Para visualizar métricas (CloudWatch console)

1. Faça login no Console de gerenciamento da AWS e abra o console do CloudWatch.
2. Escolha Alarmes e, em seguida, Criar alarme.
3. Escolha AWS/eHealthLake, em seguida, escolha uma métrica.
4. Para Intervalo de tempo, escolha um intervalo de tempo para monitorar e, em seguida, selecione Avançar.
5. Preencha os campos Nome e Descrição.
6. Em Whenever, escolha >= e digite um valor máximo.
7. Se você quiser CloudWatch enviar um e-mail quando o estado do alarme for atingido, na seção Ações, para Sempre que este alarme for atingido, escolha Estado é ALARM. Em Enviar notificação para, escolha uma lista de endereçamento ou escolha Nova lista e crie uma nova lista de endereçamento.
8. Visualize o alarme na seção Prévia do alarme. Se você estiver satisfeito com o alarme, selecione Criar alarme.

Integrando SMART com FHIR AWS HealthLake

Aplicativos médicos substituíveis e tecnologias reutilizáveis (SMART) em um armazenamento de HealthLake dados FHIR habilitado permitem que aplicativos FHIR compatíveis SMART acessem dados armazenados em um armazenamento de dados. HealthLake HealthLake os dados são acessados autenticando e autorizando solicitações usando um servidor de autorização de terceiros e configurando recursos adicionais no. AWS

Para usar o SMART on FHIR com seu armazenamento de HealthLake dados, você deve fornecer o seguinte em sua reateFHIRDatastore API solicitação C.

- Defina o AuthorizationStrategyigual `SMART_ON_FHIR_V1` a.
- Defina o IdpLambdaArn igual ao ARN que AWS Lambda você criou para gerenciar a decodificação do token com seu servidor de autorização.
- Defina os elementos de metadados especificados em seu servidor de autorização. Esses elementos de metadados são retornados no Documento de descoberta. Para saber mais, consulte Buscando um documento SMART de descoberta do armazenamento de HealthLake dados FHIR ativado.
- Opcional: ative FineGrainedAuthorizationEnabledse você tiver configurado uma autorização refinada no seu servidor de autorização.

Você pode criar um armazenamento de dados SMART FHIR ativado usando o AWS Command Line Interface (AWS CLI) ou por meio de um dos AWS suportadosSDKs. A criação SMART de um armazenamento HealthLake de dados FHIR ativado não é suportada usando o HealthLake console. Para saber mais, consulte Crie um armazenamento SMART de dados não FHIR habilitado.

Para prescrever esses parâmetros na solicitação, você precisa configurar recursos em outros AWS serviços (AWS Secrets Manager e AWS Lambda), criar novas funções de IAM serviço e configurar um SMART servidor de autorização FHIR compatível. Use a seção Configurando os recursos necessários para implementar um armazenamento de dados FHIR compatível com SMART on para saber mais sobre como configurar os recursos necessários e ter uma visão geral de alto nível de como um FHIR aplicativo SMART on interage com. HealthLake

Isso significa que, em vez de gerenciar as credenciais do usuário por meio de AWS Identity and Access Management você, você está usando um SMART servidor de autorização FHIR compatível.

HealthLake suporta SMART em FHIR 1.0. Para saber mais sobre essa estrutura, consulte o Guia de implementação do SMART Application Launch Framework, versão 1.0.

Para autorizar e autenticar solicitações de armazenamento de dados usando SMART onFHIR, HealthLake suporta o uso de:

- Integração com OpenID (AuthN): usada para autenticar que a pessoa ou o aplicativo do cliente é quem (ou o que) ela afirma ser.
- OAuthIntegração 2.0 (AuthZ): usada para autorizar quais FHIR recursos em seu armazenamento de HealthLake dados uma solicitação autenticada também pode ler ou gravar dados. Isso é definido pelos escopos configurados em seu servidor de autorização.

Sumário

Requisitos de autenticação para SMART um FHIR

Para acessar FHIR recursos SMART em um armazenamento de FHIR HealthLake dados interno, um aplicativo cliente deve ser autorizado por um servidor de autorização OAuth compatível com 2.0 e apresentar um token de OAuth portador como parte de uma solicitação. FHIR REST API Para encontrar o endpoint do servidor de autorização, use o documento HealthLake SMART on FHIR Discovery por meio de um identificador uniforme de recursos conhecido. Para saber mais sobre esse processo, consulte Buscando um documento SMART de descoberta do armazenamento de HealthLake dados FHIR ativado.

Ao criar um armazenamento de FHIR HealthLake dados SMART no, você deve definir o ponto final do servidor de autorização e o ponto final do token no `metadata` elemento da reateFHIRDatastore solicitação C. Para saber mais sobre como definir o `metadata` elemento, consulteCriação de um SMART armazenamento HealthLake de dados FHIR ativado.

Usando os endpoints do servidor de autorização, o aplicativo cliente autenticará um usuário com o serviço de autorização. Uma vez autorizado e autenticado, um JSON Web Token (JWT) é gerado pelo serviço de autorização e passado para o aplicativo cliente. Esse token contém escopos de FHIR recursos que o aplicativo cliente pode usar, o que, por sua vez, restringe quais dados o usuário pode acessar. Opcionalmente, se o escopo de lançamento foi fornecido, a resposta conterá esses detalhes. Para saber mais SMART sobre os FHIR escopos compatíveis com HealthLake, consulteApoiado SMART em FHIR OAuth escopos por HealthLake.

Usando o JWT concedido pelo servidor de autorização, um aplicativo cliente faz FHIR REST API chamadas para um armazenamento de HealthLake dados SMART FHIR ativado. Para validar

e decodificar oJWT, você precisa criar uma função Lambda. HealthLake invoca essa função do Lambda em seu nome quando uma FHIR REST API solicitação é recebida. Para ver um exemplo de função Lambda inicial, consulte. Usando AWS Lambda para validação de token com um SMART armazenamento HealthLake de dados FHIR ativado

Elementos do servidor de autorização necessários para criar um SMART armazenamento HealthLake de dados FHIR ativado

Na reateFHIRDatastore solicitação C, você precisa fornecer o endpoint de autorização e o endpoint do token como parte do `metadata` elemento no `IdentityProviderConfiguration` objeto. Tanto o endpoint de autorização quanto o endpoint do token são obrigatórios. Para ver um exemplo de como isso é especificado na reateFHIRDatastore solicitação C, consulteCriação de um SMART armazenamento HealthLake de dados FHIR ativado.

Declarações obrigatórias para concluir uma FHIR REST API solicitação SMART em um armazenamento de HealthLake dados FHIR ativado

Sua AWS Lambda função deve conter as seguintes declarações para que seja uma FHIR REST API solicitação válida SMART em um armazenamento de HealthLake dados FHIR ativado.

- `nbf`: Reivindicação (não anterior) — A reclamação “nbf” (não anterior) identifica o momento antes do qual ela JWT MUST NOT será aceita para processamento. O processamento da reclamação “nbf” exige que a atual esteja date/time MUST be after or equal to the not-before date/time listada na reivindicação “nbf”. O exemplo da função Lambda que fornecemos é convertido `iat` da resposta do servidor em. `nbf`
- `exp`: (Tempo de expiração) Reivindicação — A reivindicação “exp” (prazo de expiração) identifica o prazo de validade a partir do qual ela não JWT deve ser aceita para processamento.
- `isAuthorized`: Um booleano definido como. `True` Indica que a solicitação foi autorizada no servidor de autorização.
- `aud`: Reivindicação (Audiência) — A afirmação “aud” (audiência) identifica os destinatários aos quais se destina. JWT Esse deve ser um endpoint SMART de armazenamento HealthLake de dados FHIR ativado.
- `scope`: deve ser pelo menos um escopo relacionado a um FHIR recurso. Esse escopo é definido em seu servidor de autorização. Para saber mais sobre os escopos relacionados a FHIR recursos aceitos por HealthLake, consulteHealthLake escopos específicos de FHIR recursos de armazenamento de dados.

Apoiado SMART em FHIR OAuth escopos por HealthLake

HealthLake usa OAuth 2.0 como protocolo de autorização. O uso desse protocolo em seu servidor de autorização permite que você defina quais FHIR recursos em seu armazenamento de HealthLake dados um aplicativo cliente também pode ter acesso de leitura e/ou gravação.

A FHIR estrutura SMART on define um conjunto de escopos que podem ser solicitados do servidor de autorização. Para ver as definições de escopo na FHIR estrutura SMART on, consulte SMARTFHIREscopos no Guia de HL7 FHIR Recursos.

Por exemplo, um aplicativo cliente projetado apenas para permitir que os pacientes visualizem seus resultados laboratoriais ou visualizem seus detalhes de contato só deve ser autorizado a solicitar (por meio de FHIR REST solicitação) `read` escopos. Para defini-los como escopo, você forneceria uma string como a seguinte`patient/Observation.read`. Isso permitiria que o aplicativo cliente solicitasse acesso ao tipo de `Observation` recurso de maneira somente para leitura no tipo de `Patient` recurso.

Escopo de lançamento independente

HealthLake suporta o escopo `launch/patient` do modo de inicialização autônomo.

No modo de inicialização autônomo, um aplicativo cliente solicita acesso aos dados clínicos do paciente porque o usuário e o paciente não são conhecidos pelo aplicativo cliente. Assim, a solicitação de autorização do aplicativo do cliente solicita explicitamente que o escopo do paciente seja devolvido. Após a autenticação bem-sucedida, o servidor de autorização emite um token de acesso contendo o escopo do paciente de lançamento solicitado. O contexto necessário do paciente é fornecido junto com o token de acesso na resposta do servidor de autorização.

Escopos do modo de lançamento suportados

Escopo	Descrição
`launch/patient`	Um parâmetro em uma solicitação de autorização OAuth 2.0 solicitando que os dados do paciente sejam retornados na resposta de autorização.

HealthLake escopos específicos de FHIR recursos de armazenamento de dados

HealthLake define três níveis de escopos.

- Os escopos específicos do paciente concedem acesso a dados específicos sobre um único paciente. Qual paciente é especificado no contexto de lançamento.
- Os escopos em nível de usuário concedem acesso a dados específicos que um usuário pode acessar.
- Os escopos no nível do sistema concedem acesso de leitura/gravação a todos os FHIR recursos encontrados no armazenamento de dados. HealthLake

A tabela a seguir mostra a sintaxe para criar escopos relacionados a FHIR recursos que são suportados pelo. HealthLake O formato geral é o seguinte:

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

Escopos de autorização compatíveis em armazenamentos de HealthLake dados

Sintaxe do escopo	Exemplo de escopo	Resultado
`patient/(fhir-resource | '*').('read' | 'write' | '*')`	`patient/AllergyIntolerance.*`	Um aplicativo cliente teria acesso de leitura/gravação a alergias.
`user/(fhir-resource | '*').('read' | 'write' | '*')`	`user/Observation.read`	Um aplicativo cliente teria acesso de leitura a todas as observações registradas.
`system/('read' | 'write' | *)`	`system/*.*`	Um aplicativo cliente teria acesso de leitura/gravação a todos os dados.

Usando AWS Lambda para validação de token com um SMART armazenamento HealthLake de dados FHIR ativado

Ao criar um armazenamento SMART de HealthLake dados FHIR ativado, você precisa fornecer a ARN AWS Lambda função da `CreateFHIRDatastore` solicitação. A função Lambda ARN é especificada no `IdentityProviderConfiguration` objeto usando o `IdpLambdaArn` parâmetro.

Você deve criar a função Lambda antes de criar seu próprio armazenamento de HealthLake dados SMART FHIR habilitado. Depois de criar o armazenamento de dados, o Lambda ARN não pode ser alterado. Para ver o Lambda ARN que você especificou quando o armazenamento de dados foi criado, use a `DescribeFHIRDatastore` API operação.

Para que uma FHIR REST solicitação seja bem-sucedida SMART em um armazenamento de HealthLake dados FHIR ativado, sua função Lambda precisa fazer o seguinte:

- A função Lambda deve retornar uma resposta em menos de 1 segundo para o endpoint do armazenamento HealthLake de dados.
- Decodifique o token de acesso fornecido no cabeçalho de autorização da REST API solicitação enviada pelo aplicativo cliente.
- Atribua uma função de IAM serviço que tenha permissões suficientes para realizar a FHIR REST API solicitação.
- As seguintes reivindicações são necessárias para concluir uma FHIR REST API solicitação. Para saber mais, consulte Reivindicações obrigatórias.
  - `nbf`
  - `exp`
  - `isAuthorized`
  - `aud`
  - `scope`

Ao trabalhar com o Lambda, você precisa criar uma função de execução e uma política baseada em recursos, além da sua função Lambda. A função de execução de uma função do Lambda é uma IAM função que concede à função permissão para acessar AWS serviços e recursos necessários em tempo de execução. A política baseada em recursos que você fornece deve permitir HealthLake invocar sua função em seu nome.

As seções deste tópico descrevem um exemplo de solicitação de um aplicativo cliente e uma resposta decodificada, as etapas necessárias para criar uma função AWS Lambda e como criar uma política baseada em recursos que possa assumir. HealthLake

- Parte 1: Criando uma função Lambda
- Parte 2: Criando uma função HealthLake de serviço usada pela função AWS Lambda
- Parte 3: Atualizando a função de execução da função Lambda
- Parte 4: Adicionando uma política de recursos à sua função Lambda
- Parte 5: Provisionando a simultaneidade para sua função Lambda

Criação de uma função AWS Lambda

A função Lambda criada neste tópico é acionada quando HealthLake recebe uma solicitação para um armazenamento de HealthLake dados não SMART FHIR habilitado. A solicitação do aplicativo cliente contém uma REST API chamada e um cabeçalho de autorização contendo um token de acesso.

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

O exemplo da função Lambda neste tópico é usado AWS Secrets Manager para obscurecer as credenciais relacionadas ao servidor de autorização. É altamente recomendável não fornecer detalhes de login do servidor de autorização diretamente em uma função Lambda.

Example validando uma FHIR REST solicitação contendo um token do portador da autorização

O exemplo da função Lambda mostra como validar uma FHIR REST solicitação enviada para um armazenamento de dados SMART FHIR ativado HealthLake . Para ver step-by-steps instruções sobre como implementar essa função Lambda, consulte. Criando uma função Lambda usando o AWS Management Console

Se a FHIR REST API solicitação não contiver um endpoint de armazenamento de dados, token de acesso e REST operação válidos, a função Lambda falhará. Para saber mais sobre os elementos necessários do servidor de autorização, consulteReivindicações obrigatórias.

```
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()
```

Criando uma função Lambda usando o AWS Management Console

Esse procedimento pressupõe que você já criou a função de serviço que deseja assumir HealthLake ao lidar com uma FHIR REST API solicitação SMART em um armazenamento de HealthLake dados FHIR ativado. Se você não criou a função de serviço, ainda pode criar a função Lambda. Você precisará adicionar a função ARN of service antes que a função Lambda funcione. Para saber mais sobre como criar uma função de serviço e especificá-la na função Lambda, consulte Criação de uma função HealthLake de serviço para uso na função AWS Lambda usada para decodificar um JWT

Para criar uma função Lambda ()AWS Management Console

1. Abra a página Funções do console do Lambda.
2. Escolha a opção Criar função.
3. Selecione Criar do zero.
4. Em Informações básicas, insira um nome de função. Em Tempo de execução, escolha um tempo de execução baseado em python.
5. Em Execution role (Perfil de execução), escolha Create a new role with basic Lambda permissions (Criar um novo perfil com as permissões básicas do Lambda).

   O Lambda cria uma função de execução que concede à função permissão para fazer upload de registros para a Amazon. CloudWatch A função Lambda assume a função de execução quando você invoca sua função e usa a função de execução para criar credenciais para o. AWS SDK
6. Escolha a guia Código e adicione o exemplo da função Lambda.

Se você ainda não criou a função de serviço para usar a função Lambda, precisará criá-la antes que a função de amostra do Lambda funcione. Para saber mais sobre a criação de uma função de serviço para a função Lambda, consulte. Criação de uma função HealthLake de serviço para uso na função AWS Lambda usada para decodificar um 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()
```

Modificando a função de execução de uma função Lambda

Depois de criar a função Lambda, você precisa atualizar a função de execução para incluir as permissões necessárias para chamar o Secrets Manager. No Secrets Manager, cada segredo que você cria tem umARN. Para aplicar o menor privilégio, a função de execução só deve ter acesso aos recursos necessários para a execução da função Lambda.

Você pode modificar a função de execução de uma função Lambda pesquisando-a no IAM console ou escolhendo Configuração no console Lambda. Para saber mais sobre como gerenciar sua função de execução de funções Lambda, consulte. Função de execução do Lambda

Example Função de execução da função Lambda que concede acesso a **`GetSecretValue`**

Adicionar a IAM ação `GetSecretValue` à função de execução concede a permissão necessária para que a função Lambda de amostra funcione.

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

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

Neste ponto, você criou uma função Lambda que pode ser usada para validar o token de acesso fornecido como parte da FHIR REST solicitação enviada ao seu próprio armazenamento de dados SMART FHIR HealthLake habilitado.

Criação de uma função HealthLake de serviço para uso na função AWS Lambda usada para decodificar um JWT

Pessoa: Administrador IAM

Um usuário que pode adicionar ou remover IAM políticas e criar novas IAM identidades.

Perfil de serviço

Um perfil de serviço é um perfil do IAM que um serviço assume para realizar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para obter mais informações, consulte Criar uma função para delegar permissões a um AWS service (Serviço da AWS) no Guia do IAM usuário.

Depois que o JSON Web Token (JWT) é decodificado, a autorização Lambda também precisa retornar IAM uma função. ARN Essa função deve ter as permissões necessárias para realizar a REST API solicitação ou falhará devido a permissões insuficientes.

Ao configurar uma política personalizadaIAM, é melhor conceder as permissões mínimas necessárias. Para saber mais, consulte Aplicar permissões de privilégios mínimos no Guia do usuário. IAM

A criação HealthLake de uma função de serviço para designar na função Lambda de autorização requer duas etapas.

- Primeiro, você precisa criar uma IAM política. A política deve especificar o acesso aos FHIR recursos para os quais você forneceu escopos no servidor de autorização.
- Em segundo lugar, você precisa criar a função de serviço. Ao criar a função, você designa uma relação de confiança e anexa a política criada na etapa um. A relação de confiança é designada HealthLake como principal do serviço. Você precisa especificar um armazenamento HealthLake de dados ARN e uma ID de AWS conta nesta etapa.

Criação de uma nova IAM política

Os escopos que você define no seu servidor de autorização determinam a quais FHIR recursos um usuário autenticado tem acesso em um armazenamento de HealthLake dados.

A IAM política que você cria pode ser personalizada para corresponder aos escopos que você definiu.

As seguintes ações no `Action` elemento de uma declaração de IAM política podem ser definidas. Para cada um `Action` na tabela, você pode definir um`Resource types`. Em HealthLake um armazenamento de dados, é o único tipo de recurso compatível que pode ser definido no `Resource` elemento de uma declaração de política de IAM permissão.

FHIRRecursos individuais não são um recurso que você possa definir como um elemento em uma política de IAM permissão.

Ações definidas por HealthLake

Ações	Descrição	Nível de acess	Tipo de recurso (Obrigatório)
CreateResource	Concede permissão a um recurso de criação	Escre	Armazenamento de dados: arn:aws:healthlake ARN: ::datastore/fhir/ **your-region 111122223333 your-datastore-id**
DeleteResource	Concede permissão para excluir um recurso	Escre	Armazenamento de dados: arn:aws:healthlake ARN: ::datastore/fhir/ **your-region 111122223333 your-datastore-id**

Ações	Descrição	Nível de acess	Tipo de recurso (Obrigatório)
ReadResource	Concede permissão para ler um recurso	Leitur	Armazenamento de dados: arn:aws:healthlake ARN: ::datastore/fhir/ **your-region 111122223333 your-datastore-id**
SearchWithGet	Concede permissão para pesquisar recursos com o GET método	Leitur	Armazenamento de dados: arn:aws:healthlake ARN: ::datastore/fhir/ **your-region 111122223333 your-datastore-id**
SearchWithPost	Concede permissão para pesquisar recursos com o POST método	Leitur	Armazenamento de dados: arn:aws:healthlake ARN: ::datastore/fhir/ **your-region 111122223333 your-datastore-id**
StartFHIRExportJobWithPost	Concede permissão para iniciar um trabalho de FHIR exportação com GET	Escre	Armazenamento de dados: arn:aws:healthlake ARN: ::datastore/fhir/ **your-region 111122223333 your-datastore-id**
UpdateResource	Concede permissão para atualizar um recurso	Escre	Armazenamento de dados: arn:aws:healthlake ARN: ::datastore/fhir/ **your-region 111122223333 your-datastore-id**

Para começar, você pode usar`AmazonHealthLakeFullAccess`. Essa política permitiria leitura, gravação, pesquisa e exportação em todos os FHIR recursos encontrados em um armazenamento de dados. Para conceder permissões somente de leitura em um armazenamento de dados, use. `AmazonHealthLakeReadOnlyAccess`

Para saber mais sobre como criar uma política personalizada usando o AWS Management Console, AWS CLI, ou IAMSDKs, consulte Criação de IAM políticas no Guia IAM do usuário.

Criação de uma função de serviço para HealthLake (IAMconsole)

Use esse procedimento para criar uma função de serviço. Ao criar um serviço, você também precisará designar uma IAM política.

Para criar a função de serviço para HealthLake (IAMconsole)

1. Faça login no AWS Management Console e abra o IAM console em https://console.aws.amazon.com/iam/.
2. No painel de navegação do console do IAM, selecione Roles (Funções).
3. Depois, escolha Create role (Criar função).
4. Na página Selecionar entidade confiável, escolha Política de confiança personalizada.
5. Em seguida, em Política de confiança personalizada, atualize a política de amostra da seguinte forma. **`your-account-id`**Substitua pelo número da sua conta e adicione o armazenamento ARN de dados que você deseja usar em seus trabalhos de importação ou exportação.

```
{
  "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. Em seguida, escolha Próximo.
7. Na página Adicionar permissões, escolha a política que você deseja que o HealthLake serviço assuma. Para encontrar sua política, procure-a em Políticas de permissões.
8. Em seguida, escolha Anexar política.
9. Em seguida, na página Nome, revise e crie, em Nome da função, insira um nome.
10. (Opcional) Em seguida, em Descrição, adicione uma breve descrição para sua função.

11. Se possível, insira um nome de função ou sufixo de nome de função para ajudar a identificar o propósito desta função. Os nomes das funções devem ser exclusivos em seu Conta da AWS. Eles não são diferenciados por letras maiúsculas e minúsculas. Por exemplo, não é possível criar perfis denominados **`PRODROLE`** e **`prodrole`**. Como várias entidades podem fazer referência à função, não é possível editar o nome da função depois que ela é criada.
12. Revise os detalhes da função e escolha Criar função.

Para saber como especificar a função ARN no exemplo da função Lambda, consulte. Criação de uma função AWS Lambda

Função de execução do Lambda

A função de execução de uma função Lambda é uma IAM função que concede à função permissão para acessar AWS serviços e recursos. Esta página fornece informações sobre como criar, visualizar e gerenciar o perfil de execução de uma função do Lambda.

Por padrão, o Lambda cria uma função de execução com permissões mínimas quando você cria uma nova função do Lambda usando o. AWS Management Console Para gerenciar as permissões concedidas na função de execução, consulte Criação de uma função de execução no IAM console no Lambda Developer Guide.

O exemplo da função Lambda fornecido neste tópico usa o Secrets Manager para ocultar as credenciais do servidor de autorização.

Como acontece com qualquer IAM função que você criar, é importante seguir as melhores práticas de privilégios mínimos. Durante a frase de desenvolvimento, às vezes você pode conceder permissões além do necessário. Antes de publicar sua função no ambiente de produção, como prática recomendada, ajuste a política para incluir somente as permissões necessárias. Para obter mais informações, consulte Aplicar o least-privelege no Guia do usuário. IAM

Permitir HealthLake acionar sua função Lambda

Para que HealthLake possa invocar a função Lambda em seu nome, você deve fazer o seguinte:

- Você precisa definir `IdpLambdaArn` igual à função ARN Lambda que você deseja HealthLake invocar na solicitação. `CreateFHIRDatastore`
- Você precisa de uma política baseada em recursos que permita HealthLake invocar a função Lambda em seu nome.

Quando HealthLake recebe uma FHIR REST API solicitação SMART em um armazenamento de HealthLake dados FHIR ativado, ele precisa de permissões para invocar a função Lambda especificada na criação do armazenamento de dados em seu nome. Para conceder HealthLake acesso, você usará uma política baseada em recursos. Para saber mais sobre como criar uma política baseada em recursos para uma função do Lambda, consulte Permitir que um AWS serviço chame uma função do Lambda no Guia do desenvolvedor.AWS Lambda

Provisionando a simultaneidade para sua função Lambda

> **Important**
>
> HealthLake exige que o tempo máximo de execução da função Lambda seja inferior a um segundo (1000 milissegundos).
> Se sua função Lambda exceder o limite de tempo de execução, você receberá uma TimeOutexceção.

Para evitar essa exceção, recomendamos configurar a simultaneidade provisionada. Ao alocar simultaneidade provisionada antes de um aumento nas invocações, é possível garantir que todas as solicitações sejam atendidas por instâncias inicializadas com latência baixa. Para saber mais sobre como configurar a simultaneidade provisionada, consulte Como configurar a simultaneidade provisionada no Lambda Developer Guide

Para ver o tempo médio de execução de sua função Lambda atualmente, use a página de monitoramento de sua função Lambda no console Lambda. Por padrão, o console Lambda fornece um gráfico de duração que mostra a quantidade média, mínima e máxima de tempo que seu código de função gasta processando um evento. Para saber mais sobre o monitoramento das funções do Lambda, consulte Funções de monitoramento no console do Lambda no Guia do desenvolvedor do Lambda.

Se você já provisionou a simultaneidade para sua função do Lambda e deseja monitorá-la, consulte Monitoramento da simultaneidade no Guia do desenvolvedor do Lambda.

Criação de um SMART armazenamento HealthLake de dados FHIR ativado

Para usar a FHIR estrutura SMART on com HealthLake, crie um armazenamento de HealthLake dados com o `IdentityProviderConfiguration` parâmetro especificado em sua

reateFHIRDatastore solicitação C. No `IdentityProviderConfiguration` parâmetro, você especifica as seguintes informações:

- Defina o AuthorizationStrategyigual `SMART_ON_FHIR_V1` a.
- Defina o IdpLambdaArnigual ao ARN que AWS Lambda você criou para gerenciar a decodificação do token com seu servidor de autorização.
- Defina os elementos de metadados especificados no servidor de autorização como um JSON bloco. Esses elementos de metadados são retornados no Documento de descoberta.
- Opcional: Ativar FineGrainedAuthorizationEnabled. Especifique `True` para usar a autorização refinada fornecida por HealthLake

Você pode criar um armazenamento de dados SMART FHIR ativado usando o AWS Command Line Interface (AWS CLI) ou por meio de um dos AWS suportadosSDKs. A criação SMART de um armazenamento HealthLake de dados FHIR ativado não é suportada usando o HealthLake console.

Usando o AWS CLI para criar um armazenamento SMART de HealthLake dados FHIR ativado

Você pode usar o exemplo de código a seguir para criar um SMART armazenamento HealthLake de dados FHIR habilitado usando AWS CLI o. Ao criar um armazenamento SMART de HealthLake dados FHIR ativado, você deve especificar o `identity-provider-configuration`parâmetro.

No `identity-provider-configuration` parâmetro, você pode habilitar opcionalmente uma autorização refinada definindo igual a. `FineGrainedAuthorizationEnabled` `True` Para saber mais sobre a autorização refinada, consulteUsando autorização refinada com um armazenamento de dados SMART ativado FHIR HealthLake . O exemplo abaixo contém um caractere especial \ para indicar quebras de linha ou como um caractere de escape. Isso é para maior clareza.

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

Quando bem-sucedido, você recebe a seguinte JSON resposta:

```
{
  "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"
}
```

Usando autorização refinada com um armazenamento de dados SMART ativado FHIR HealthLake

Os escopos por si só não fornecem a especificidade necessária sobre quais dados um solicitante está autorizado a acessar em um armazenamento de dados. O uso de autorização refinada permite um nível mais alto de especificidade ao conceder acesso a um SMART armazenamento de dados ativado. FHIR HealthLake Para usar uma autorização refinada, defina `FineGrainedAuthorizationEnabled` igual a `True` no `IdentityProviderConfiguration` parâmetro da sua solicitação C. reateFHIRDatastore

Se você habilitou a autorização refinada, seu servidor de autorização retornará um `fhirUser` escopo `id_token` junto com o token de acesso. Isso permite que as informações sobre o usuário sejam recuperadas pelo aplicativo do cliente. O aplicativo cliente deve tratar a `fhirUser` declaração

como a URI de um FHIR recurso representando o usuário atual. Pode ser `Patient`, `Practitioner` ou `RelatedPerson`. A resposta do servidor de autorização também inclui um `user/` escopo que define quais dados o usuário pode acessar. Isso usa a sintaxe definida para escopos relacionados a escopos específicos FHIR de recursos:

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

Veja a seguir exemplos de como a autorização refinada pode ser usada para especificar ainda mais os tipos de recursos relacionados ao FHIR acesso a dados.

- Quando `fhirUser` é uma `Practitioner` autorização refinada determina a coleção de pacientes que o usuário pode acessar. `fhirUser`O acesso é permitido apenas para os pacientes em que o paciente se refere ao `fhirUser` como clínico geral.

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

- Quando `fhirUser` é um `Patient` ou `RelatedPerson` e o paciente referenciado na solicitação é diferente do`fhirUser`, uma autorização refinada determina o acesso `fhirUser` do paciente solicitado. O acesso é permitido quando há um relacionamento especificado no `Patient` recurso solicitado.

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

Buscando um documento SMART de descoberta do armazenamento de HealthLake dados FHIR ativado

Para que um aplicativo cliente faça uma FHIR REST solicitação bem-sucedida, ele precisa reunir os requisitos de autorização definidos no armazenamento de HealthLake dados. Nenhuma autorização (token do portador) é necessária para que essa solicitação seja bem-sucedida.

Para fazer isso, faça uma `GET` solicitação e anexe `/.well-known/smart-configuration` ao endpoint do armazenamento de dados

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

Isso retorna o documento de descoberta do armazenamento de HealthLake dados como um JSON blob. Nele, você encontrará o `authorization_endpoint` e o `token_endpoint` junto com as especificações e recursos definidos no armazenamento HealthLake de dados.

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

URLsnecessário para iniciar um aplicativo cliente com sucesso

- Ponto final de autorização: o URL necessário para autorizar um aplicativo ou usuário cliente.
- Ponto final do token: o ponto final do servidor de autorização que o aplicativo cliente usa para se comunicar com ele.

Fazendo uma FHIR REST API solicitação em um armazenamento de HealthLake dados SMART habilitado

Você pode fazer FHIR REST API solicitações SMART em um armazenamento FHIR de HealthLake dados ativado. O exemplo a seguir mostra uma solicitação do aplicativo cliente contendo um JWT no cabeçalho de autorização e como o Lambda deve decodificar a resposta. Depois que a solicitação do aplicativo cliente for autorizada e autenticada, ela deverá receber um token portador do servidor de autorização. Use o token do portador no cabeçalho de autorização ao enviar uma FHIR REST API solicitação SMART em um armazenamento FHIR de HealthLake dados ativado.

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

Como um token portador foi encontrado no cabeçalho de autorização e nenhuma AWS IAM identidade foi detectada, HealthLake invoca a função Lambda especificada quando o armazenamento de dados SMART FHIR ativado HealthLake foi criado. Quando o token é decodificado com sucesso pela sua função Lambda, aqui está um exemplo de resposta enviada para. 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
}
```

Configurando os recursos necessários para implementar um SMART armazenamento de dados FHIR compatível

Este tópico descreve os recursos que você precisa provisionar em sua AWS conta externa HealthLake, criando um SMART armazenamento de HealthLake dados FHIR ativado e como um aplicativo SMART no FHIR cliente interagiria com um servidor de autorização e um armazenamento de HealthLake dados.

As etapas desse fluxo de trabalho definem as etapas básicas de como SMART as FHIR solicitações são tratadas e quais recursos são necessários para que elas sejam bem-sucedidas.

SMARTEm um processo FHIR sob solicitação, três aplicativos funcionam juntos:

- O usuário final: geralmente, um paciente ou médico que usa um FHIR aplicativo de terceiros para acessar dados SMART em um HealthLake armazenamento de dados.
- O FHIR aplicativo SMART on (chamado de aplicativo cliente): um aplicativo que deseja acessar dados encontrados no armazenamento de HealthLake dados.
- O servidor de autorização: um servidor compatível com o OpenID Connect que é capaz de autenticar usuários e emitir tokens de acesso.
- O armazenamento de HealthLake dados: um armazenamento de HealthLake dados SMART FHIR ativado que usa uma função Lambda para responder às FHIR REST solicitações que fornecem um token portador.

Para que esses aplicativos funcionem juntos, você precisa criar os seguintes recursos.

Recomendamos criar o armazenamento de HealthLake dados SMART FHIR ativado depois de configurar o servidor de autorização, definir os escopos necessários nele e criar uma AWS Lambda função para lidar com a introspecção de tokens.

1. Configurando um endpoint do servidor de autorização — Servidor de autorização

Para usar a FHIR estrutura SMART on, você precisa configurar um servidor de autorização de terceiros que possa validar as FHIR REST solicitações feitas em um armazenamento de dados. Para saber mais sobre como configurar um endpoint do servidor de autorização com o qual funcionará HealthLake, consulteRequisitos de autenticação para SMART um FHIR.

2. Defina escopos para controlar quem pode acessar quais dados em seu armazenamento de HealthLake dados em seu servidor de autorização — Servidor de autorização

A FHIR estrutura SMART on usa OAuth escopos para determinar a quais FHIR recursos uma solicitação autenticada tem acesso e em que extensão. Definir escopos é uma forma de projetar com o mínimo de privilégios. Para saber mais sobre os escopos definidos pela FHIR estrutura SMART on e suportados pelo, HealthLake consulte,Apoiado SMART em FHIR OAuth escopos por HealthLake.

3. Configure uma AWS Lambda função capaz de realizar a introspecção de tokens — sua conta AWS

Uma FHIR REST solicitação enviada pelo aplicativo cliente SMART em um armazenamento de dados FHIR ativado conterá um JSON Web Token (JWT). Para saber mais sobre como configurar uma função Lambda capaz de decodificá-la e validá-la, consulte Decodificando a. JWT

4. Crie um armazenamento SMART de HealthLake dados FHIR ativado — sua AWS conta

Para criar um armazenamento SMART de FHIR HealthLake dados interno, você precisa fornecer um`IdentityProviderConfiguration`. Para saber mais sobre `IdentityProviderConfiguration` os parâmetros necessários em uma reateFHIRDatastore solicitação C, consulte Criação de um armazenamento de HealthLake dados SMART FHIR ativado.

Como um aplicativo cliente inicia e solicita dados de um SMART armazenamento de HealthLake dados FHIR ativado

Esta seção explica como um aplicativo cliente é iniciado no FHIR contexto SMART on e é capaz de fazer uma FHIR REST solicitação bem-sucedida em um armazenamento de HealthLake dados.

1. O aplicativo cliente faz uma **`GET`** solicitação ao Known Uniform Resource Identifier

Um aplicativo cliente SMART habilitado precisa fazer uma `GET` solicitação para encontrar os endpoints de autorização do seu armazenamento de HealthLake dados. Isso é feito por meio de uma solicitação conhecida do Uniform Resource Identifier (URI). Para saber mais sobre isso, consulte Buscando um documento SMART de descoberta do armazenamento de HealthLake dados FHIR ativado.

2. Solicitando acesso e escopos

O aplicativo cliente usa o ponto final de autorização do servidor de autorização para que o usuário possa fazer login. Esse processo autentica o usuário. Os escopos são usados para definir quais FHIR recursos em seu armazenamento HealthLake de dados um aplicativo cliente pode acessar. Para saber mais sobre a definição de escopos, consulteApoiado SMART em FHIR OAuth escopos por HealthLake.

3. Tokens de acesso

Agora que o usuário foi autenticado, um aplicativo cliente recebe um token de JWT acesso do servidor de autorização. Esse token é fornecido quando o aplicativo cliente envia uma FHIR REST solicitação para HealthLake o. Para saber mais sobre como o JWT é decodificado usando uma função Lambda, consulte. Realizando a validação do token

4. Fazendo uma FHIR REST solicitação SMART em um armazenamento de HealthLake dados FHIR habilitado

Agora, o aplicativo cliente pode enviar uma FHIR REST solicitação para um endpoint do armazenamento de HealthLake dados usando o token de acesso fornecido pelo servidor de autorização. Para ver um exemplo de FHIR REST solicitação, consulteFazendo uma FHIR REST API solicitação em um armazenamento de HealthLake dados SMART habilitado.

5. Validando o token de JWT acesso

Para validar o token de acesso enviado na FHIR REST solicitação, use uma função Lambda. Para ver como criar uma função Lambda que possa realizar a introspecção de tokens, consulte. Criação de uma função AWS Lambda

Usando a geração automatizada de recursos com base no processamento de linguagem natural (NLP) do tipo de FHIR DocumentReference recurso em AWS HealthLake

> **Note**
>
> Depois de 20 de fevereiro de 2023, HealthLake os armazenamentos de dados não usam processamento de linguagem natural integrado (NLP) por padrão. Se você estiver interessado em ativar esse recurso em seu armazenamento de dados, consulte o Como faço para ativar o recurso integrado HealthLake de processamento de linguagem natural do? capítulo Solução de problemas.
> Se você ativou o Amazon Comprehend Medical NLP integrado, ao criar ou `DocumentReference` atualizar recursos, você incorrerá em cobranças em sua conta. AWS Para obter mais detalhes, consulte AWS HealthLake os preços.
> O Amazon Comprehend Medical não está disponível na Ásia-Pacífico (Mumbai). HealthLake os armazenamentos de dados criados na região Ásia-Pacífico (Mumbai) não oferecem suporte ao processamento integrado de linguagem natural (NLP).

HealthLake fornece automaticamente o processamento integrado de linguagem natural (NLP) usando o Amazon Comprehend Medical para processamento de dados não estruturados para dados armazenados no tipo de recurso. `DocumentReference` Para fazer isso, HealthLake ligue para o Amazon Comprehend Medical `DetectEntities-V2` e as operações. `InferICD10-CM` `InferRxNorm` API Os resultados são automaticamente anexados ao `DocumentReference` recurso como uma extensão. Quando as API operações do Amazon Comprehend Medical detectam características que são`SIGN`,, e `SYMPTOM``DIAGNOSIS`, um tipo de `Linkage` recurso é gerado automaticamente. Novos recursos de condição e observação são criados a partir de entidades identificadas com as características de `SIGN``SYMPTOM`,`DIAGNOSIS`, ou e são vinculados ao documento de origem com esse recurso de vinculação.

Para recursos gerados pelo integrado`NLP`, você pode fazer `GET` solicitações, mas não há suporte para pesquisar esses novos recursos.

Para saber mais sobre como pesquisar essas extensões usando a integração com HealthLake o Athena, consulte. Consulte seu armazenamento HealthLake de dados usando SQL

Sumário

Como o Amazon Comprehend Medical está integrado com HealthLake

HealthLake infere dados encontrados no tipo de `DocumentReference` recurso usando o Amazon Comprehend Medical. O Amazon Comprehend Medical API opera e *`InferRxNorm`* detecta *`DetectEntities-V2`* condições *`InferICD10-CM`* médicas como características. Cada operação fornece informações diferentes.

> ⚠ Suporte para linguagens
>
> As API operações do Amazon Comprehend Medical só detectam entidades médicas em textos em inglês.

- DetectEntities-V2: inspeciona o texto clínico de uma variedade de entidades médicas e retorna informações específicas sobre elas, como categoria da entidade, localização e pontuação de confiança.
- Inferir ICD1 0-CM: Detecta condições médicas em um prontuário de paciente como entidades e vincula essas entidades a identificadores de conceitos normalizados na base de conhecimento ICD -10-CM do CDC National Center for Health Statistics, sob autorização da Organização Mundial da Saúde (). WHO
- InferRxNorm: Detecta medicamentos como entidades listadas em um prontuário de paciente e os vincula aos identificadores de conceitos normalizados no RxNorm banco de dados da National Library of Medicine.

As características suportadas para cada API operação são `SIGNSYMPTOM`, `DIAGNOSIS` e. Se características forem detectadas, elas serão FHIR adicionadas como extensões compatíveis em diferentes locais em seu armazenamento de HealthLake dados.

Locais onde as extensões são adicionadas.

- `DocumentReference`: Os resultados das API operações do Amazon Comprehend Medical são adicionados como um `extension` a cada documento encontrado dentro do tipo de recurso. DocumentReference Os resultados da extensão são divididos em dois grupos. Você pode encontrá-los nos resultados com base em seus`URL`.
  - `http://healthlake.amazonaws.com/system-generated-resources/`
    - Esses são tipos de recursos que foram criados ou adicionados por HealthLake.
  - `http://healthlake.amazonaws.com/aws-cm/`
    - Onde a saída bruta das API operações do Amazon Comprehend Medical é adicionada ao seu armazenamento de dados. HealthLake
- `Linkage`: esse tipo de recurso é adicionado ou criado como resultado da integraçãoNLP. Uma `GET` solicitação em um específico `Linkage` retorna uma lista de recursos vinculados. Para identificar se a `Linkage` foi adicionado por HealthLake, procure o par de `"tag": [{"display": "SYSTEM_GENERATED"}]` valores-chave adicionados. Para saber mais sobre as FHIR especificações do Linkage, consulte Tipo de recurso: Linkage no Índice de FHIRdocumentação.
- FHIRtipos de recursos gerados como resultado das operações do Amazon Comprehend Medical. API
  - `Observation`: Tem resultados das API operações DetectEntities -V2 e Infer ICD1 0-CM do Amazon Comprehend Medical adicionados quando as características são ou. `SIGN` `SYMPTOM`
  - `Condition`: Tem os resultados das API operações DetectEntities -V2 e Infer ICD1 0-CM do Amazon Comprehend Medical adicionados quando as características são. `DIAGNOSIS`
  - `MedicationStatement`: Tem os resultados da API operação InferRxNorm Amazon Comprehend Medical adicionados a ele.

Integração com as FHIR REST API operações

Por padrão, as características detectadas pelas API operações do Amazon Comprehend Medical não são retornadas ao fazer uma solicitação. `GET`

Para ver os resultados das NLP operações integradas desses tipos de recursos, você deve especificar um conhecido`ID`.

- `Linkage`
- `Observation`

- `Condition`
- `MedicationStatement`

Os resultados das NLP operações integradas fora do tipo de DocumentReference recurso só estão disponíveis usando uma `GET` solicitação em que se sabe que `ID` o especificado contém resultados das operações do Amazon Comprehend Medical. API

Exemplos de como as operações do Amazon Comprehend Medical API são integradas ao HealthLake

Exemplo 1: Registro do paciente ingerido em um armazenamento de HealthLake dados

Aqui está um exemplo de nota clínica baseada no encontro de um paciente com um profissional médico.

> ⚠ Dados sintéticos
>
> O texto neste exemplo tem conteúdo sintético e não contém informações pessoais de saúde (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 lembrete, essas informações são codificadas no formato base64 no recurso. DocumentReference Quando esse documento é inserido HealthLake e as operações do Amazon Comprehend API Medical são concluídas, para ver os resultados, você pode começar com a solicitação GET do tipo de recurso. `DocumentReference`

```
 GET https://https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/
r4/eeb8005725ae22b35b4edbdc68cf2dfd/r4/DocumentReference
```

Quando as API operações do Amazon Comprehend Medical forem bem-sucedidas, procure esses pares de valores-chave dentro do link a seguir `extension "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."
}
```

As guias a seguir mostram como o registro médico ingerido é relatado em seu armazenamento de HealthLake dados com base no tipo de recurso.

DocumentReference

Para ver os resultados de um único tipo de `DocumentReference` recurso, faça uma `GET` solicitação em que o `id` de um recurso específico seja fornecido.

```
GET https://https://healthlake.your-region.amazonaws.com/datastore/your-datastore-
id/r4/eeb8005725ae22b35b4edbdc68cf2dfd/r4/DocumentReference/0e938f03-da7f-4178-acd8-
eea9586c46ed
```

Quando bem-sucedido, você recebe um código de `200` HTTP resposta e a seguinte JSON resposta (que foi truncada para maior clareza).

Aqui está a `http://healthlake.amazonaws.com/system-generated-resources/` porção. Você pode ver que um novo `Linkage/`*`e366d29f-2c22-4c19-866e-09603937935a`* foi adicionado. Você também pode ver onde HealthLake foram adicionadas descobertas baseadas em inferências a tipos específicos `Observation` e de `Condition` recursos.

Para ver como esses tipos de recursos foram alterados, escolha as guias 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 os resultados de um único tipo de `Linkage` recurso, faça uma `GET` solicitação em que o `ID` de um recurso específico seja fornecido.

```
GET https://https://healthlake.your-region.amazonaws.com/
datastore/your-datastore-id/r4/eeb8005725ae22b35b4edbdc68cf2dfd/r4/
Linkage/e366d29f-2c22-4c19-866e-09603937935a
```

Quando bem-sucedido, você recebe um código de `200` HTTP resposta e a seguinte resposta truncadaJSON.

A resposta contém o `item` elemento. Nele, o par de valores-chave `"type": "source"` indica a `DocumentReference` entrada específica usada para modificar `Condition` e `Observations` listada sob o par de `"type": "alternate"` valores-chave.

Você também vê o *`meta`* elemento e um par de valores-chave correspondente*`"tag": [{"display": "SYSTEM_GENERATED"}]`*, indicando que esses recursos foram criados 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 os resultados de um único tipo de `Observation` recurso, faça uma `GET` solicitação em que o `ID` de um recurso específico seja fornecido.

```
GET https://https://healthlake.your-region.amazonaws.com/
datastore/your-datastore-id/r4/eeb8005725ae22b35b4edbdc68cf2dfd/r4/
Observation/e366d29f-2c22-4c19-866e-09603937935a
```

Os resultados das API operações do Amazon Comprehend Medical são alterados para os seguintes elementos:`code`,, e. `meta` `modifierExtension`

`code`

Um elemento do tipo`CodeableConcept`. Para saber mais, consulte CodeableConcepto Índice de FHIR documentação.

HealthLake acrescenta os três pares de valores-chave a seguir.

- `"system": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/"`: Onde URL se refere a uma operação específica do Amazon Comprehend Medical. API Nesse caso, inferir ICD1 0CM.
- `"code": "A52.06"`: Onde `A52.06` está o código ICD -10-CM que identifica o conceito encontrado na base de conhecimento dos Centros de Controle de Doenças.
- `"display": "Other syphilitic heart involvement"`: Onde `"Other syphilitic heart involvement"` está a descrição longa do código ICD -10-CM na ontologia.

A JSON resposta truncada a seguir contém somente o `code` elemento.

```
"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 a confiança do modelo de que o código ICD -10-CM atribuído está correto, use o `modifierExtension` elemento.

`meta`

O `meta` elemento contém metadados que indicam se o `code` elemento contém detalhes que foram adicionados pelas operações do Amazon API Comprehend Medical.

A JSON resposta truncada a seguir contém somente o `meta` elemento.

```
"meta": {
  "lastUpdated": "2022-10-21T19:38:30.879Z",
  "tag": [{
   "display": "SYSTEM_GENERATED"
  }]
 }
```

`modifierExtension`

O `modifierExtension` elemento contém mais detalhes sobre o nível de confiança dos códigos atribuídos encontrados no `code` elemento. Ele também tem pares de valores-chave que fornecem um link para o original DocumentReference usado para gerar os resultados e o tipo de recurso Linkage relacionado.

Para cada `coding` elemento adicionado, você verá um `entity-score` e um `entity-Concept-Score` adicionados aomodifierExtension. Para cada valor no par de valores-chave, você vê uma pontuação. Pois`entity-score`, essa pontuação é o nível de confiança que o Amazon Comprehend Medical tem na precisão da detecção. Pois`entity-Concept-Score`, essa pontuação é o nível de confiança que o Amazon Comprehend Medical tem de que a entidade está vinculada com precisão a um conceito de -10 CM. ICD

A JSON resposta truncada a seguir contém somente o `modifierExtension` elemento.

```
"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"
   }
  }
 ]
```

JSONResposta completa

```
{
  "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 os resultados de um único tipo de `Condition` recurso, faça uma `GET` solicitação em que o `ID` de um recurso específico seja fornecido.

```
GET https://https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/eeb8005725ae22b35b4edbdc68cf2dfd/r4/Condition/b06d343d-ddb8-4f36-82cb-853fcd434dfd
```

Os resultados das API operações do Amazon Comprehend Medical são alterados para os seguintes elementos:`code`,, e. `meta` `modifierExtension`

`code`

Um elemento do tipo`CodeableConcept`. Para saber mais, consulte CodeableConcepto Índice de FHIR documentação.

HealthLake acrescenta os três pares de valores-chave a seguir.

- `"system": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/"`: Onde URL se refere a uma operação específica do Amazon Comprehend Medical. API Nesse caso, inferir ICD1 0CM.
- `"code": "I70.0"`: Onde `A52.06` está o código ICD -10-CM que identifica o conceito encontrado na base de conhecimento dos Centros de Controle de Doenças.
- `"display": "Atherosclerosis of aorta"`: Onde `"Other syphilitic heart involvement"` está a descrição longa do código ICD -10-CM na ontologia.

A JSON resposta truncada a seguir contém somente o `code` elemento.

```
"code": {
  "coding":
  [
    {
   "system": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/",
   "code": "I70.0",
   "display": "Atherosclerosis of aorta"
    }
  ],
  "text": "Atherosclerosis of aorta"
 }
```

Para entender a confiança do modelo de que o código ICD -10-CM atribuído está correto, use o `modifierExtension` elemento.

`meta`

O `meta` elemento contém metadados que indicam se o `code` elemento contém detalhes que foram adicionados pelas operações do Amazon API Comprehend Medical.

A JSON resposta truncada a seguir contém somente o `meta` elemento.

```
"meta": {
  "lastUpdated": "2022-10-21T19:38:30.877Z",
  "tag": [{
   "display": "SYSTEM_GENERATED"
  }]
 }
```

`modifierExtension`

O `modifierExtension` elemento contém mais detalhes sobre o nível de confiança dos códigos atribuídos encontrados no `code` elemento. Ele também tem pares de valores-chave que fornecem um link para o original DocumentReference usado para gerar os resultados e o tipo de recurso Linkage relacionado.

Para cada `coding` elemento adicionado, você verá um `entity-score` e um `entity-Concept-Score` adicionados aomodifierExtension. Para cada valor no par de valores-chave, você vê uma pontuação. Poi`sentity-score`, essa pontuação é o nível de confiança que o Amazon Comprehend Medical tem na precisão da detecção. Poi`sentity-Concept-Score`, essa pontuação é o nível de confiança que o Amazon Comprehend Medical tem de que a entidade está vinculada com precisão a um conceito de -10 CM. ICD

A JSON resposta truncada a seguir contém somente o `modifierExtension` elemento.

```
"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"
   }
  }
 ]
```

JSONResposta completa

```
{
  "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"
 }
```


Exemplo 2: A **`DocumentReference`** que contém o tipo MedicationStatement de recurso

Aqui está um exemplo de uma nota clínica baseada no encontro de um paciente com um profissional médico.

> ⚠ Dados sintéticos
>
> O texto neste exemplo tem conteúdo sintético e não contém informações pessoais de saúde (PHI).

```
Tom is not prescribed Advil
```

As guias a seguir mostram como o registro médico ingerido é relatado em seu armazenamento de HealthLake dados com base no tipo de recurso.

DocumentReference

Para ver os resultados de um único tipo de `DocumentReference` recurso, faça uma `GET` solicitação em que o `ID` de um recurso específico seja fornecido.

```
GET https://https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/eeb8005725ae22b35b4edbdc68cf2dfd/r4/DocumentReference/c549125d-a218-421f-b8bf-23614c5e796c
```

Em caso de sucesso, você recebe um código de `200` HTTP resposta e a seguinte resposta truncadaJSON.

O par de valores-chave,`"url": "http://healthlake.amazonaws.com/system-generated-resources/"`, indica que os tipos de recursos dentro dele `extension` foram adicionados pelas operações do Amazon Comprehend Medical. API Você pode ver o novo tipo de `Linkage` recurso e vários `MedicationStatement` recursos.


```
"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 os resultados de um único tipo de `Linkage` recurso, faça uma `GET` solicitação em que o `ID` de um recurso específico seja fornecido.

```
GET https://https://healthlake.your-region.amazonaws.com/
datastore/your-datastore-id/r4/eeb8005725ae22b35b4edbdc68cf2dfd/r4/
Linkage/394bb244-177b-4409-8657-26b20ed56dd7
```

Em caso de sucesso, você recebe um código de `200` HTTP resposta e a seguinte JSON resposta.

A resposta contém o `item` elemento. Nele, o par de valores-chave `"type": "source"` indica a `DocumentReference` entrada específica usada para modificar os tipos de `MedicationStatement` recursos.

Você também pode ver o `meta` elemento e um par de valores-chave correspondente`"tag": [{"display": "SYSTEM_GENERATED"}]`, indicando que esses recursos foram criados 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 os resultados de um único tipo de `MedicationStatement` recurso, faça uma `GET` solicitação em que o `ID` de um recurso específico seja fornecido.

```
GET https://https://healthlake.your-region.amazonaws.com/
datastore/your-datastore-id/r4/eeb8005725ae22b35b4edbdc68cf2dfd/r4/
MedicationStatement/9a89b0d3-6681-45ca-9926-27951edce5c7
```

O tipo de MedicationStatement recurso é onde os resultados da operação Amazon Comprehend Medical InferRxNorm API são encontrados. Os resultados são alterados para os seguintes elementos: `medicationCodeableConceptmeta`, `modifierExtension` e.

medicationCodeableConcept

Um elemento do tipo`CodeableConcept`. Para saber mais, consulte CodeableConcepto Índice de FHIR documentação.

HealthLake acrescenta os três pares de valores-chave a seguir.

- `"system": ""http://healthlake.amazonaws.com/aws-cm/infer-rxnorm/`: Onde URL se refere a uma operação específica do Amazon Comprehend Medical. API Nesse caso, InferRxNorm.
- `"code": "731533"`: Onde `731533` está um ID RxNorm conceitual, também conhecido como RxCUI.
- `"display": "ibuprofen 200 MG Oral Capsule [Advil]"`: Onde `ibuprofen 200 MG Oral Capsule [Advil]` está a descrição do RxNorm conceito.

A JSON resposta truncada a seguir contém somente o `MedicationStatement` elemento.

```
"medicationCodeableConcept": {
  "coding": [
   {
    "system": "http://healthlake.amazonaws.com/aws-cm/infer-rxnorm/",
    "code": "731533",
    "display": "ibuprofen 200 MG Oral Capsule [Advil]"
   }
  ]
 }
```

meta

O `meta` elemento contém metadados que indicam se o `code` elemento contém detalhes que foram adicionados pelas operações do Amazon API Comprehend Medical.

A JSON resposta truncada a seguir contém somente o `meta` elemento.

```
"meta": {
  "lastUpdated": "2022-10-24T20:05:02.800Z",
  "tag": [
   {
    "display": "SYSTEM_GENERATED"
   }
  ]
 }
```

modifierExtension

O `modifierExtension` elemento contém pares de valores-chave que fornecem um link para o original DocumentReference usado para gerar os resultados e o tipo de recurso Linkage 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"
    }
   }
 ]
```

Parâmetros de pesquisa

A tabela a seguir lista os atributos pesquisáveis para medicina NLP integrada.

Parâmetros de pesquisa

Parâmetros de pesquisa	Encontra correspondências para
detectEntities-entidade-categoria	Categoria de entidade dentro da DetectEntities subextensão dentro da extensão AWS CM
detectEntities-entidade-texto	Texto da entidade dentro da DetectEntities subextensão dentro da extensão AWS CM
detectEntities-tipo de entidade	Tipo de entidade dentro da DetectEntities subextensão dentro da extensão AWS CM
detectEntities-pontuação da entidade	Pontuação da entidade dentro da DetectEntities subextensão dentro da extensão AWS CM
infer-icd10 cm-entity-text	Texto da entidade dentro da subextensão Infer ICD10CM dentro da extensão CM AWS
infer-icd10 cm-entity-score	Pontuação da entidade dentro da subextensão Infer ICD1 0CM dentro da extensão CM AWS
infer-icd10 cm-entity-concept-code	Código de conceito de entidade dentro da subextensão Infer ICD1 0CM dentro da extensão CM AWS
infer-icd10 cm-entity-concept-description	Descrição do conceito de entidade dentro da subextensão Infer ICD1 0CM dentro da extensão CM AWS
infer-icd10 cm-entity-concept-score	Pontuação do conceito de entidade dentro da subextensão Infer ICD1 0CM dentro da extensão CM AWS
infer-rxnorm-entity-score	Pontuação da entidade dentro da InferRxNorm subextensão dentro da extensão AWS CM
infer-rxnorm-entity-text	Texto da entidade dentro da InferRxNorm subextensão dentro da extensão AWS CM

Parâmetros de pesquisa	Encontra correspondências para
infer-rxnorm-entity-concept-código	Código de conceito de entidade dentro da InferRxNorm subextensão dentro da extensão AWS CM
infer-rxnorm-entity-concept-descrição	Descrição do conceito de entidade dentro da InferRxNorm subextensão dentro da extensão AWS CM
infer-rxnorm-entity-concept-pontuação	Pontuação do conceito de entidade dentro da InferRxNorm subextensão dentro da extensão AWS CM

Para corresponder aos critérios em `EntityCategory` que `EntityText` e fazem parte da mesma entidade, HealthLake fornece uma pesquisa especial. A tabela a seguir descreve os parâmetros de pesquisa especiais que são suportados no HealthLake.

Parâmetros de pesquisa

Parâmetros de pesquisa	Partidas devolvidas
detectEntities-entity-text-category	Se houver pelo menos uma entidade na DetectEntities subextensão que corresponda ao entityText e. entityCategory
detectEntities-entity-type-score	Se houver pelo menos uma entidade na DetectEntities subextensão que corresponda ao entityType e. entityScore
detectEntities-entity-text-score	Se houver pelo menos uma entidade na DetectEntities subextensão que corresponda ao entityText e. entityScore
detectEntities-entity-text-type	Se houver pelo menos uma entidade na DetectEntities subextensão que corresponda ao entityText e. entityType

Parâmetros de pesquisa	Partidas devolvidas
detectEntities-entity-category-score	Se houver pelo menos uma entidade que correspon da ao entityCategory entityScore e.
código cm-entity-text-concept infer-icd10	Se houver pelo menos uma entidade na subextens ão Infer ICD1 0CM que corresponda à entityText e houver pelo menos uma conceptCode para essa entidade que corresponda ao código.
infer-icd10 - pontuação cm-entity-text-con cept	Se houver pelo menos uma entidade na subextens ão Infer ICD1 0CM que corresponda à entityText e houver pelo menos uma conceptScore para essa entidade que corresponda à pontuação.
cm-entity-concept-descriptioninfer-icd10 - pontuação conceitual	Se houver pelo menos um conceito dentro da entidade na subextensão Infer ICD1 0CM que corresponda à descrição do conceito e a. conceptSc ore
infer-rxnorm-entity-text-código conceitual	Se houver pelo menos uma entidade na InferRxNorm subextensão que corresponda ao entityText e houver pelo menos uma conceptCode para essa entidade que corresponda ao código.
infer-rxnorm-entity-text-pontuação conceitual	Se houver pelo menos uma entidade na InferRxNorm subextensão que corresponda à entityText e houver pelo menos uma conceptScore para essa entidade que corresponda à pontuação.
infer-rxnorm-entity-concept-description-concept-score	Se houver pelo menos um conceito dentro da entidade na InferRxNorm subextensão que correspon da à descrição do conceito e a. conceptScore

Segurança em AWS HealthLake

A segurança na nuvem AWS é a maior prioridade. Como AWS cliente, você se beneficia de uma arquitetura de data center e rede criada para atender aos requisitos das organizações mais sensíveis à segurança.

A segurança é uma responsabilidade compartilhada entre você AWS e você. O modelo de responsabilidade compartilhada descreve isso como segurança da nuvem e segurança na nuvem:

- A segurança da nuvem AWS é responsável por proteger a infraestrutura que executa AWS os serviços na AWS nuvem. AWS também fornece serviços que você pode usar com segurança. Auditores terceirizados testam e verificam regularmente a eficácia de nossa segurança como parte dos Programas de Conformidade Programas de AWS de . Para saber mais sobre os programas de conformidade aplicáveis HealthLake, consulte AWSServiços no escopo do programa de conformidade AWS .
- Segurança na nuvem — Sua responsabilidade é determinada pelo AWS serviço que você usa. Você também é responsável por outros fatores, incluindo a confidencialidade de seus dados, os requisitos da empresa e as leis e regulamentos aplicáveis.

Esta documentação ajuda você a entender como aplicar o modelo de responsabilidade compartilhada ao usar HealthLake. Os tópicos a seguir mostram como configurar para atender HealthLake aos seus objetivos de segurança e conformidade. Você também aprenderá a usar outros AWS serviços que ajudam a monitorar e proteger seus HealthLake recursos.

Tópicos

Proteção de dados em AWS HealthLake

O modelo de responsabilidade AWS compartilhada modelo se aplica à proteção de dados em AWS HealthLake. Conforme descrito neste modelo, AWS é responsável por proteger a infraestrutura global que executa todos os Nuvem AWS. Você é responsável por manter o controle sobre seu conteúdo hospedado nessa infraestrutura. Você também é responsável pelas tarefas de configuração e gerenciamento de segurança dos Serviços da AWS que usa. Para obter mais informações sobre privacidade de dados, consulte Privacidade de dados FAQ. Para obter informações sobre proteção de dados na Europa, consulte o Modelo de Responsabilidade AWS Compartilhada e GDPR a postagem no blog AWS de segurança.

Para fins de proteção de dados, recomendamos que você proteja Conta da AWS as credenciais e configure usuários individuais com AWS IAM Identity Center ou AWS Identity and Access Management (IAM). Dessa maneira, cada usuário receberá apenas as permissões necessárias para cumprir suas obrigações de trabalho. Recomendamos também que você proteja seus dados das seguintes formas:

- Use a autenticação multifator (MFA) com cada conta.
- UseSSL/TLSpara se comunicar com AWS os recursos. Exigimos TLS 1,2 e recomendamos TLS 1,3.
- Configure API e registre as atividades do usuário com AWS CloudTrail. Para obter informações sobre o uso de CloudTrail trilhas para capturar AWS atividades, consulte Como trabalhar com CloudTrail trilhas no Guia AWS CloudTrail do usuário.
- Use soluções de AWS criptografia, juntamente com todos os controles de segurança padrão Serviços da AWS.
- Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a localizar e proteger dados sigilosos armazenados no Amazon S3.
- Se você precisar de FIPS 140-3 módulos criptográficos validados ao acessar AWS por meio de uma interface de linha de comando ou umaAPI, use um endpoint. FIPS Para obter mais informações sobre os FIPS endpoints disponíveis, consulte Federal Information Processing Standard (FIPS) 140-3.

É altamente recomendável que nunca sejam colocadas informações de identificação confidenciais, como endereços de e-mail dos seus clientes, em marcações ou campos de formato livre, como um campo Nome. Isso inclui quando você trabalha com HealthLake ou Serviços da AWS usa o console,API, AWS CLI, ou AWS SDKs. Quaisquer dados inseridos em tags ou campos de texto de

formato livre usados para nomes podem ser usados para logs de faturamento ou de diagnóstico. Se você fornecer um URL para um servidor externo, é altamente recomendável que você não inclua informações de credenciais no URL para validar sua solicitação para esse servidor.

Criptografia em REST para AWS HealthLake

HealthLake fornece criptografia por padrão para proteger dados confidenciais do cliente em repouso usando uma chave de serviço de gerenciamento de AWS chaves (AWSKMS) de propriedade do serviço. As KMS chaves gerenciadas pelo cliente também são suportadas e são necessárias para importar e exportar arquivos de um armazenamento de dados. Para saber mais sobre a KMS chave gerenciada pelo cliente, consulte Amazon Key Management Service. Os clientes podem escolher uma KMS chave AWS própria ou uma KMS chave gerenciada pelo cliente ao criar um armazenamento de dados. A configuração de criptografia não pode ser alterada após a criação de um armazenamento de dados. Se um armazenamento de dados estiver usando AWS uma KMS chave própria, ela será indicada como AWS_OWNED_KMS_KEY e você não verá a chave específica usada para criptografia em repouso.

AWSKMSchave de propriedade

HealthLake usa essas chaves por padrão para criptografar automaticamente informações potencialmente confidenciais, como dados de identificação pessoal ou dados de Informações de Saúde Privadas (PHI) em repouso. AWSKMSas chaves de propriedade não são armazenadas em sua conta. Eles fazem parte de uma coleção de KMS chaves que AWS possui e gerencia para uso em várias AWS contas. AWSos serviços podem usar KMS chaves AWS próprias para proteger seus dados. Você não pode visualizar, gerenciar, usar KMS chaves AWS próprias ou auditar seu uso. No entanto, você não precisa fazer nenhum trabalho nem alterar nenhum programa para proteger as chaves que criptografam seus dados.

Não é cobrada uma taxa mensal ou taxa de uso se você usar KMS chaves AWS próprias, e elas não contam nas AWS KMS cotas da sua conta. Para obter mais informações, consulte chaves AWS próprias.

Chaves KMS gerenciadas pelo cliente

HealthLake suporta o uso de uma KMS chave simétrica gerenciada pelo cliente que você cria, possui e gerencia para adicionar uma segunda camada de criptografia sobre a criptografia existenteAWS. Como você tem controle total dessa camada de criptografia, é possível realizar tarefas como:

- Estabelecer e manter as principais IAM políticas, políticas e subsídios
- Rotacionar os materiais de chave de criptografia
- Habilitar e desabilitar políticas de chaves
- Adicionar etiquetas
- Criar réplicas de chaves
- Programar chaves para exclusão

Você também pode usar CloudTrail para rastrear as solicitações HealthLake enviadas AWS KMS em seu nome. AWS KMS Cobranças adicionais se aplicam. Para obter mais informações, consulte as chaves de propriedade do cliente.

Criar uma chave gerenciada pelo cliente

Você pode criar uma chave simétrica gerenciada pelo cliente usando o AWS Management Console ou o. AWS KMS APIs

Siga as etapas para criar uma chave simétrica gerenciada pelo cliente no Guia do desenvolvedor do AWS Key Management Service.

As políticas de chaves controlam o acesso à chave gerenciada pelo cliente. Cada chave gerenciada pelo cliente deve ter exatamente uma política de chaves, que contém declarações que determinam quem pode usar a chave e como pode usá-la. Ao criar a chave gerenciada pelo cliente, é possível especificar uma política de chaves. Para obter mais informações, consulte Administre o acesso a chaves gerenciadas pelo cliente no Guia do desenvolvedor da AWS Key Management Service.

Para usar sua chave gerenciada pelo cliente com seus HealthLake recursos, CreateGrant as operações kms: devem ser permitidas na política de chaves. Isso adiciona uma concessão a uma chave gerenciada pelo cliente que controla o acesso a uma KMS chave especificada, o que dá ao usuário acesso às operações kms:grant necessárias. HealthLake Consulte Uso de subsídios para obter mais informações.

Para usar sua KMS chave gerenciada pelo cliente com seus HealthLake recursos, as seguintes API operações devem ser permitidas na política de chaves:

- kms: CreateGrant adiciona concessões a uma KMS chave específica gerenciada pelo cliente, que permite o acesso às operações de concessão.
- kms: DescribeKey fornece os detalhes da chave gerenciada pelo cliente necessários para validar a chave. Isso é necessário para todas as operações.

- kms: GenerateDataKey fornece acesso para criptografar recursos em repouso para todas as operações de gravação.
- O KMS:Decrypt fornece acesso às operações de leitura ou pesquisa de recursos criptografados.

Veja a seguir um exemplo de declaração de política que permite que um usuário crie e interaja com um armazenamento de AWS HealthLake dados criptografado por essa chave:

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

IAMPermissões necessárias para usar uma KMS chave gerenciada pelo cliente

Ao criar um armazenamento de dados com AWS KMS criptografia habilitada usando uma KMS chave gerenciada pelo cliente, há permissões necessárias tanto para a política de chaves quanto para a IAM política do usuário ou função que está criando o armazenamento de HealthLake dados.

Você pode usar a chave de ViaService condição kms: para limitar o uso da KMS chave somente às solicitações originadas de. HealthLake

Para obter mais informações sobre as principais políticas, consulte Habilitando IAM políticas no Guia do desenvolvedor do AWS Key Management Service.

O IAM usuário, a IAM função ou a AWS conta que está criando seus repositórios deve ter as kms: CreateGrant permissões kms: eGenerateDataKey, além das kms: DescribeKey permissões necessárias HealthLake.

Como HealthLake usa subsídios em AWS KMS

HealthLake exige uma concessão para usar sua KMS chave gerenciada pelo cliente. Quando você cria um armazenamento de dados criptografado com uma KMS chave gerenciada pelo cliente, HealthLake cria uma concessão em seu nome enviando uma CreateGrantsolicitação para AWSKMS. As concessões AWS KMS são usadas para dar HealthLake acesso a uma KMS chave na conta do cliente.

Os subsídios HealthLake criados em seu nome não devem ser revogados ou retirados. Se você revogar ou retirar a concessão que dá HealthLake permissão para usar as AWS KMS chaves em sua conta, HealthLake não poderá acessar esses dados, criptografar novos FHIR recursos enviados para o armazenamento de dados ou descriptografá-los quando forem retirados. Quando você revoga ou retira um subsídio HealthLake, a alteração ocorre imediatamente. Para revogar os direitos de acesso, você deve excluir o armazenamento de dados em vez de revogar a concessão. Quando um armazenamento de dados é excluído, HealthLake as concessões são retiradas em seu nome.

Monitoramento das suas chaves de criptografia para HealthLake

Você pode usar CloudTrail para rastrear as solicitações HealthLake enviadas AWS KMS em seu nome ao usar uma KMS chave gerenciada pelo cliente. As entradas de registro no CloudTrail registro mostram healthlake.amazonaws.com no userAgent campo para distinguir claramente as solicitações feitas por. HealthLake

Os exemplos a seguir são CloudTrail eventos para CreateGrant, GenerateDataKey, Decrypt e DescribeKey para monitorar AWS KMS operações chamadas por HealthLake para acessar dados criptografados pela chave gerenciada pelo cliente.

A seguir, mostramos como usar CreateGrant HealthLake para permitir o acesso a uma KMS chave fornecida pelo cliente, permitindo usar essa KMS chave HealthLake para criptografar todos os dados do cliente em repouso.

Os usuários não precisam criar suas próprias concessões. HealthLake cria uma concessão em seu nome enviando uma CreateGrant solicitação para AWSKMS. As concessões AWS KMS são usadas para dar HealthLake acesso a uma AWS KMS chave na conta de um 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"
}
```

Os exemplos a seguir mostram como usar GenerateDataKey para garantir que o usuário tenha as permissões necessárias para criptografar dados antes de armazená-los.

```
            {
    "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"
}
```


O exemplo a seguir mostra como HealthLake chama a operação Decrypt para usar a chave de dados criptografada armazenada para acessar os dados criptografados.

```
         {
    "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"
}
```

O exemplo a seguir mostra como HealthLake usa a DescribeKey operação para verificar se a AWS KMS chave de propriedade do AWS KMS cliente está em um estado utilizável e para ajudar o usuário a solucionar problemas se ela não estiver funcionando.

```
         {
    "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"
}
```

Saiba mais

Os recursos a seguir fornecem mais informações sobre criptografia de dados em repouso.

Para obter mais informações sobre os conceitos básicos do AWS Key Management Service, consulte a AWS KMS documentação.

Para obter mais informações sobre as melhores práticas de segurança na AWS KMS documentação.

Criptografia em trânsito para AWS HealthLake

AWS HealthLake usa TLS 1.2 para criptografar dados em trânsito pelo endpoint público e por meio de serviços de back-end.

Gerenciamento de identidade e acesso para AWS HealthLake

AWS Identity and Access Management (IAM) é uma ferramenta AWS service (Serviço da AWS) que ajuda o administrador a controlar com segurança o acesso aos AWS recursos. IAMos

administradores controlam quem pode ser autenticado (conectado) e autorizado (tem permissões) a usar HealthLake os recursos. IAMé um AWS service (Serviço da AWS) que você pode usar sem custo adicional.

Tópicos

Público

A forma como você usa AWS Identity and Access Management (IAM) difere, dependendo do trabalho que você faz HealthLake.

Usuário do serviço — Se você usar o HealthLake serviço para realizar seu trabalho, seu administrador fornecerá as credenciais e as permissões de que você precisa. À medida que você usa mais HealthLake recursos para fazer seu trabalho, talvez precise de permissões adicionais. Entender como o acesso é gerenciado pode ajudá-lo a solicitar as permissões corretas ao seu administrador. Se não for possível acessar um atributo no HealthLake, consulte Solução de problemas AWS HealthLake de identidade e acesso.

Administrador de serviços — Se você é responsável pelos HealthLake recursos da sua empresa, provavelmente tem acesso total HealthLake a. É seu trabalho determinar quais HealthLake recursos e recursos seus usuários do serviço devem acessar. Assim, é necessário enviar solicitações ao administrador do IAM para alterar as permissões dos usuários de seu serviço. Revise as informações nesta página para entender os conceitos básicos do IAM. Para saber mais sobre como sua empresa pode usar IAM com HealthLake, consulteComo AWS HealthLake funciona com IAM.

IAMadministrador — Se você for IAM administrador, talvez queira saber detalhes sobre como criar políticas para gerenciar o acesso HealthLake. Para ver exemplos de políticas HealthLake baseadas em identidade que você pode usar emIAM, consulte. Exemplos de políticas baseadas em identidade para AWS HealthLake

Autenticação com identidades

A autenticação é a forma como você faz login AWS usando suas credenciais de identidade. Você deve estar autenticado (conectado AWS) como IAM usuário ou assumindo uma IAM função. Usuário raiz da conta da AWS

Você pode entrar AWS como uma identidade federada usando credenciais fornecidas por meio de uma fonte de identidade. AWS IAM Identity Center Os usuários (do IAM Identity Center), a autenticação de login único da sua empresa e suas credenciais do Google ou do Facebook são exemplos de identidades federadas. Quando você faz login como uma identidade federada, o administrador já configurou anteriormente a federação de identidades usando funções do IAM. Ao acessar AWS usando a federação, você está assumindo indiretamente uma função.

Dependendo do tipo de usuário que você é, você pode entrar no AWS Management Console ou no portal de AWS acesso. Para obter mais informações sobre como fazer login em AWS, consulte Como fazer login Conta da AWS no Guia do Início de Sessão da AWS usuário.

Se você acessar AWS programaticamente, AWS fornece um kit de desenvolvimento de software (SDK) e uma interface de linha de comando (CLI) para assinar criptograficamente suas solicitações usando suas credenciais. Se você não usa AWS ferramentas, você mesmo deve assinar as solicitações. Para obter mais informações sobre como usar o método recomendado para você mesmo assinar solicitações, consulte AWS Signature versão 4 para API solicitações no Guia IAM do usuário.

Independentemente do método de autenticação usado, também pode ser exigido que você forneça informações adicionais de segurança. Por exemplo, AWS recomenda que você use a autenticação multifator (MFA) para aumentar a segurança da sua conta. Para saber mais, consulte Autenticação multifator no Guia do AWS IAM Identity Center usuário e Autenticação AWS multifator IAM no Guia do IAMusuário.

Conta da AWS usuário root

Ao criar uma Conta da AWS, você começa com uma identidade de login que tem acesso completo a todos Serviços da AWS os recursos da conta. Essa identidade é chamada de usuário Conta da AWS raiz e é acessada fazendo login com o endereço de e-mail e a senha que você usou para criar a conta. É altamente recomendável não usar o usuário-raiz para tarefas diárias. Proteja as credenciais do usuário-raiz e use-as para executar as tarefas que somente ele puder executar. Para obter a lista completa de tarefas que requerem login como usuário raiz, consulte Tarefas que exigem credenciais de usuário raiz no Guia do usuário do IAM.

Identidade federada

Como prática recomendada, exija que usuários humanos, incluindo usuários que precisam de acesso de administrador, usem a federação com um provedor de identidade para acessar Serviços da AWS usando credenciais temporárias.

Uma identidade federada é um usuário do seu diretório de usuários corporativo, de um provedor de identidade da web AWS Directory Service, do diretório do Identity Center ou de qualquer usuário que acesse usando credenciais fornecidas Serviços da AWS por meio de uma fonte de identidade. Quando as identidades federadas são acessadas Contas da AWS, elas assumem funções, e as funções fornecem credenciais temporárias.

Para o gerenciamento de acesso centralizado, recomendamos usar o AWS IAM Identity Center. Você pode criar usuários e grupos no IAM Identity Center ou pode se conectar e sincronizar com um conjunto de usuários e grupos em sua própria fonte de identidade para uso em todos os seus Contas da AWS aplicativos. Para obter informações sobre o IAM Identity Center, consulte O que é o IAM Identity Center? no Guia do AWS IAM Identity Center usuário.

Grupos e usuários do IAM

Um IAMusuário é uma identidade dentro da sua Conta da AWS que tem permissões específicas para uma única pessoa ou aplicativo. Sempre que possível, recomendamos confiar em credenciais temporárias em vez de criar IAM usuários que tenham credenciais de longo prazo, como senhas e chaves de acesso. No entanto, se você tiver casos de uso específicos que exijam credenciais de longo prazo com IAM os usuários, recomendamos que você alterne as chaves de acesso. Para obter mais informações, consulte Alterne as chaves de acesso regularmente para casos de uso que exigem credenciais de longo prazo no Guia do usuário do IAM.

Um grupo do IAM é uma identidade que especifica uma coleção de usuários do IAM. Não é possível fazer login como um grupo. É possível usar grupos para especificar permissões para vários usuários de uma vez. Os grupos facilitam o gerenciamento de permissões para grandes conjuntos de usuários. Por exemplo, você pode ter um grupo chamado IAMAdminse conceder a esse grupo permissões para administrar IAM recursos.

Usuários são diferentes de perfis. Um usuário é exclusivamente associado a uma pessoa ou a uma aplicação, mas um perfil pode ser assumido por qualquer pessoa que precisar dele. Os usuários têm credenciais permanentes de longo prazo, mas os perfis fornecem credenciais temporárias. Para saber mais, consulte Casos de uso para IAM usuários no Guia IAM do usuário.

Funções do IAM

Uma IAMfunção é uma identidade dentro da sua Conta da AWS que tem permissões específicas. Ela é semelhante a um usuário do IAM, mas não está associada a uma pessoa específica. Para assumir temporariamente uma IAM função no AWS Management Console, você pode alternar de um usuário para uma IAM função (console). Você pode assumir uma função chamando uma AWS API operação AWS CLI or ou usando uma personalizadaURL. Para obter mais informações sobre métodos de uso de funções, consulte Métodos para assumir uma função no Guia IAM do usuário.

As funções do IAM com credenciais temporária são úteis nas seguintes situações:

- Acesso de usuário federado: para atribuir permissões a identidades federadas, você pode criar um perfil e definir permissões para ele. Quando uma identidade federada é autenticada, essa identidade é associada ao perfil e recebe as permissões definidas por ele. Para obter informações sobre funções para federação, consulte Criar uma função para um provedor de identidade terceirizado (federação) no Guia IAM do usuário. Se você usa o IAM Identity Center, configura um conjunto de permissões. Para controlar o que suas identidades podem acessar após a autenticação, o IAM Identity Center correlaciona o conjunto de permissões a uma função em. IAM Para obter informações sobre conjuntos de permissões, consulte Conjuntos de permissões no Guia do usuário do AWS IAM Identity Center .
- Permissões temporárias IAM de IAM usuário — Um usuário ou função pode assumir uma IAM função para assumir temporariamente permissões diferentes para uma tarefa específica.
- Acesso entre contas: é possível usar uma função do IAM para permitir que alguém (uma entidade principal confiável) em outra conta acesse recursos em sua conta. Os perfis são a principal forma de conceder acesso entre contas. No entanto, com alguns Serviços da AWS, você pode anexar uma política diretamente a um recurso (em vez de usar uma função como proxy). Para saber a diferença entre funções e políticas baseadas em recursos para acesso entre contas, consulte Acesso a recursos entre contas IAM no Guia do IAM usuário.
- Acesso entre serviços — Alguns Serviços da AWS usam recursos em outros Serviços da AWS. Por exemplo, quando você faz uma chamada em um serviço, é comum que esse serviço execute aplicativos na Amazon EC2 ou armazene objetos no Amazon S3. Um serviço pode fazer isso usando as permissões da entidade principal de chamada, usando um perfil de serviço ou um perfil vinculado a serviço.
  - Sessões de acesso direto (FAS) — Quando você usa um IAM usuário ou uma função para realizar ações em AWS, você é considerado principal. Ao usar alguns serviços, você pode executar uma ação que, em seguida, inicia outra ação em um serviço diferente. FASusa as permissões do diretor chamando um AWS service (Serviço da AWS), combinadas com a

solicitação AWS service (Serviço da AWS) para fazer solicitações aos serviços posteriores. FASas solicitações são feitas somente quando um serviço recebe uma solicitação que requer interações com outros Serviços da AWS ou com recursos para ser concluída. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer FAS solicitações, consulte Encaminhar sessões de acesso.

- Função de serviço — Uma função de serviço é uma IAMfunção que um serviço assume para realizar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para obter mais informações, consulte Criar uma função para delegar permissões a um AWS service (Serviço da AWS) no Guia do IAM usuário.
- Função vinculada ao serviço — Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um. AWS service (Serviço da AWS) O serviço pode assumir o perfil para executar uma ação em seu nome. As funções vinculadas ao serviço aparecem em você Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não pode editar as permissões para funções vinculadas ao serviço.

- Aplicativos em execução na Amazon EC2 — Você pode usar uma IAM função para gerenciar credenciais temporárias para aplicativos que estão sendo executados em uma EC2 instância e fazendo AWS CLI AWS API solicitações. É preferível fazer isso do que armazenar chaves de acesso na instância do EC2. Para atribuir uma AWS função a uma EC2 instância e disponibilizá-la para todos os aplicativos, você cria um perfil de instância anexado à instância. Um perfil de instância contém a função e permite que os programas que estejam em execução na instância do EC2 obtenham credenciais temporárias. Para obter mais informações, consulte Usar uma IAM função para conceder permissões a aplicativos executados em EC2 instâncias da Amazon no Guia IAM do usuário.

Gerenciar acesso usando políticas

Você controla o acesso AWS criando políticas e anexando-as a AWS identidades ou recursos. Uma política é um objeto AWS que, quando associada a uma identidade ou recurso, define suas permissões. AWS avalia essas políticas quando um principal (usuário, usuário raiz ou sessão de função) faz uma solicitação. As permissões nas políticas determinam se a solicitação será permitida ou negada. A maioria das políticas é armazenada AWS como JSON documentos. Para obter mais informações sobre a estrutura e o conteúdo dos documentos de JSON política, consulte Visão geral das JSON políticas no Guia IAM do usuário.

Os administradores podem usar AWS JSON políticas para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

Por padrão, usuários e perfis não têm permissões. Para conceder aos usuários permissões para executar ações nos recursos de que eles precisam, um administrador do IAM pode criar políticas do IAM. O administrador pode, então, adicionar as políticas do IAM às funções, e os usuários podem assumir as funções.

As políticas do IAM definem permissões para uma ação, independentemente do método usado para executar a operação. Por exemplo, suponha que você tenha uma política que permite a ação `iam:GetRole`. Um usuário com essa política pode obter informações de função do AWS Management Console AWS CLI, do ou do AWS API.

Políticas baseadas em identidade

Políticas baseadas em identidade são documentos de políticas de JSON permissões que você pode anexar a uma identidade, como um IAM usuário, grupo de usuários ou função. Essas políticas controlam quais ações os usuários e perfis podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte Definir IAM permissões personalizadas com políticas gerenciadas pelo cliente no Guia do IAMusuário.

As políticas baseadas em identidade podem ser categorizadas ainda adicionalmente como políticas em linha ou políticas gerenciadas. As políticas em linha são anexadas diretamente a um único usuário, grupo ou perfil. As políticas gerenciadas são políticas autônomas que você pode associar a vários usuários, grupos e funções em seu Conta da AWS. As políticas AWS gerenciadas incluem políticas gerenciadas e políticas gerenciadas pelo cliente. Para saber como escolher entre uma política gerenciada ou uma política em linha, consulte Escolher entre políticas gerenciadas e políticas em linha no Guia do IAMusuário.

Políticas baseadas no recurso

Políticas baseadas em recursos são documentos JSON de política que você anexa a um recurso. As políticas de confiança de função do IAM e as políticas de bucket do Amazon S3 são exemplos de políticas baseadas em recursos. Em serviços que suportem políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o recurso ao qual a política está anexada, a política define quais ações um principal especificado pode executar nesse recurso e em que condições. Você deve especificar uma entidade principal em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou. Serviços da AWS

Políticas baseadas em recursos são políticas em linha localizadas nesse serviço. Você não pode usar políticas AWS gerenciadas de uma política baseada IAM em recursos.

Listas de controle de acesso (ACLs)

As listas de controle de acesso (ACLs) controlam quais diretores (membros da conta, usuários ou funções) têm permissões para acessar um recurso. ACLssão semelhantes às políticas baseadas em recursos, embora não usem o formato de documento JSON de política.

Amazon S3, AWS WAF, e Amazon VPC são exemplos de serviços que oferecem suporte. ACLs Para saber maisACLs, consulte a visão geral da lista de controle de acesso (ACL) no Guia do desenvolvedor do Amazon Simple Storage Service.

Outros tipos de política

AWS oferece suporte a tipos de políticas adicionais menos comuns. Esses tipos de política podem definir o máximo de permissões concedidas a você pelos tipos de política mais comuns.

- Limites de permissões — Um limite de permissões é um recurso avançado no qual você define as permissões máximas que uma política baseada em identidade pode conceder a uma IAM entidade (IAMusuário ou função). É possível definir um limite de permissões para uma entidade. As permissões resultantes são a interseção das políticas baseadas em identidade de uma entidade com seus limites de permissões. As políticas baseadas em recurso que especificam o usuário ou o perfil no campo `Principal` não são limitadas pelo limite de permissões. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações sobre limites de permissões, consulte Limites de permissões para entidades do IAM no Guia do usuário do IAM.
- Políticas de controle de serviço (SCPs) — SCPs são JSON políticas que especificam as permissões máximas para uma organização ou unidade organizacional (OU) em AWS Organizations. AWS Organizations é um serviço para agrupar e gerenciar centralmente vários Contas da AWS que sua empresa possui. Se você habilitar todos os recursos em uma organização, poderá aplicar políticas de controle de serviço (SCPs) a qualquer uma ou a todas as suas contas. Os SCP limites de permissões para entidades nas contas dos membros, incluindo cada uma Usuário raiz da conta da AWS. Para obter mais informações sobre Organizations eSCPs, consulte Políticas de controle de serviços no Guia AWS Organizations do Usuário.
- Políticas de controle de recursos (RCPs) — RCPs são JSON políticas que você pode usar para definir o máximo de permissões disponíveis para recursos em suas contas sem atualizar as IAM políticas anexadas a cada recurso que você possui. Isso RCP limita as permissões de recursos nas contas dos membros e pode afetar as permissões efetivas das identidades, incluindo a Usuário raiz da conta da AWS, independentemente de pertencerem à sua organização. Para obter mais informações sobre Organizations eRCPs, incluindo uma lista Serviços da AWS desse

suporteRCPs, consulte Políticas de controle de recursos (RCPs) no Guia AWS Organizations do usuário.

- Políticas de sessão: são políticas avançadas que você transmite como um parâmetro quando cria de forma programática uma sessão temporária para um perfil ou um usuário federado. As permissões da sessão resultante são a interseção das políticas baseadas em identidade do usuário ou do perfil e das políticas de sessão. As permissões também podem ser provenientes de uma política baseada em recursos. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para ter mais informações, consulte Políticas de sessão no Guia do usuário do IAM.

Vários tipos de política

Quando vários tipos de política são aplicáveis a uma solicitação, é mais complicado compreender as permissões resultantes. Para saber como AWS determinar se uma solicitação deve ser permitida quando vários tipos de política estão envolvidos, consulte Lógica de avaliação de políticas no Guia IAM do usuário.

Como AWS HealthLake funciona com IAM

Antes de usar IAM para gerenciar o acesso ao HealthLake, saiba quais IAM recursos estão disponíveis para uso HealthLake.

IAMrecursos que você pode usar com AWS HealthLake

Recurso do IAM	HealthLake apoio
Políticas baseadas em identidade	Sim
Políticas baseadas em recursos	Não
Ações das políticas	Sim
Atributos de políticas	Sim
Chaves de condição de políticas	Sim
ACLs	Não
ABAC(tags nas políticas)	Sim

Recurso do IAM	HealthLake apoio
Credenciais temporárias	Sim
Permissões de entidade principal	Sim
Perfis de serviço	Sim
Perfis vinculados a serviços	Não

Para obter uma visão geral de como HealthLake e outros AWS serviços funcionam com a maioria dos IAM recursos, consulte AWS os serviços que funcionam com IAM no Guia do IAM usuário.

Políticas baseadas em identidade para AWS HealthLake

Compatível com políticas baseadas em identidade: sim

Políticas baseadas em identidade são documentos de políticas de JSON permissões que você pode anexar a uma identidade, como um IAM usuário, grupo de usuários ou função. Essas políticas controlam quais ações os usuários e perfis podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte Definir IAM permissões personalizadas com políticas gerenciadas pelo cliente no Guia do IAMusuário.

Com as políticas baseadas em identidade do IAM, é possível especificar ações ou recursos permitidos ou negados, bem como as condições sob as quais as ações são permitidas ou negadas. Você não pode especificar a entidade principal em uma política baseada em identidade porque ela se aplica ao usuário ou perfil ao qual ela está anexada. Para saber mais sobre todos os elementos que você pode usar em uma JSON política, consulte a referência IAM JSON de elementos de política no Guia IAM do usuário.

Exemplos de políticas baseadas em identidade para AWS HealthLake

Para ver exemplos de políticas HealthLake baseadas em identidade, consulte. Exemplos de políticas baseadas em identidade para AWS HealthLake

Políticas baseadas em recursos dentro AWS HealthLake

Suporte a políticas baseadas em recursos: não

Políticas baseadas em recursos são documentos JSON de política que você anexa a um recurso. As políticas de confiança de função do IAM e as políticas de bucket do Amazon S3 são exemplos de políticas baseadas em recursos. Em serviços que suportem políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o recurso ao qual a política está anexada, a política define quais ações um principal especificado pode executar nesse recurso e em que condições. Você deve especificar uma entidade principal em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou. Serviços da AWS

Para permitir o acesso entre contas, você pode especificar uma conta inteira ou as entidades do IAM em outra conta como o principal em uma política baseada em recurso. Adicionar uma entidade principal entre contas à política baseada em recurso é apenas metade da tarefa de estabelecimento da relação de confiança. Quando o principal e o recurso são diferentes Contas da AWS, um IAM administrador na conta confiável também deve conceder permissão à entidade principal (usuário ou função) para acessar o recurso. Eles concedem permissão ao anexar uma política baseada em identidade para a entidade. No entanto, se uma política baseada em recurso conceder acesso a uma entidade principal na mesma conta, nenhuma política baseada em identidade adicional será necessária. Para obter mais informações, consulte Acesso a recursos entre contas IAM no Guia do IAM usuário.

Ações políticas para AWS HealthLake

Compatível com ações de políticas: sim

Os administradores podem usar AWS JSON políticas para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O `Action` elemento de uma JSON política descreve as ações que você pode usar para permitir ou negar acesso em uma política. As ações de política geralmente têm o mesmo nome da AWS API operação associada. Há algumas exceções, como ações somente de permissão que não têm uma operação correspondente. API Algumas operações também exigem várias ações em uma política. Essas ações adicionais são chamadas de ações dependentes.

Incluem ações em uma política para conceder permissões para executar a operação associada.

Para ver uma lista de HealthLake ações, consulte Ações definidas por AWS HealthLake na Referência de Autorização de Serviço.

As ações de política HealthLake usam o seguinte prefixo antes da ação:

```
healthlake
```

Para especificar várias ações em uma única declaração, separe cada ação com uma vírgula.

```
"Action": [
      "healthlake:action1",
      "healthlake:action2"
         ]
```

Para ver exemplos de políticas HealthLake baseadas em identidade, consulte. Exemplos de políticas baseadas em identidade para AWS HealthLake

Recursos políticos para AWS HealthLake

Compatível com recursos de políticas: sim

Os administradores podem usar AWS JSON políticas para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O elemento `Resource` JSON de política especifica o objeto ou objetos aos quais a ação se aplica. As instruções devem incluir um elemento `Resource` ou `NotResource`. Como prática recomendada, especifique um recurso usando seu Amazon Resource Name (ARN). Isso pode ser feito para ações que oferecem compatibilidade com um tipo de recurso específico, conhecido como permissões em nível de recurso.

Para ações que não oferecem compatibilidade com permissões em nível de recurso, como operações de listagem, use um curinga (*) para indicar que a instrução se aplica a todos os recursos.

```
"Resource": "*"
```

Para ver uma lista dos tipos de HealthLake recursos e seusARNs, consulte Recursos definidos por AWS HealthLake na Referência de Autorização de Serviço. Para saber as ações com as quais você pode especificar cada recurso, consulte Ações definidas por AWS HealthLake. ARN

Para ver exemplos de políticas HealthLake baseadas em identidade, consulte. Exemplos de políticas baseadas em identidade para AWS HealthLake

Chaves de condição de política para AWS HealthLake

Compatível com chaves de condição de política específicas de serviço: sim

Os administradores podem usar AWS JSON políticas para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O elemento `Condition` (ou bloco `Condition`) permite que você especifique condições nas quais uma instrução estiver em vigor. O elemento `Condition` é opcional. É possível criar expressões condicionais que usem agentes de condição, como "igual a" ou "menor que", para fazer a condição da política corresponder aos valores na solicitação.

Se você especificar vários elementos `Condition` em uma instrução ou várias chaves em um único `Condition` elemento, a AWS os avaliará usando uma operação lógica `AND`. Se você especificar vários valores para uma única chave de condição, AWS avalia a condição usando uma `OR` operação lógica. Todas as condições devem ser atendidas antes que as permissões da instrução sejam concedidas.

Você também pode usar variáveis de espaço reservado ao especificar condições. Por exemplo, você pode conceder a um usuário do IAM permissão para acessar um recurso somente se ele estiver marcado com seu nome de usuário do IAM. Para obter mais informações, consulte Elementos de política do IAM: variáveis e tags no Guia do usuário do IAM.

AWS suporta chaves de condição globais e chaves de condição específicas do serviço. Para ver todas as chaves de condição AWS globais, consulte as chaves de contexto de condição AWS global no Guia IAM do usuário.

Para ver uma lista de chaves de HealthLake condição, consulte Chaves de condição AWS HealthLake na Referência de autorização de serviço. Para conhecer as ações e os recursos com os quais você pode usar uma chave de condição, consulte Ações definidas por AWS HealthLake.

Para ver exemplos de políticas HealthLake baseadas em identidade, consulte. Exemplos de políticas baseadas em identidade para AWS HealthLake

Listas de controle de acesso (ACLs) em AWS HealthLake

SuportesACLs: Não

As listas de controle de acesso (ACLs) controlam quais diretores (membros da conta, usuários ou funções) têm permissões para acessar um recurso. ACLssão semelhantes às políticas baseadas em recursos, embora não usem o formato de documento JSON de política.

Controle de acesso baseado em atributos () ABAC com AWS HealthLake

Suportes ABAC (tags nas políticas): Sim

O controle de acesso baseado em atributos (ABAC) é uma estratégia de autorização que define permissões com base em atributos. Em AWS, esses atributos são chamados de tags. Você pode anexar tags a IAM entidades (usuários ou funções) e a vários AWS recursos. Marcar entidades e recursos é a primeira etapa doABAC. Em seguida, você cria ABAC políticas para permitir operações quando a tag do diretor corresponde à tag do recurso que ele está tentando acessar.

ABACé útil em ambientes que estão crescendo rapidamente e ajuda em situações em que o gerenciamento de políticas se torna complicado.

Para controlar o acesso baseado em tags, forneça informações sobre as tags no elemento de condição de uma política usando as `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`* ou chaves de condição `aws:TagKeys`.

Se um serviço oferecer suporte às três chaves de condição para cada tipo de recurso, o valor será Sim para o serviço. Se um serviço oferecer suporte às três chaves de condição somente para alguns tipos de recursos, o valor será Parcial

Para obter mais informações sobreABAC, consulte Definir permissões com ABAC autorização no Guia IAM do usuário. Para ver um tutorial com etapas de configuraçãoABAC, consulte Usar controle de acesso baseado em atributos (ABAC) no Guia do IAMusuário.

Usando credenciais temporárias com AWS HealthLake

Compatível com credenciais temporárias: sim

Alguns Serviços da AWS não funcionam quando você faz login usando credenciais temporárias. Para obter informações adicionais, incluindo quais Serviços da AWS funcionam com credenciais temporárias, consulte Serviços da AWS esse trabalho IAM no Guia do IAM usuário.

Você está usando credenciais temporárias se fizer login AWS Management Console usando qualquer método, exceto um nome de usuário e senha. Por exemplo, quando você acessa AWS usando o link de login único (SSO) da sua empresa, esse processo cria automaticamente credenciais temporárias. Você também cria automaticamente credenciais temporárias quando faz login no console como usuário e, em seguida, alterna perfis. Para obter mais informações sobre a troca de funções, consulte Alternar de um usuário para uma IAM função (console) no Guia IAM do usuário.

Você pode criar manualmente credenciais temporárias usando o AWS CLI ou AWS API. Em seguida, você pode usar essas credenciais temporárias para acessar AWS. AWS recomenda que você gere credenciais temporárias dinamicamente em vez de usar chaves de acesso de longo prazo. Para obter mais informações, consulte Credenciais de segurança temporárias emIAM.

Permissões principais entre serviços para AWS HealthLake

Suporta sessões de acesso direto (FAS): Sim

Quando você usa um IAM usuário ou uma função para realizar ações em AWS, você é considerado principal. Ao usar alguns serviços, você pode executar uma ação que, em seguida, inicia outra ação em um serviço diferente. FASusa as permissões do diretor chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service (Serviço da AWS) para fazer solicitações aos serviços posteriores. FASas solicitações são feitas somente quando um serviço recebe uma solicitação que requer interações com outros Serviços da AWS ou com recursos para ser concluída. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer FAS solicitações, consulte Encaminhar sessões de acesso.

Funções de serviço para AWS HealthLake

Compatível com perfis de serviço: sim

Um perfil de serviço é um perfil do IAM que um serviço assume para realizar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para obter mais informações, consulte Criar uma função para delegar permissões a um AWS service (Serviço da AWS) no Guia do IAM usuário.

Para obter informações sobre funções de serviço e a política em linha necessária para acesso total AWS HealthLake, consulteConfigurando permissões para começar a usar AWS HealthLake.

> **Warning**
>
> Alterar as permissões de uma função de serviço pode interromper HealthLake a funcionalidade. Edite as funções de serviço somente quando HealthLake fornecer orientação para fazer isso.

Funções vinculadas a serviços para AWS HealthLake

Compatível com perfis vinculados ao serviço: Não

Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um. AWS service (Serviço da AWS) O serviço pode assumir o perfil para executar uma ação em seu nome. As funções vinculadas ao serviço aparecem em você Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não pode editar as permissões para funções vinculadas ao serviço.

Para obter detalhes sobre a criação ou o gerenciamento de funções vinculadas a serviço, consulte Serviços da AWS que funcionam com o IAM. Encontre um serviço na tabela que inclua um `Yes` na coluna Perfil vinculado ao serviço. Escolha o link Sim para visualizar a documentação do perfil vinculado a serviço desse serviço.

Exemplos de políticas baseadas em identidade para AWS HealthLake

Por padrão, usuários e perfis não têm permissão para criar ou modificar recursos do HealthLake. Eles também não podem realizar tarefas usando o AWS Management Console, AWS Command Line Interface (AWS CLI) ou AWS API. Para conceder aos usuários permissões para executar ações nos recursos de que eles precisam, um administrador do IAM pode criar políticas do IAM. O administrador pode, então, adicionar as políticas do IAM às funções, e os usuários podem assumir as funções.

Para saber como criar uma política IAM baseada em identidade usando esses exemplos de documentos de JSON política, consulte Criar IAM políticas (console) no Guia do IAMusuário.

Para obter detalhes sobre ações e tipos de recursos definidos por HealthLake, incluindo o formato do ARNs para cada um dos tipos de recursos, consulte Ações, recursos e chaves de condição AWS HealthLake na Referência de Autorização de Serviço.

Tópicos

- Melhores práticas de política
- Usar o console do AWS HealthLake
- Acessando um armazenamento AWS HealthLake de dados em Amazon Athena
- Permitir que os usuários visualizem suas próprias permissões

Melhores práticas de política

As políticas baseadas em identidade determinam se alguém pode criar, acessar ou excluir HealthLake recursos em sua conta. Essas ações podem incorrer em custos para seus Conta da AWS. Ao criar ou editar políticas baseadas em identidade, siga estas diretrizes e recomendações:

- Comece com as políticas AWS gerenciadas e avance para as permissões de privilégios mínimos — Para começar a conceder permissões aos seus usuários e cargas de trabalho, use as políticas AWS gerenciadas que concedem permissões para muitos casos de uso comuns. Eles estão disponíveis no seu Conta da AWS. Recomendamos que você reduza ainda mais as permissões definindo políticas gerenciadas pelo AWS cliente que sejam específicas para seus casos de uso. Para obter mais informações, consulte Políticas gerenciadas da AWS ou Políticas gerenciadas da AWS para funções de trabalho no Guia do usuário do IAM.
- Aplique permissões de privilégio mínimo: ao definir permissões com as políticas do IAM, conceda apenas as permissões necessárias para executar uma tarefa. Você faz isso definindo as ações que podem ser executadas em recursos específicos sob condições específicas, também conhecidas como permissões de privilégio mínimo. Para obter mais informações sobre o uso do IAM para aplicar permissões, consulte Políticas e permissões no IAM no Guia do usuário do IAM.
- Use condições nas políticas do IAM para restringir ainda mais o acesso: você pode adicionar uma condição às políticas para limitar o acesso a ações e recursos. Por exemplo, você pode escrever uma condição de política para especificar que todas as solicitações devem ser enviadas usandoSSL. Você também pode usar condições para conceder acesso às ações de serviço se elas forem usadas por meio de uma ação específica AWS service (Serviço da AWS), como AWS CloudFormation. Para obter mais informações, consulte Elementos IAM JSON da política: Condição no Guia IAM do usuário.
- Use o IAM Access Analyzer para validar suas IAM políticas e garantir permissões seguras e funcionais — o IAM Access Analyzer valida políticas novas e existentes para que as políticas sigam a linguagem da IAM política (JSON) e as melhores práticas. IAM IAM O Access Analyzer fornece mais de 100 verificações de políticas e recomendações práticas para ajudá-lo a criar políticas seguras e funcionais. Para obter mais informações, consulte Validar políticas com o IAM Access Analyzer no Guia do IAMUsuário.
- Exigir autenticação multifatorial (MFA) — Se você tiver um cenário que exija IAM usuários ou um usuário root Conta da AWS, ative MFA para obter segurança adicional. Para exigir MFA quando API as operações são chamadas, adicione MFA condições às suas políticas. Para obter mais informações, consulte APIAcesso seguro com MFA no Guia IAM do usuário.

Para obter mais informações sobre as práticas recomendadas no IAM, consulte Práticas recomendadas de segurança no IAM no Guia do usuário do IAM.

Usar o console do AWS HealthLake

Para acessar o AWS HealthLake console, você deve ter um conjunto mínimo de permissões. Essas permissões devem permitir que você liste e visualize detalhes sobre os HealthLake recursos em seu Conta da AWS. Caso crie uma política baseada em identidade mais restritiva que as permissões mínimas necessárias, o console não funcionará como pretendido para entidades (usuários ou perfis) com essa política.

Você não precisa permitir permissões mínimas do console para usuários que estão fazendo chamadas somente para AWS CLI o. ou AWS API o. Em vez disso, permita o acesso somente às ações que correspondam à API operação que eles estão tentando realizar.

Para obter acesso total a HealthLake, anexe as seguintes políticas a um IAM usuário ou função: `AmazonHealthLakeFullAccess` `AWSLakeFormationDataAdmin` e. Você também precisa anexar a política HealthLake embutida, que é uma função de serviço. Um perfil de serviço é um perfil do IAM que um serviço assume para realizar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para obter mais informações, consulte Criar uma função para delegar permissões a um AWS service (Serviço da AWS) no Guia do IAM usuário. Para obter informações sobre a política em linha que cria a função de serviço necessária, consulteConfigurando permissões para começar a usar AWS HealthLake. Você também deve usar o AWS Lake Formation console ou CLI designar seu HealthLake administrador para ser um administrador do AWS Lake Formation Data Lake. Para obter mais informações, consulte Configurando permissões para começar a usar AWS HealthLake.

Acessando um armazenamento AWS HealthLake de dados em Amazon Athena

Se você quiser fornecer aos usuários e funções acesso aos armazenamentos de HealthLake dados em Amazon Athena, anexe as seguintes IAM políticas à função ou ao usuário: `AmazonAthenaFullAccess` `AmazonS3FullAccess` e. `Selecte` `Describe` as permissões também são necessárias em tabelas gerenciadas pelo AWS Lake Formation. AWS Lake Formation as permissões de tabela são concedidas por um AWS Lake Formation administrador no AWS Lake Formation console ou por meio doCLI. Para ter mais informações, consulte Configurando permissões para começar a usar AWS HealthLake

Permitir que os usuários visualizem suas próprias permissões

Este exemplo mostra como você pode criar uma política que permite que os usuários do IAM visualizem as políticas gerenciadas e em linha anexadas a sua identidade de usuário. Essa política

inclui permissões para concluir essa ação no console ou programaticamente usando o AWS CLI ou. AWS API

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS políticas gerenciadas para AWS HealthLake

Uma política AWS gerenciada é uma política autônoma criada e administrada por AWS. AWS as políticas gerenciadas são projetadas para fornecer permissões para muitos casos de uso comuns, para que você possa começar a atribuir permissões a usuários, grupos e funções.

Lembre-se de que as políticas AWS gerenciadas podem não conceder permissões de privilégio mínimo para seus casos de uso específicos porque elas estão disponíveis para uso de todos os AWS clientes. Recomenda-se que as permissões sejam ainda mais reduzidas, definindo políticas gerenciadas pelo cliente da específicas para os casos de uso.

Você não pode alterar as permissões definidas nas políticas AWS gerenciadas. Se AWS atualizar as permissões definidas em uma política AWS gerenciada, a atualização afetará todas as identidades principais (usuários, grupos e funções) às quais a política está anexada. AWS é mais provável que atualize uma política AWS gerenciada quando uma nova AWS service (Serviço da AWS) é lançada ou novas API operações são disponibilizadas para os serviços existentes.

Para obter mais informações, consulte políticas gerenciadas AWS no Guia do Usuário IAM.

AWS política gerenciada: AmazonHealthLakeFullAccess

A `AmazonHealthLakeFullAccess` política fornece acesso total HealthLake a. Com essa política anexada ao usuário ou função, os usuários podem HealthLake usá-la para acessar, consultar, importar e exportar dados em HealthLake. Para realizar muitas ações comuns em HealthLake, você deve adicionar políticas adicionais ao usuário ou à função. Para obter mais informações, consulte HealthLake operações Configurando permissões para começar a usar AWS HealthLake e permissões.

É possível anexar a política `AmazonHealthLakeFullAccess` às suas identidades do IAM.

Essa política concede *`administrative and contributor`* permissões que permitem que usuários e funções consultem, pesquisem, importem e exportem HealthLake, além de possibilitar HealthLake a execução de ações em nome dos usuários e funções que têm essas permissões.

Detalhes da permissão

Essa política inclui a seguinte declaração.

```
{
 "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 gerenciada: AmazonHealthLakeReadOnlyAccess

`AmazonHealthLakeReadOnlyAccess`a política concede acesso e permissões somente para leitura HealthLake e recursos relacionados em outros AWS serviços. Aplique essa política aos usuários aos quais você deseja conceder a capacidade de consultar e visualizar armazenamentos de HealthLake dados, mas não a capacidade de criar ou fazer alterações neles.

É possível anexar a política `AmazonHealthLakeReadOnlyAccess` às suas identidades do IAM.

Essa política concede *`read-only`* permissões que permitem que usuários e funções consultem HealthLake.

Detalhes da permissão

Essa política inclui a seguinte declaração.

```
{
    "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 operações e permissões

A tabela a seguir lista as operações típicas HealthLake e as permissões necessárias para executá-las.

HealthLake operações	Permissões obrigatórias
Crie um armazenamento de dados em HealthLake	`AmazonHealthLakeFullAccess` ,`AmazonLakeFormationDataAdmin` , política em linha e permissões de AWS Lake Formation administrador gerenciadas por AWS Lake Formation
Excluir um armazenamento de dados no HealthLake	`AmazonHealthLakeFullAccess` ,`AmazonLakeFormationDataAdmin` , política em linha e permissõe

HealthLake operações	Permissões obrigatórias
	s de AWS Lake Formation administrador gerenciadas por AWS Lake Formation
Listar, pesquisar ou consultar um armazenamento de dados no HealthLake	`AmazonHealthLakeReadOnlyAccess`
Consulte um armazenamento de dados usando Amazon Athena	`AmazonAthenaFullAccess` ,`AmazonS3FullAccess` , AWS Lake Formation `Select` e `Describe` permissões em tabelas gerenciadas pelo AWS Lake Formation
Importar dados de HealthLake	Consulte Configurando permissões para trabalhos de importação.
Exportar dados de HealthLake	Consulte Exportando arquivos do seu armazenamento de dados ()AWS SDKs.

HealthLake atualizações nas políticas AWS gerenciadas

Veja detalhes sobre as atualizações das políticas AWS gerenciadas a HealthLake partir do momento em que esse serviço começou a rastrear essas alterações. Para receber alertas automáticos sobre alterações nessa página, assine o RSS feed na página Histórico do HealthLake documento.

Alteração	Descrição	Data
AmazonHealthLakeFullAccess	`AmazonHealthLakeFullAccess` política necessária para permitir acesso total HealthLake a.	14 de novembro de 2022
AmazonHealthLakeReadOnlyAccess	`AmazonHealthLakeReadOnlyAccess` política necessária para acesso somente para leitura a. HealthLake	14 de novembro de 2022

Alteração	Descrição	Data
HealthLake começou a rastrear as alterações	HealthLake começou a rastrear as mudanças em suas políticas AWS gerenciadas.	14 de novembro de 2022

Solução de problemas AWS HealthLake de identidade e acesso

Use as informações a seguir para ajudá-lo a diagnosticar e corrigir problemas comuns que você pode encontrar ao trabalhar com HealthLake e. IAM

Tópicos

- Não estou autorizado a realizar uma ação em AWS HealthLake
- Não estou autorizado a realizar iam: PassRole
- Quero permitir que pessoas fora da minha AWS conta acessem meus AWS HealthLake recursos

Não estou autorizado a realizar uma ação em AWS HealthLake

Se isso AWS Management Console indicar que você não está autorizado a realizar uma ação, entre em contato com o administrador para obter ajuda. O administrador é a pessoa que forneceu o seu nome de usuário e senha.

O exemplo de erro a seguir ocorre quando o `mateojackson` IAM usuário tenta usar o console para ver detalhes sobre um *`my-example-widget`* recurso fictício, mas não tem as permissões fictícias`healthlake:`*`GetWidget`*.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 healthlake:GetWidget on resource: my-example-widget
```

Neste caso, Mateo pede ao administrador para atualizar suas políticas e permitir o acesso ao recurso *`my-example-widget`* usando a ação `healthlake:`*`GetWidget`*.

Não estou autorizado a realizar iam: PassRole

Se você receber uma mensagem de erro informando que não está autorizado a executar a ação `iam:PassRole`, as suas políticas devem ser atualizadas para permitir que você passe uma função para o HealthLake.

Alguns Serviços da AWS permitem que você passe uma função existente para esse serviço em vez de criar uma nova função de serviço ou uma função vinculada ao serviço. Para fazer isso, é preciso ter permissões para passar o perfil para o serviço.

O erro de exemplo a seguir ocorre quando uma usuária do IAM chamada `marymajor` tenta usar o console para executar uma ação no HealthLake. No entanto, a ação exige que o serviço tenha permissões concedidas por um perfil de serviço. Mary não tem permissões para passar o perfil para o serviço.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Nesse caso, as políticas de Mary devem ser atualizadas para permitir que ela realize a ação `iam:PassRole`.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

Quero permitir que pessoas fora da minha AWS conta acessem meus AWS HealthLake recursos

Você pode criar um perfil que os usuários de outras contas ou pessoas fora da sua organização podem usar para acessar seus recursos. Você pode especificar quem é confiável para assumir o perfil. Para serviços que oferecem suporte a políticas baseadas em recursos ou listas de controle de acesso (ACLs), você pode usar essas políticas para conceder às pessoas acesso aos seus recursos.

Para saber mais, consulte:

- Para saber se é HealthLake compatível com esses recursos, consulteComo AWS HealthLake funciona com IAM.
- Para saber como fornecer acesso aos seus recursos em todos os Contas da AWS que você possui, consulte Fornecer acesso a um IAM usuário em outro Conta da AWS de sua propriedade no Guia do IAM usuário.

- Para saber como fornecer acesso aos seus recursos a terceiros Contas da AWS, consulte Fornecer Contas da AWS acesso a terceiros no Guia do IAM usuário.
- Para saber como conceder acesso por meio da federação de identidades, consulte Conceder acesso a usuários autenticados externamente (federação de identidades) no Guia do usuário do IAM.
- Para saber a diferença entre usar funções e políticas baseadas em recursos para acesso entre contas, consulte Acesso a recursos entre contas IAM no Guia do IAM usuário.

Registro em log de chamadas do AWS HealthLake API com o AWS CloudTrail

AWS HealthLake é integrado com AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, função ou AWS serviço em HealthLake. CloudTrail captura todas as API chamadas para HealthLake eventos. As chamadas capturadas incluem chamadas do HealthLake console e chamadas de código para as HealthLake API operações. Se você criar uma trilha, poderá habilitar a entrega contínua de CloudTrail eventos para um bucket do Amazon S3, incluindo eventos para. HealthLake Se você não configurar uma trilha, ainda poderá ver os eventos mais recentes no CloudTrail console no Histórico de eventos. Usando as informações coletadas por CloudTrail, você pode determinar a solicitação que foi feita HealthLake, o endereço IP do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita e detalhes adicionais.

Para saber mais sobre isso CloudTrail, consulte o Guia AWS CloudTrail do usuário.

AWS HealthLake Informações no CloudTrail

CloudTrail é ativado em sua AWS conta quando você cria a conta. Quando a atividade ocorre em HealthLake, essa atividade é registrada em um CloudTrail evento junto com outros eventos AWS de serviço no histórico de eventos. É possível visualizar, pesquisar e baixar eventos recentes em sua AWS conta. Para obter mais informações, consulte Visualização de eventos com histórico de CloudTrail eventos.

Para um registro contínuo dos eventos em sua AWS conta, incluindo eventos para HealthLake, crie uma trilha. Uma trilha permite CloudTrail entregar arquivos de log para um bucket do Amazon S3. Por padrão, ao criar uma trilha no console, a mesma é aplicada a todas as regiões da AWS. A trilha registra eventos de todas as regiões na AWS partição e entrega os arquivos de log ao bucket do Amazon S3 que você especificar. Além disso, você pode configurar outros AWS serviços para

analisar e agir com base nos dados de eventos coletados nos CloudTrail registros. Para obter mais informações, consulte as informações a seguir.

- Visão Geral para Criar uma Trilha
- CloudTrail Serviços e integrações compatíveis
- Configurando as SNS notificações da Amazon para CloudTrail
- Recebendo arquivos de CloudTrail log de várias regiões e recebendo arquivos de CloudTrail log de várias contas

Todas HealthLake as ações são registradas CloudTrail e documentadas na HealthLake APIReferência e neste Guia do Desenvolvedor para ações realizadas usando o. FHIR REST API Por exemplo, chamadas para as ações a seguir geram entradas nos arquivos de CloudTrail log:

- `DescribeFHIRImportJob`
- `DescribeFHIRExportJob`
- `StartFHIRImportJob`
- `ListFHIRImportJobs`
- `StartFHIRExportJob`
- `ListFHIRExportJobs`
- `CreateFHIRDatastore`
- `ListFHIRDatastores`
- `DeleteFHIRDatastore`
- `DescribeFHIRDatastore`
- `UpdateResource`
- `CreateResource`
- `DeleteResource`
- `ReadResource`
- `GetCapabilities`
- `SearchWithGet`
- `SearchWithPost`

Cada entrada de log ou evento contém informações sobre quem gerou a solicitação. As informações de identidade ajudam a determinar o seguinte:

- Se a solicitação foi feita com credenciais de usuário root ou AWS Identity and Access Management (IAM).
- Se a solicitação foi feita com credenciais de segurança temporárias de um perfil ou de um usuário federado.
- Se a solicitação foi feita por outro AWS serviço.

Para obter mais informações, consulte o CloudTrail userIdentityElemento.

Entendendo as entradas do arquivo de AWS HealthLake log

Uma trilha é uma configuração que permite a entrega de eventos como arquivos de log para um bucket do Amazon S3 que você especificar. CloudTrail os arquivos de log contêm uma ou mais entradas de log. Um evento representa uma única solicitação de qualquer fonte e inclui informações sobre a ação solicitada, a data e a hora da ação, os parâmetros da solicitação e assim por diante. CloudTrail os arquivos de log não são um rastreamento de pilha ordenado das API chamadas públicas, portanto, eles não aparecem em nenhuma ordem específica.

O exemplo a seguir mostra uma entrada de CloudTrail registro que demonstra a `CreateFHIRDatastore` ação.


```
{
    "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"
}
```


Validação de conformidade para AWS HealthLake

Auditores terceirizados avaliam a segurança e a conformidade AWS HealthLake como parte de vários programas de AWS conformidade. Pois HealthLake isso incluiHIPAA.

Para saber se um AWS service (Serviço da AWS) está dentro do escopo de programas de conformidade específicos, consulte Serviços da AWS Escopo por Programa de Conformidade Serviços da AWS e escolha o programa de conformidade em que você está interessado. Para obter informações gerais, consulte Programas de AWS conformidade Programas AWS de .

Você pode baixar relatórios de auditoria de terceiros usando AWS Artifact. Para obter mais informações, consulte Baixar relatórios em AWS Artifact .

Sua responsabilidade de conformidade ao usar Serviços da AWS é determinada pela confidencialidade de seus dados, pelos objetivos de conformidade de sua empresa e pelas leis e regulamentações aplicáveis. AWS fornece os seguintes recursos para ajudar na conformidade:

- Governança e conformidade de segurança: esses guias de implementação de solução abordam considerações sobre a arquitetura e fornecem etapas para implantar recursos de segurança e conformidade.
- Arquitetura para HIPAA segurança e conformidade na Amazon Web Services — Este whitepaper descreve como as empresas podem usar AWS para criar HIPAA aplicativos qualificados.

> **Note**
>
> Nem todos Serviços da AWS são HIPAA elegíveis. Para obter mais informações, consulte a Referência de serviços HIPAA elegíveis.

- AWS Recursos de https://aws.amazon.com/compliance/resources/ de conformidade — Essa coleção de pastas de trabalho e guias pode ser aplicada ao seu setor e local.
- AWS Guias de conformidade do cliente — Entenda o modelo de responsabilidade compartilhada sob a ótica da conformidade. Os guias resumem as melhores práticas de proteção Serviços da AWS e mapeiam as diretrizes para controles de segurança em várias estruturas (incluindo o Instituto Nacional de Padrões e Tecnologia (NIST), o Conselho de Padrões de Segurança do Setor de Cartões de Pagamento (PCI) e a Organização Internacional de Padronização ()). ISO
- Avaliação de recursos com regras no Guia do AWS Config desenvolvedor — O AWS Config serviço avalia o quão bem suas configurações de recursos estão em conformidade com as práticas internas, as diretrizes e os regulamentos do setor.
- AWS Security Hub— Isso AWS service (Serviço da AWS) fornece uma visão abrangente do seu estado de segurança interno AWS. O Security Hub usa controles de segurança para avaliar os recursos da AWS e verificar a conformidade com os padrões e as práticas recomendadas do setor

de segurança. Para obter uma lista dos serviços e controles aceitos, consulte a Referência de controles do Security Hub.

- Amazon GuardDuty — Isso AWS service (Serviço da AWS) detecta possíveis ameaças às suas cargas de trabalho Contas da AWS, contêineres e dados monitorando seu ambiente em busca de atividades suspeitas e maliciosas. GuardDuty pode ajudá-lo a atender a vários requisitos de conformidade, por exemplo PCIDSS, atendendo aos requisitos de detecção de intrusões exigidos por determinadas estruturas de conformidade.
- AWS Audit Manager— Isso AWS service (Serviço da AWS) ajuda você a auditar continuamente seu AWS uso para simplificar a forma como você gerencia o risco e a conformidade com as regulamentações e os padrões do setor.

Resiliência em AWS HealthLake

A infraestrutura AWS global é construída em torno de AWS regiões e zonas de disponibilidade. AWS As regiões fornecem várias zonas de disponibilidade fisicamente separadas e isoladas, conectadas a redes de baixa latência, alta taxa de transferência e alta redundância. Com as zonas de disponibilidade, é possível projetar e operar aplicações e bancos de dados que automaticamente executam o failover entre as zonas sem interrupção. As zonas de disponibilidade são altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de data center tradicionais.

Para obter mais informações sobre AWS regiões e zonas de disponibilidade, consulte Infraestrutura AWS global.

Além da infraestrutura AWS global, HealthLake oferece vários recursos para ajudar a suportar suas necessidades de resiliência e backup de dados.

Segurança da Infraestrutura no AWS HealthLake

Como serviço gerenciado, AWS HealthLake é protegido pelos procedimentos AWS globais de segurança de rede descritos no whitepaper Amazon Web Services: Visão geral dos processos de segurança.

Você usa API chamadas AWS publicadas para acessar HealthLake pela rede. Os clientes devem oferecer suporte ao Transport Layer Security (TLS) 1.0 ou posterior. Recomendamos TLS 1.2 ou posterior. Os clientes também devem oferecer suporte a pacotes de criptografia com sigilo direto perfeito (), como Ephemeral Diffie-Hellman (PFS) ou Elliptic Curve Ephemeral Diffie-Hellman (). DHE

ECDHE A maioria dos sistemas modernos, como Java 7 e versões posteriores, comporta esses modos.

Além disso, as solicitações devem ser assinadas usando um ID da chave de acesso e uma chave de acesso secreta associada a uma entidade principal do IAM. Ou é possível usar o AWS Security Token Service (AWS STS) para gerar credenciais de segurança temporárias para assinar solicitações.

Melhores práticas de segurança no AWS HealthLake

AWS HealthLake fornece vários recursos de segurança a serem considerados ao desenvolver e implementar suas próprias políticas de segurança. As práticas recomendadas a seguir são diretrizes gerais e não representam uma solução completa de segurança. Como essas práticas recomendadas podem não ser adequadas ou suficientes para o seu ambiente, trate-as como considerações úteis em vez de prescrições.

- Implemente o acesso de privilégio mínimo.
- Sempre que possível, use Customer-Managed-Keys (CMKs) para criptografar seus dados. Para saber mais sobre issoCMKs, consulte Amazon Key Management Service.
- Use Pesquisar comPOST, não Pesquisar com, GET ao consultar PHI ou PII em seu armazenamento de dados.
- Limite o acesso a funções de auditoria importantes e confidenciais.
- Ao criar recursos por meio da atualização ou importação em massaAPIs, não use PHI ouPII, incluindo os nomes dos armazenamentos de dados e trabalhos, em nenhum campo visível ou na FHIR ID lógica (LID).
- Ao enviar solicitações de criação, leitura, atualização, exclusão ou pesquisa, não use PHI no HTTP cabeçalho.
- Permita AWS CloudTrail auditar o AWS HealthLake uso e garantir que não haja nenhuma atividade inesperada.
- Analise as melhores práticas para usar buckets do Amazon S3 com segurança. Para saber mais, consulte as melhores práticas de segurança no guia do usuário do Amazon S3.

AWS HealthLake endpoints e cotas

As seções a seguir contêm informações sobre AWS HealthLake cotas e endpoints. É possível solicitar aumentos de cota para cotas ajustáveis do IAM usando o console do Service Quotas. Para obter mais informações, consulte Solicitando um Aumento de Cota no Guia do Usuário do Service Quotas.

Service endpoints

A tabela mostra os endpoints HealthLake de serviço disponíveis em uma determinada região.

Nome da região	Região	Endpoint	Protocolo	
Leste dos EUA (Ohio)	us-east-2	healthlake.us-east-2.amazonaws.com healthlake-fips.us-east-2.amazonaws.com	HTTPS HTTPS	
Leste dos EUA (Norte da Virgínia)	us-east-1	healthlake.us-east-1.amazonaws.com healthlake-fips.us-east-1.amazonaws.com	HTTPS HTTPS	
Oeste dos EUA (Oregon)	us-west-2	healthlake.us-west-2.amazonaws.com healthlake-fips.us-west-2.amazonaws.com	HTTPS HTTPS	
Ásia-Pacífico (Mumbai)	ap-south-1	healthlake.ap-south-1.amazonaws.com	HTTPS	
Ásia-Pacífico (Sydney)	ap-southeast-2	healthlake.ap-southeast-2.amazonaws.com	HTTPS	
Europa (Londres)	eu-west-2	healthlake.eu-west-2.amazonaws.com	HTTPS	

Cotas de serviço para HealthLake

A seguir estão as cotas padrão para HealthLake.

Nome	Padrão	Ajustá	Descrição
Número de caracteres em uma observação médica	Cada região compatível: 10.000	Não	O número máximo de caracteres em um atestado médico individual dentro do tipo de DocumentReference recurso (POST/PUT solicitações).
Número de tartFHIRImport trabalhos S Job simultâneos	Cada região compatível: 1	Não	O máximo de trabalhos S tartFHIRImport Job simultâneos.
Número de concurrentStart FHIRExpor tJob empregos	Cada região compatível: 1	Não	O máximo de trabalhos S tartFHIRExport Job simultâneos.
Número de armazenamentos de dados por conta	Cada região compatível: 10	Sim	O número máximo padrão de armazenam entos de dados ativos por conta.
Número de arquivos em um S tartFHIRI mport Job	Cada região compatível: 10.000	Não	O número máximo de arquivos em um S tartFHIRImport Job.
Número de recursos por pacote	Cada região compatível: 160	Não	O número máximo de recursos permitidos em uma solicitação de pacote.
Taxa de solicitações de pacotes por conta	Cada região compatível: 20	Sim	O número máximo de solicitações de POST

Nome	Padrão	Ajustá	Descrição
			pacote que você pode fazer por segundo por conta.
Taxa de solicitações de pacotes por armazenamento de dados	Cada região compatível: 10	Sim	O número máximo de solicitações de POST pacote que você pode fazer por segundo por armazenamento de dados. Os armazenamentos de dados criados antes de 21/08/2023 serão limitados a 1 solicitação por segundo.
Taxa de solicitações de ancelFHIR Export trabalho C usadas DELETE por conta	Cada região compatível: 1	Não	O número máximo de solicitações de CancelFHIRExport Job usando DELETE que você pode fazer por minuto por conta.
Taxa de reateFHIRDatastore solicitações C por conta	Cada região compatível: 1	Não	O número máximo de reateFHIRDatastore solicitações C que você pode fazer por minuto por conta.
Taxa de solicitações DELETE por conta	Cada região compatível: 2.000	Sim	O número máximo de DELETE solicitações que você pode fazer por segundo por conta.

Nome	Padrão	Ajustá	Descrição
Taxa de DELETE solicitações por armazenamento de dados	Cada região compatível: 1.000	Sim	O número máximo de DELETE solicitações que você pode fazer por segundo por armazenamento de dados. Os armazenamentos de dados criados antes de 21/08/2023 serão limitados a 100 solicitações por segundo.
Taxa de eleteFHIRDatastore solicitações de D por conta	Cada região compatível: 1	Não	O número máximo de eleteFHIRDatastore solicitações D que você pode fazer por minuto por conta.
Taxa de escribeFHIRDatastore solicitações de D por conta	Cada região com suporte: 10	Não	O número máximo de escribeFHIRDatastore solicitações D que você pode fazer por segundo por conta.
Taxa de solicitações de escribeFHIRExport emprego D por conta	Cada região com suporte: 10	Não	O número máximo de solicitações de D escribeFHIRExport Job que você pode fazer por segundo por conta.
Taxa de D: solicitações de escribeFHIRExport emprego usadas GET por conta	Cada região com suporte: 10	Não	O número máximo de solicitações de D escribeFHIRExport Job usando GET que você pode fazer por segundo por conta.

Nome	Padrão	Ajustá	Descrição
Taxa de solicitações de escribeFHIRImport emprego D por conta	Cada região com suporte: 10	Não	O número máximo de solicitações de DescribeFHIRImport Job que você pode fazer por segundo por conta.
Taxa de solicitações de descoberta por conta	Cada região com suporte: 10	Não	Número máximo de solicitações Discovery que é possível fazer por minuto por conta.
Taxa de solicitações GET por conta	Cada região compatível: 6.000	Sim	O número máximo de GET solicitações que você pode fazer por segundo por conta.
Taxa de GET solicitações por armazenamento de dados	Cada região compatível: 3.000	Sim	O número máximo de GET solicitações que você pode fazer por segundo por armazenamento de dados. Os armazenamentos de dados criados antes de 21/08/2023 serão limitados a 100 solicitações por segundo.
Taxa de GetCapabilities solicitações por conta	Cada região com suporte: 10	Não	O número máximo de GetCapabilities solicitações que você pode fazer por segundo por conta.

Nome	Padrão	Ajustá	Descrição
Taxa de istFHIRDatastores solicitações L por conta	Cada região com suporte: 10	Não	O número máximo de istFHIRDatastores solicitações L que você pode fazer por segundo por conta.
Taxa de solicitações de L istFHIRExport Jobs por conta	Cada região com suporte: 10	Não	O número máximo de solicitações de L istFHIRExport Jobs que você pode fazer por segundo por conta.
Taxa de solicitações de L istFHIRImport Jobs por conta	Cada região com suporte: 10	Não	O número máximo de solicitações de L istFHIRImport Jobs que você pode fazer por segundo por conta.
Taxa de ListTagsforResource solicitações por conta	Cada região com suporte: 10	Não	O número máximo de ListTagsforResource solicitações que você pode fazer por segundo por conta.
Taxa de solicitações POST por conta	Cada região compatível: 2.000	Sim	O número máximo de POST solicitações que você pode fazer por segundo por conta.

Nome	Padrão	Ajustá	Descrição
Taxa de POST solicitações por armazenamento de dados	Cada região compatível: 1.000	Sim	O número máximo de POST solicitações que você pode fazer por segundo por armazenamento de dados. Os armazenamentos de dados criados antes de 21/08/2023 serão limitados a 100 solicitações por segundo.
Taxa de solicitações PUT por conta	Cada região compatível: 2.000	Sim	O número máximo de PUT solicitações que você pode fazer por segundo por conta.
Taxa de PUT solicitações por armazenamento de dados	Cada região compatível: 1.000	Sim	O número máximo de PUT solicitações que você pode fazer por segundo por armazenamento de dados. Os armazenamentos de dados criados antes de 21/08/2023 serão limitados a 100 solicitações por segundo.
Taxa de solicitações de tartFHIRExport emprego S por conta	Cada região compatível: 1	Não	O número máximo de solicitações de StartFHIRExport Job que você pode fazer por minuto por conta.

Nome	Padrão	Ajustá	Descrição
Taxa de S: Solicitações de tartFHIRE xport emprego POST usadas por conta	Cada região compatível: 1	Não	O número máximo de solicitações de S tartFHIRExport Job usando POST que você pode fazer por minuto por conta.
Taxa de solicitações de tartFHIRImport emprego S por conta	Cada região compatível: 1	Não	O número máximo de solicitações de S tartFHIRImport Job que você pode fazer por minuto por conta.
Taxa de TagResource solicitações por conta	Cada região com suporte: 10	Não	O número máximo de TagResource solicitações que você pode fazer por segundo.
Taxa de UntagResource solicitações por conta	Cada região com suporte: 10	Não	O número máximo de UntagResource solicitaç ões que você pode fazer por segundo por conta.
Taxa de solicitações de pesquisa usadas GET por conta	Cada região compatível: 200	Sim	O número máximo de solicitações de pesquisa usando GET isso que você pode fazer por segundo por conta.
Taxa de solicitações de pesquisa usadas GET por armazenamento de dados	Cada região compatível: 100	Sim	O número máximo de solicitações de pesquisa usando GET isso que você pode fazer por segundo por armazenam ento de dados.

Nome	Padrão	Ajustá	Descrição
Taxa de solicitações de pesquisa usadas POST por conta	Cada região compatível: 200	Sim	O número máximo de solicitações de pesquisa usando POST isso que você pode fazer por segundo.
Taxa de solicitações de pesquisa usadas POST por armazenamento de dados	Cada região compatível: 100	Sim	O número máximo de solicitações de pesquisa usando POST isso que você pode fazer por segundo por armazenam ento de dados.
Tamanho do arquivo importado individual	Cada região compatível: 5 gigabytes	Não	O tamanho máximo (em GB) de um arquivo individual incluído em um S tartFHIRImport Job.
Como visualizar resultados de trabalho de importação	Cada região compatível: 500 gigabytes	Não	O tamanho máximo (em GB) de todos os arquivos incluídos na tarefa de importação.

Solução de problemas

A documentação a seguir pode ajudá-lo a solucionar problemas que você possa ter com o uso AWS HealthLake.

Tópicos

Por que não consigo criar um armazenamento HealthLake de dados?

Em 14 de novembro de 2022, HealthLake atualizou as IAM permissões necessárias para criar um novo armazenamento de dados. Se você não atualizou as políticas anexadas ao usuário ou à função que acessa, HealthLake você receberá o seguinte erro.

```
AccessDeniedException: Insufficient Lake Formation permission(s): Required Database on
 Catalog
```

Para ver os requisitos IAM de política atualizados para criar um armazenamento de dados, consulte a política AWS gerenciada: AmazonHealthLakeFullAccess. Para obter step-by-step instruções sobre como adicionar essas políticas ao seu IAM usuário ou função, consulteConfigurando permissões para começar a usar AWS HealthLake.

Para criar um armazenamento de dados, você também precisa usar uma chave simétrica de propriedade do cliente ou da Amazon. KMS Verifique se você tem as permissões corretas em sua IAM política. Para saber mais sobre isso AWS KMS, consulte AWS Key Management Serviceo Guia do AWS Key Management Service desenvolvedor.

Número excedido de armazenamentos de dados permitidos por conta

HealthLake tem uma cota de 10 armazenamentos de dados por conta. Para saber como solicitar um aumento de cota, visite o AWS Support Center.

Como faço para criar uma autorização para o FHIR RESTfulAPIs?

Os usuários devem usar um processo de assinatura Signature versão 4 para adicionar autenticação às HealthLake API solicitações enviadas por meio de um HTTP cliente. Para saber mais, consulte Processo de assinatura do Signature versão 4.

Para criar a autorização sigv4 usando o AWS SDK for Python, crie um script semelhante ao exemplo a seguir.

```
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())
```

Informações adicionais sobre o uso da autorização sigv4 para AWS SDK Python podem ser encontradas no tópico Credenciais do Boto3.

Meus dados não estão no formato FHIR R4. Ainda posso usar HealthLake?

Somente dados formatados em FHIR R4 podem ser importados para um armazenamento de HealthLake dados. Para ver uma lista de parceiros que oferecem produtos para ajudar os usuários a transformar seus dados, consulte AWS HealthLake Parceiros.

Por que estou recebendo AccessDenied erros ao usar o FHIR RESTful APIs para um armazenamento de dados criptografado com uma KMS chave gerenciada pelo cliente?

As permissões para a chave e IAM as políticas gerenciadas pelo cliente são necessárias para que um usuário ou função acesse um armazenamento de dados. Um usuário deve ter as IAM permissões necessárias para usar uma chave gerenciada pelo cliente. Se um usuário revogou ou retirou uma concessão que dava HealthLake permissão para usar a KMS chave gerenciada pelo cliente, HealthLake retornará um AccessDenied erro.

HealthLake deve ter a permissão para acessar os dados do cliente, criptografar novos FHIR recursos importados para um armazenamento de dados e descriptografar os FHIR recursos quando solicitados.

Para saber mais, consulte Solução de problemas de acesso por chave.

Por que minha importação falhou?

Um trabalho de importação bem-sucedido gerará uma pasta com inputFileName arquivos.ndjson de saída, no entanto, registros individuais podem falhar na importação. Quando isso acontecer, uma segunda FAILURE pasta será gerada com um manifesto de registros que não foram importados. O local de saída do trabalho para acessar o arquivo de manifesto é JobProperties. JobOutputDataConfig.Configuração do S3.Uri do S3.

Esse arquivo de manifesto contém detalhes sobre a saída do trabalho, como a localização de todas as respostas bem-sucedidas (successOutput. successOutputS3Uri), a localização de todas as respostas com falha (. failureOutput failureOutputS3Uri) e métricas de trabalho adicionais. O conteúdo do arquivo de manifesto pode ser analisado programaticamente. O exemplo de arquivo de manifesto a seguir lista os buckets de entrada e saída do Amazon S3 e também informações sobre o número de recursos escaneados e quantos foram importados com sucesso.

```
            {
            "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 analisar por que um trabalho de importação falhou, use o D escribeFHIRImport Job API para analisar JobProperties o. O seguinte é recomendado:

- Se o status for FAILED e uma mensagem estiver presente, as falhas estão relacionadas a parâmetros do trabalho, como tamanho dos dados de entrada ou número de arquivos de entrada, que estão além das HealthLake cotas.
- Se o status do trabalho de importação for COMPLETED _ WITH _ERRORS, verifique o arquivo de manifesto, Manifest.json, para obter informações sobre quais arquivos não foram importados com êxito.
- Se o status do trabalho de importação for FAILED e uma mensagem não estiver presente, acesse o local de saída do trabalho para acessar o arquivo de manifesto, Manifest.json.

Para cada arquivo de entrada, há um arquivo de saída de falha com o nome do arquivo de entrada para qualquer recurso que falhe na importação. As respostas contêm o número da linha (lineId) correspondente à localização dos dados de entrada, objeto de FHIR resposta (UpdateResourceResponse) e código de status (statusCode) da resposta.

Um exemplo de arquivo de saída teria a seguinte aparência:

{"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}
```


O exemplo mostra que houve falhas na linha 3, 4, 7, 9, 15 das linhas de entrada correspondentes do arquivo de entrada. Para cada uma dessas linhas, as explicações são as seguintes:

- Na linha 3, a resposta explica que o resourceType fornecido na linha 3 do arquivo de entrada não é válido.
- Na linha 5, a resposta explica que há um erro de FHIR validação na linha 5 do arquivo de entrada.
- Na linha 7, a resposta explica que há um problema de validação com o resourceId fornecimento como entrada.
- Na linha 9, a resposta explica que o arquivo de entrada deve conter um ID de recurso válido.
- Na linha 15, a resposta do arquivo de entrada é que o arquivo não está em um JSON formato válido.

Como faço para encontrar DocumentReference recursos que não puderam ser processados?

Se um DocumentReference recurso não for válido, HealthLake fornecerá uma extensão indicando um erro de validação em vez da NLP saída médica integrada. Para encontrar DocumentReference

recursos que causaram um erro de validação durante o NLP processamento, os clientes podem usar a função HealthLake de pesquisa com chave de pesquisa cm-decoration-statuse valor de pesquisa VALIDATION_ ERROR. Essa pesquisa listará todos os DocumentReference recursos que levaram a erros de validação, junto com uma mensagem de erro descrevendo a natureza do erro. A estrutura do campo de extensão nesses DocumentReference recursos com erros de validação será semelhante ao exemplo a seguir.

```
"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/"
          }
    ]
```

Um VALIDATION_ também ERROR pode ocorrer se a NLP decoração criar mais de 10.000 objetos aninhados. Quando isso acontece, o documento deve ser dividido em documentos menores antes do processamento.

Migração de um armazenamento de dados existente para usar o Amazon Athena

os armazenamentos de dados criados antes de 14 de novembro de 2022 são funcionais, mas não podem ser consultados no Athena usando o. SQL Para consultar um armazenamento de dados preexistente com o Athena, você deve primeiro migrá-lo para um novo armazenamento de dados.

Para migrar dados para um novo armazenamento de dados

1. Crie um novo armazenamento de dados.

2. Exporte os dados do pré-existente para um bucket do Amazon S3.
3. Importe os dados para o novo armazenamento de dados do bucket do Amazon S3.

A exportação de dados para um bucket do Amazon S3 incorre em uma taxa extra. A taxa extra depende do tamanho dos dados que você exporta.

Conectando resultados de pesquisa no Athena a outros serviços AWS

Você pode ter problemas ao compartilhar seus resultados de pesquisa do Athena com outros AWS serviços.

O problema pode ocorrer quando você usa `json_extract[1]` como parte de uma consulta de SQL pesquisa.

Para corrigir esse problema, você deve atualizar para `CATVAR` o.

Você pode encontrar esse problema ao tentar criar resultados salvos, uma tabela (estática) ou uma visualização (dinâmica).

O console do Athena não está funcionando depois de importar dados para um novo armazenamento de dados

Depois de importar dados para um novo armazenamento de dados, os dados podem não estar disponíveis para uso imediato. Isso é para dar tempo para que os dados sejam ingeridos nas tabelas de iceberg. Tente novamente mais tarde.

Por que recebo um erro de permissões do Lake Formation: lakeformation: PutDataLakeSettings ao adicionar um novo administrador do data lake?

Se seu IAM usuário ou função contiver a política `AWSLakeFormationDataAdmin` AWS gerenciada, você não poderá adicionar novos administradores de data lake. Você receberá um erro contendo o seguinte:

```
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
```

A política AWS gerenciada `AdministratorAccess` é necessária para adicionar um IAM usuário ou uma função como administrador de data lake do AWS Lake Formation. Se seu IAM usuário ou função também contiver`AWSLakeFormationDataAdmin`, a ação falhará. A política `AWSLakeFormationDataAdmin` AWS gerenciada contém uma negação explícita da API operação AWS Lake Formation,`PutDataLakeSetting`.

Até mesmo administradores com acesso total ao AWS uso da política `AdministratorAccess` AWS gerenciada podem ser limitados pela `AWSLakeFormationDataAdmin` política.

Como faço para ativar o recurso integrado HealthLake de processamento de linguagem natural do?

Em 20 de fevereiro de 2023, o comportamento padrão dos armazenamentos de HealthLake dados mudou.

Armazenamentos de dados atuais: todos os armazenamentos de HealthLake dados atuais deixarão de usar processamento de linguagem natural (NLP) em recursos codificados em base64`DocumentReference`. Isso significa que novos `DocumentReference` recursos não serão analisados usandoNLP, e nenhum novo recurso será gerado com base no texto do tipo de `DocumentReference` recurso. Para `DocumentReference` os recursos existentes, os dados e os recursos gerados por meio deles NLP permanecem, mas não serão atualizados após 20 de fevereiro de 2023.

Novos armazenamentos de HealthLake dados: os armazenamentos de dados criados após 20 de fevereiro de 2023 não executarão processamento de linguagem natural (NLP) em recursos codificados em base64`DocumentReference`.

Para ativar esse recurso, você deve criar um caso usando AWS Support Center Console. Para criar seu caso, faça login no seu Conta da AWS e escolha Criar caso. Para saber mais sobre a criação de um caso e o gerenciamento de casos, consulte Criação de casos de suporte e gerenciamento de casos no Guia Suporte do usuário.

O status do meu armazenamento de dados não está mudando de Criação

Se você tentar criar um novo armazenamento de HealthLake dados e o status do armazenamento de dados não mudar de Criando, você precisará atualizar o Athena para usar o. AWS Glue Data Catalog

Para saber mais, consulte Atualização para o catálogo de dados AWS Glue step-by-step no Guia do usuário do Amazon Athena.

Depois de atualizar com sucesso o AWS Glue Data Catalog, agora você pode criar um armazenamento de dados.

Para remover o armazenamento de dados antigo, comece criando um caso usando AWS Support Center Console. Para criar seu caso, faça login no seu Conta da AWS e escolha Criar caso. Para saber mais, consulte Criação de casos de suporte e gerenciamento de casos no Guia Suporte do usuário.

O status de criação do meu armazenamento de SDK dados retorna uma exceção ou um status desconhecido

Atualize sua versão SDK para a versão mais recente se suas API chamadas de armazenamento de dados de lista ou descrição do armazenamento de dados retornarem uma exceção ou um status de armazenamento de dados desconhecido.

Minha FHIR POST API operação com um documento de 10 MB para HealthLake obter um erro 413Request Entity Too Large.

AWS HealthLake tem um API limite síncrono de criação e atualização de 5 MB para evitar maiores latências e tempos limite.

Você pode ingerir documentos grandes, de até 164 MB, usando o binário ResourceType usando a importação em massa. API

Histórico do documento para o guia do AWS HealthLake desenvolvedor

A tabela a seguir descreve as alterações na documentação das AWS HealthLake versões.

- APIversão: mais recente
- Última atualização da documentação: 25/10/2024

Alteração	Descrição	Data
HealthLake agora suportes FHIR `history` e `vread` interações	HealthLake agora suporta a FHIR `history` interação para recuperar o histórico de um recurso específico e a `vread` interação para realizar uma leitura específica da versão de um recurso.	25 de outubro de 2024
HealthLake agora suporta novos parâmetros FHIR de pesquisa, extensão e tipo de recurso.	HealthLake agora suporta novos parâmetros FHIR de pesquisa, extensão e tipo de recurso.	9 de dezembro de 2023
HealthLake agora suporta a FHIR estrutura SMART on	HealthLake agora suporta a criação SMART em armazenamentos HealthLake de dados FHIR habilitados.	31 de maio de 2023
HealthLake agora suporta validação de perfil	HealthLake agora oferece suporte à validação FHIR do perfil.	31 de maio de 2023
HealthLake agora suporta `export`	HealthLake agora suporta a exportação de arquivos usando a FHIR REST API operação`export`.	31 de maio de 2023

Região Ásia-Pacífico (Mumbai)	AWS HealthLake agora está disponível na região Ásia-Pacífico (Mumbai).	4 de abril de 2023
Processamento integrado de linguagem natural desativado	HealthLake desativou o processamento integrado de linguagem natural (NLP) em todos os armazenamentos de dados a partir de 20 de fevereiro de 2023.	20 de fevereiro de 2023
HealthLake integra-se ao Amazon Athena	Agora você pode usar o Athena para consultar armazenamentos de dados criados após 14 de novembro de 2022.	14 de novembro de 2022
O tamanho total do trabalho de importação aumentou	O tamanho total máximo de todos os arquivos em uma solicitação de S tartFHIRI mport Job agora é de 500 GB.	3 de outubro de 2022
Suporte para pacotes	HealthLake agora é compatível com o tipo de recurso Bundle para ingerir vários recursos.	5 de agosto de 2022
Cotas atualizadas para CRUD operações em HealthLake	HealthLake agora oferece suporte a limites mais altos para CRUD solicitações.	14 de julho de 2022
Incluir suporte	HealthLake agora oferece suporte `_include` em consultas de armazenamento de dados.	14 de julho de 2022
AWS HealthLake agora está disponível ao público em geral	HealthLake agora está disponível ao público em geral.	30 de julho de 2020

AWS Glossário

Para obter a AWS terminologia mais recente, consulte o AWS glossário na Glossário da AWS Referência.

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