Guida per gli sviluppatori

AWS HealthLake

AWS HealthLake: Guida per gli sviluppatori

Table of Contents

Che cos'è AWS HealthLake?

AWS HealthLake è un servizio HIPAA idoneo per l'inserimento, l'archiviazione e l'analisi di dati clinici che utilizza la specifica Healthcare Interoperability FHIR (R4).

> **Note**
>
> Dopo il 20 febbraio 2023, gli archivi HealthLake dati non utilizzano l'elaborazione integrata del linguaggio naturale () per impostazione predefinita. NLP Se sei interessato ad attivare questa funzionalità sul tuo data store, consulta il Come posso attivare la funzionalità integrata HealthLake di elaborazione del linguaggio naturale? capitolo Risoluzione dei problemi.

I dati sanitari sono spesso incompleti e incoerenti. Inoltre, sono spesso non strutturati, con informazioni contenute in note cliniche, rapporti di laboratorio, richieste di risarcimento, immagini mediche, conversazioni registrate e dati di serie temporali (ad esempio tracce cardiache ECG o cerebraliEEG).

Gli operatori sanitari possono utilizzarli HealthLake per archiviare, trasformare, interrogare e analizzare i dati nel AWS cloud. Utilizzando le funzionalità HealthLake integrate di elaborazione del linguaggio naturale (NLP) per uso medico, è possibile analizzare testi clinici non strutturati provenienti da diverse fonti. HealthLake trasforma i dati non strutturati utilizzando modelli di elaborazione del linguaggio naturale e fornisce potenti funzionalità di interrogazione e ricerca. È possibile utilizzarli HealthLake per organizzare, indicizzare e strutturare le informazioni sui pazienti in modo sicuro, conforme e verificabile.

HealthLake è inoltre integrato con Amazon Athena e AWS Lake Formation. Puoi utilizzare questa integrazione per interrogare il tuo archivio dati utilizzandoSQL.

Vantaggi di AWS HealthLake

Con AWS HealthLake, puoi:

- Inserimento rapido e semplice di dati sanitari: puoi importare in blocco file Fast Healthcare Interoperability Resources (FHIR) locali, tra cui note cliniche, report di laboratorio, richieste di risarcimento e altro, in un bucket Amazon Simple Storage Service (Amazon S3). Puoi quindi utilizzare i dati in applicazioni o flussi di lavoro a valle.

- Usa le FHIR REST API operazioni: HealthLake supporta l'utilizzo delle FHIR REST API operazioni per eseguire CRUD (Create/Read/Update/Delete) operazioni sul tuo data store. FHIRè supportata anche la ricerca.
- Archivia i tuoi dati nel AWS cloud in modo sicuro, HIPAA idoneo e verificabile: puoi archiviare i dati nel FHIR formato desiderato, in modo che possano essere facilmente interrogati. HealthLake crea una visualizzazione cronologica completa della storia medica di ogni paziente e la struttura nel formato standard R4. FHIR
- Integrazione con Athena: HealthLake l'integrazione con Athena consente di creare potenti query SQL basate su cui creare e salvare criteri di filtro complessi. Quindi, puoi utilizzare questi dati in applicazioni downstream come l' SageMaker intelligenza artificiale per addestrare un modello di apprendimento automatico o Amazon QuickSight per creare dashboard e visualizzazioni dei dati.
- Trasforma i dati non strutturati utilizzando modelli di machine learning (ML) specializzati: HealthLake fornisce l'elaborazione medica integrata del linguaggio naturale (NLP) utilizzando Amazon Comprehend Medical. I dati di testo medico non elaborati vengono trasformati utilizzando modelli ML specializzati. Questi modelli sono stati addestrati per comprendere ed estrarre informazioni significative da dati sanitari non strutturati. Con Integrated MedicalNLP, è possibile estrarre automaticamente dal testo medico le entità (ad esempio, procedure mediche e farmaci), le relazioni tra entità (ad esempio, un farmaco e il relativo dosaggio) e le caratteristiche delle entità (ad esempio, il risultato positivo o negativo del test o l'ora della procedura). HealthLake quindi crea nuove risorse in base ai tratti, al segno, al sintomo e alla condizione. Questi vengono aggiunti come nuovi tipi di condizione, osservazione e MedicationStatement risorsa.

HealthLake casi d'uso

È possibile utilizzarlo HealthLake per le seguenti applicazioni sanitarie:

- Gestione della salute della popolazione: HealthLake aiuta le organizzazioni sanitarie ad analizzare le tendenze, i risultati e i costi sulla salute della popolazione. Questo aiuta le organizzazioni a identificare l'intervento più appropriato per una popolazione di pazienti e a scegliere migliori opzioni di gestione dell'assistenza.
- Miglioramento della qualità dell' HealthLake assistenza: ospedali, compagnie di assicurazione sanitaria e organizzazioni del settore delle scienze della vita colmano le lacune nell'assistenza, migliorano la qualità dell'assistenza e riducono i costi compilando una visione completa della storia medica del paziente.

- Ottimizzazione dell'efficienza ospedaliera: HealthLake offre agli ospedali strumenti chiave di analisi e apprendimento automatico per migliorare l'efficienza e ridurre gli sprechi ospedalieri.

Accedendo HealthLake

È possibile accedere HealthLake tramite AWS Management Console, AWS Command Line Interface (AWS CLI) o AWS SDKs.

1. AWS Management Console — Fornisce un'interfaccia web che è possibile utilizzare per accedere HealthLake.
2. AWS Command Line Interface (AWS CLI) — Fornisce comandi per un'ampia gamma di AWS servizi HealthLake, inclusi ed è supportato su Windows, macOS e Linux. Per ulteriori informazioni sull'installazione di AWS CLI, vedere AWS Command Line Interface.
3. AWS SDKs— AWS fornisce SDKs (kit di sviluppo software) costituiti da librerie e codice di esempio per vari linguaggi e piattaforme di programmazione (Java, Python, Ruby,. NET, iOS, Android e così via). SDKsForniscono un modo conveniente per creare un accesso programmatico a HealthLake e AWS. Per ulteriori informazioni, vedere AWSSDKfor Python.

HIPAAidoneità e sicurezza dei dati

Questo è un servizio HIPAA idoneo. Per ulteriori informazioni sull' AWS U.S. Health Insurance Portability and Accountability Act del 1996 (HIPAA) e sull'utilizzo AWS dei servizi per elaborare, archiviare e trasmettere informazioni sanitarie protette (PHI), vedere HIPAA Panoramica.

Le connessioni che HealthLake contengono informazioni di identificazione personale (PII) devono essere crittografate. Per impostazione predefinita, tutte le connessioni devono essere HealthLake utilizzateHTTPS. TLS HealthLake archivia i contenuti crittografati dei clienti e opera secondo il principio di responsabilità AWS condivisa.

Prezzi

Per informazioni sui HealthLake prezzi, consulta la pagina AWS HealthLake dei prezzi. Per stimare meglio i costi potenziali associati a HealthLake, puoi utilizzare il calcolatore HealthLake dei prezzi.

Come AWS HealthLake funziona

AWS HealthLake crea un archivio dati che archivia le cartelle cliniche utilizzando la specifica Healthcare Interoperability FHIR (R4). Con HealthLake, è possibile eseguire le seguenti attività.

> **Note**
>
> Dopo il 20 febbraio 2023, gli archivi HealthLake dati non utilizzano l'elaborazione integrata del linguaggio naturale (NLP) per impostazione predefinita. Se sei interessato ad attivare questa funzionalità sul tuo data store, consulta il Come posso attivare la funzionalità integrata HealthLake di elaborazione del linguaggio naturale? capitolo Risoluzione dei problemi.

- Crea, monitora ed elimina un archivio dati.
- `StartFHIRImportJob`Utilizzalo per importare dati sanitari in blocco da un bucket Amazon Simple Storage Service (Amazon S3) in un data store.
- Usa le operazioni di creazione, lettura, aggiornamento ed eliminazione (CRUD) per gestire i dati archiviati nel tuo archivio dati.
- Utilizzalo SQL in Amazon Athena per interrogare il tuo archivio dati.
- Usa un HTTP client nelle FHIR REST API operazioni per effettuare ricerche nel tuo archivio dati.
- Consenti alle operazioni di Amazon Comprehend API Medical di cercare approfondimenti medici nei tuoi dati utilizzando l'elaborazione del linguaggio naturale NLP ().

Creazione e monitoraggio di archivi dati

Con HealthLake, puoi creare e monitorare archivi di dati in grado di archiviare i dati di Fast Healthcare Interoperability Resources (FHIR).

Per creare un nuovo archivio dati, puoi usare C reateFHIRDatastore o la HealthLake console. Per vedere lo stato di un data store, usa escribeFHIRDatastoreD. Per visualizzare lo stato di più archivi dati attivi, usa istFHIRDatastoresL. Per eliminare un archivio dati, usa eleteFHIRDatastoreD.

FHIRRESTAPIoperazioni

È possibile utilizzare le FHIR REST API operazioni per eseguire operazioni di creazione, lettura, aggiornamento, eliminazione (CRUD) sull'archivio HealthLake dati. Per ulteriori informazioni su come

HealthLake supporta le FHIR REST API operazioni, consultaUtilizzo FHIR REST API delle interazioni con un archivio HealthLake dati.

Generazione automatizzata di FHIR DocumentReference risorse da estensioni di risorse

> **Note**
>
> Quando HealthLake crei un archivio dati e aggiungi dati che lo contengonoDocumentReference, verranno addebitati degli addebiti sul tuo AWS account. Per ulteriori dettagli, consulta AWS HealthLake la pagina dei prezzi.

HealthLake fornisce NLP informazioni sui documenti presenti nel tipo di DocumentReference risorsa. Per analizzare il testo, HealthLake utilizza le seguenti operazioni di Amazon Comprehend API Medical.

- DetectEntitiesV2: esamina il testo clinico per una serie di entità mediche e restituisce informazioni specifiche su di esse, come la categoria dell'entità, l'ubicazione e il punteggio di fiducia.
- InferICD10CM: ispeziona il testo clinico per rilevare le condizioni mediche come entità elencate nella cartella clinica di un paziente e collega tali entità agli identificatori concettuali normalizzati presenti nella knowledge base ICD -10-CM dei Centers for Disease Control.
- InferRxNorm: Esamina il testo clinico per individuare i farmaci come entità elencate nella cartella clinica di un paziente e collega gli identificatori concettuali normalizzati presenti nel database della National Library of Medicine. RxNorm

HealthLake analizza automaticamente i dati trovati nel tipo di DocumentReference risorsa quando viene aggiunto all'archivio dati. I file di DocumentReference risorse originali rimangono invariati. Le informazioni mediche estratte vengono aggiunte automaticamente come FHIR estensioni conformi. Per ulteriori informazioni su come funziona NLP in, consulta. HealthLake Utilizzo della generazione automatizzata di risorse basata sull'elaborazione del linguaggio naturale (NLP) del tipo di FHIR DocumentReference risorsa in AWS HealthLake

Ricerca con interrogazioni SQL basate

> **Note**
>
> Per gli archivi dati creati prima del 14 novembre 2022, la ricerca è limitata alle FHIR REST API operazioni. Per utilizzare le query SQL basate sui dati nel tuo archivio HealthLake dati, consultaEsegui query negli archivi AWS HealthLake dati utilizzando SQL Amazon Athena.

Amazon Athena è un servizio di query SQL basato su server. HealthLake gli archivi di dati vengono inseriti in Athena come tabelle Apache Iceberg. Queste tabelle sono progettate per supportare set di dati analitici di grandi dimensioni. In Athena, ogni tipo di FHIR risorsa è rappresentato come una tabella. Utilizzando Athena, puoi effettuare `READ` richieste solo sul tuo data store. Per ulteriori informazioni sulla ricerca SQL basata, consultaInterroga il tuo archivio HealthLake dati utilizzando SQL.

Ricerca con FHIR REST API operazioni

È possibile cercare nelle cartelle cliniche archiviate nel data store specificando un tipo di risorsa con parametri di ricerca supportati o utilizzando un ID di risorsa trovato nel server, senza specificare il tipo di risorsa. Per ulteriori informazioni sulla ricerca utilizzando le FHIR REST API operazioni, vedere. Utilizzo FHIR REST API delle interazioni con un archivio HealthLake dati

Azioni per l'importazione di dati

AWS HealthLake Usalo per importare i tuoi file in blocco da un bucket Amazon S3. Usa la console o S tartFHIRImport Job per iniziare un processo di importazione. Dopo aver importato i file, puoi usare D escribeFHIRImport Job per monitorare lo stato del lavoro. Una volta completato il processo di importazione, i dati possono essere aggiunti ad Athena, trasformati o analizzati e utilizzati nelle applicazioni downstream.

Azioni per l'esportazione dei dati

HealthLake Usalo per esportare i tuoi file in blocco in un bucket Amazon S3. Usa la console o S tartFHIRExport Job per iniziare un processo di esportazione. Dopo aver esportato i file, puoi utilizzare D escribeFHIRExport Job per monitorare lo stato del lavoro e visualizzarne le proprietà. Una volta

completato il processo di esportazione, puoi visualizzare i dati utilizzando Amazon QuickSight o accedervi utilizzando altri AWS servizi.

AWS HealthLake convalide dei FHIR profili supportate

HealthLake supporta la specifica FHIR R4 di base. Nella specifica R4 sono FHIR inclusi i profili. I profili vengono utilizzati su un tipo di FHIR risorsa per definire una definizione più specifica del tipo di risorsa utilizzando vincoli e/o estensioni sul tipo di risorsa di base. Ad esempio, un FHIR profilo può identificare campi obbligatori come estensioni e set di valori. Una risorsa può supportare più profili. Tutti gli archivi HealthLake dati supportano l'utilizzo FHIR dei profili.

L'aggiunta FHIR di un profilo non è necessaria quando si aggiungono dati a un HealthLake data store. Se non viene specificato alcun FHIR profilo quando una risorsa viene aggiunta o aggiornata, la risorsa viene convalidata solo rispetto allo schema FHIR R4 di base.

FHIRI profili a cui una risorsa è conforme vengono inclusi nella risorsa prima che venga inserita. HealthLake HealthLake convalida i FHIR profili specificati quando vengono aggiunti all'archivio dati. HealthLake

FHIRI profili sono specificati in una guida all'implementazione. HealthLake convalida i FHIR profili definiti nelle seguenti guide all'implementazione.

FHIRProfili supportati da HealthLake

Nome	Versi	Guida all'implementazione	Funzionalità
US Core	3.1.1	http://hl7.org/fhir/us/core/STU3.1.1/	Predefinita
U.S. Core	4.0.0	https://hl7.org/fhir/us/core/STU4/index.html	Supportato
CARINBottone blu	1.1.0	http://hl7.org/fhir/us/carin-bb/STU1.1/	Predefinita
CARINBottone blu	1.0.0	https://hl7.org/fhir/us/carin-bb/STU1/	Supportato
Da Vinci Payer Data Exchange	1.0.0	https://hl7.org/fhir/us/davinci-pdex/	Predefinita
Da Vinci Health Record Exchange () HRex	0.2.0	https://hl7.org/fhir/us/davinci-hrex/2020Sep/	Predefinita

Nome	Versi	Guida all'implementazione	Funzionalità
DaVinci PDEXPiano Net	1.1.0	https://hl7.org/fhir/us/davinci-pdex-plan-net/STU1.1/	Predefinita
DaVinci PDEXPiano Net	1.0.0	https://hl7.org/fhir/us/davinci-pdex-plan-net/STU1/	Supportato
DaVinci Payer Data Exchange (PDex) US Drug Formulary	1.1.0	https://hl7.org/fhir/us/davinci-drug-formulary/STU1.1/	Predefinita
DaVinci Payer Data Exchange (PDex) US Drug Formulary	1.0.1	https://hl7.org/fhir/us/davinci-drug-formulary/STU1.0.1/	Supportato
Missione digitale Ayushman Bharat della National Health Authority () ABDM	2.0	https://www.nrces.in/ndhm/fhir/r4/index.html	Predefinita

Convalida dei profili specificati in una risorsa FHIR

Per convalidare un FHIR profilo, aggiungilo all'`profile`elemento delle singole risorse utilizzando il profilo URL indicato nella guida all'implementazione.

FHIRI profili vengono convalidati quando aggiungi una nuova risorsa al tuo data store. Per aggiungere una nuova risorsa, puoi utilizzare l'APIoperazione S tartFHIRImport Job, fare una `POST` richiesta per aggiungere una nuova risorsa o `PUT` aggiornare una risorsa esistente.

Example — Per vedere a quale FHIR profilo viene fatto riferimento in una risorsa

Il profilo URL viene aggiunto all'`profile`elemento nella coppia `"meta" : "profile"` chiave-valore. Questa risorsa è stata troncata per motivi di chiarezza.

```
{
 "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 — Come fare riferimento a un profilo supportato non predefinito FHIR

Per eseguire la convalida rispetto a un profilo non predefinito supportato (ad es. CarinBB 1.0.0) - aggiungi il profilo URL con la versione (separata da '|') e il profilo di base nell'elemento. URL `meta.profile` Questa risorsa di esempio è stata troncata per motivi di chiarezza.

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

Tipi di dati precaricati

HealthLake supporta solo SYNTHEA come tipo di dati precaricato. Synthea è un generatore sintetico di pazienti che modella la storia medica dei pazienti generati da modelli. È un repository Git open source che consente di generare pacchetti di risorse FHIR conformi HealthLake a R4 in modo che gli utenti possano testare i modelli senza utilizzare i dati effettivi dei pazienti.

I seguenti tipi di risorse sono disponibili negli archivi dati precaricati.

Tipi di risorse Synthea supportati

AllergyIntolerance	Ubicazione
CarePlan	MedicationAdministration
CareTeam	MedicationRequest
Richiedi	Osservazione
Condizione	Organizzazione
Dispositivo	Paziente
DiagnosticReport	Professionista
Incontro	PractitionerRole
ExplanationofBenefit	Procedura
ImagingStudy	Provenienza
Immunizzazione	

Configurazione delle autorizzazioni per iniziare a utilizzare AWS HealthLake

In questo capitolo, si utilizza il AWS Management Console per impostare le autorizzazioni necessarie per iniziare a utilizzare AWS HealthLake e creare un data store. Per impostare le autorizzazioni per creare un data store, è necessario creare un IAM utente o un ruolo che sia amministratore e HealthLake amministratore del data lake. Rendi questo utente un amministratore di data lake in AWS Lake Formation. L'amministratore del data lake concede a Lake Formation l'accesso alle risorse necessarie per utilizzare Amazon Athena per interrogare un data store.

Dopo aver creato un data store in HealthLake, puoi impostare le autorizzazioni per importare file nel data store o esportarli. Per informazioni sulla configurazione delle autorizzazioni per importare file, consulta. Impostazione delle autorizzazioni per i lavori di importazione Per informazioni sull'impostazione delle autorizzazioni per l'esportazione di file, vedere. Impostazione delle autorizzazioni per i lavori di esportazione

Argomenti

- Registrati per un Account AWS
- Crea un utente con accesso amministrativo
- Configura un IAM utente o un ruolo da utilizzare HealthLake (amministratore) IAM
- Aggiungere un utente o un ruolo come amministratore del Data Lake in Lake Formation (IAMamministratore)

Registrati per un Account AWS

Se non ne hai uno Account AWS, completa i seguenti passaggi per crearne uno.

Per iscriverti a un Account AWS

1. Apri la https://portal.aws.amazon.com/billing/registrazione.
2. Segui le istruzioni online.

   Nel corso della procedura di registrazione riceverai una telefonata, durante la quale sarà necessario inserire un codice di verifica attraverso la tastiera del telefono.

Quando ti iscrivi a un Account AWS, Utente root dell'account AWSviene creato un. L'utente root dispone dell'accesso a tutte le risorse e tutti i Servizi AWS nell'account. Come best practice di sicurezza, assegna l'accesso amministrativo a un utente e utilizza solo l'utente root per eseguire attività che richiedono l'accesso di un utente root.

AWS ti invia un'email di conferma dopo il completamento della procedura di registrazione. In qualsiasi momento, puoi visualizzare l'attività corrente del tuo account e gestirlo accedendo a https://aws.amazon.com/e scegliendo Il mio account.

Crea un utente con accesso amministrativo

Dopo esserti registrato Account AWS, proteggi Utente root dell'account AWS AWS IAM Identity Center, abilita e crea un utente amministrativo in modo da non utilizzare l'utente root per le attività quotidiane.

Proteggi i tuoi Utente root dell'account AWS

1. Accedi AWS Management Consolecome proprietario dell'account scegliendo Utente root e inserendo il tuo indirizzo Account AWS email. Nella pagina successiva, inserisci la password.

   Per informazioni sull'accesso utilizzando un utente root, consulta la pagina Signing in as the root user della Guida per l'utente di Accedi ad AWS .
2. Attiva l'autenticazione a più fattori (MFA) per il tuo utente root.

   Per istruzioni, consulta Abilitare un MFA dispositivo virtuale per l'utente Account AWS root (console) nella Guida per l'IAMutente.

Crea un utente con accesso amministrativo

1. Abilita IAM Identity Center.

   Per istruzioni, consulta Abilitazione di AWS IAM Identity Center nella Guida per l'utente di AWS IAM Identity Center .
2. In IAM Identity Center, concedi l'accesso amministrativo a un utente.

Per un tutorial sull'utilizzo di IAM Identity Center directory come fonte di identità, consulta Configurare l'accesso utente con i valori predefiniti IAM Identity Center directory nella Guida per l'AWS IAM Identity Center utente.

Accesso come utente amministratore

- Per accedere con l'utente dell'IAMIdentity Center, utilizza l'accesso URL che è stato inviato al tuo indirizzo e-mail quando hai creato l'utente IAM Identity Center.

  Per informazioni sull'accesso con un utente di IAM Identity Center, consulta Accesso al portale di AWS accesso nella Guida per l'Accedi ad AWS utente.

Assegna l'accesso a ulteriori utenti

1. In IAM Identity Center, crea un set di autorizzazioni che segua la migliore pratica di applicazione delle autorizzazioni con privilegi minimi.

   Segui le istruzioni riportate nella pagina Creazione di un set di autorizzazioni nella Guida per l'utente di AWS IAM Identity Center .
2. Assegna al gruppo prima gli utenti e poi l'accesso con autenticazione unica (Single Sign-On).

   Per istruzioni, consulta Aggiungere gruppi nella Guida per l'utente di AWS IAM Identity Center .

Configura un IAM utente o un ruolo da utilizzare HealthLake (amministratore) IAM

> Persona: IAM amministratore
>
> Un utente che può creare IAM utenti e ruoli e aggiungere amministratori del data lake.

I passaggi descritti in questo argomento devono essere eseguiti da un IAM amministratore.

Per connettere il tuo HealthLake data store ad Athena, devi creare un IAM utente o un ruolo che sia un amministratore del data lake e un HealthLake amministratore. Questo nuovo utente o ruolo

concede l'accesso alle risorse presenti in un data store tramite AWS Lake Formation e la policy `AmazonHealthLakeFullAccess` AWS gestita viene aggiunta al relativo utente o ruolo.

> **Important**
>
> Un IAM utente o un ruolo che è un amministratore di data lake non può creare nuovi amministratori di data lake. Per aggiungere un amministratore del data lake aggiuntivo, è necessario utilizzare un IAM utente o un ruolo a cui è stato concesso `AdministratorAccess` l'accesso.

Per creare un amministratore

1. Aggiungi la politica **`AmazonHealthlakeFullAccess`** IAM AWS gestita a un utente o a un ruolo nell'organizzazione.

   Se non hai dimestichezza con la creazione di un IAM utente, consulta Creazione di un IAM utente e Panoramica delle AWS IAM politiche nella Guida per l'IAMutente.
2. Concedi all'IAMutente o al ruolo l'accesso a AWS Lake Formation.

   - Aggiungi la seguente politica IAM AWS gestita a un utente o a un ruolo nella tua organizzazione: **`AWSLakeFormationDataAdmin`**

     > **Note**
     >
     > La `AWSLakeFormationDataAdmin` politica garantisce l'accesso a tutte le risorse di AWS Lake Formation. È consigliabile utilizzare sempre le autorizzazioni minime necessarie per eseguire l'attività. Per ulteriori informazioni, consulta IAMBest Practices nella Guida per l'IAMutente.
3. Aggiungi la seguente politica in linea all'utente o al ruolo. Per ulteriori informazioni, consulta Politiche in linea nella Guida per l'IAMutente.

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

Per ulteriori informazioni sulla `AWSLakeFormationDataAdmin` politica, consulta Lake Formation Personas and IAM Permissions Reference nella AWS Lake Formation Developer Guide.

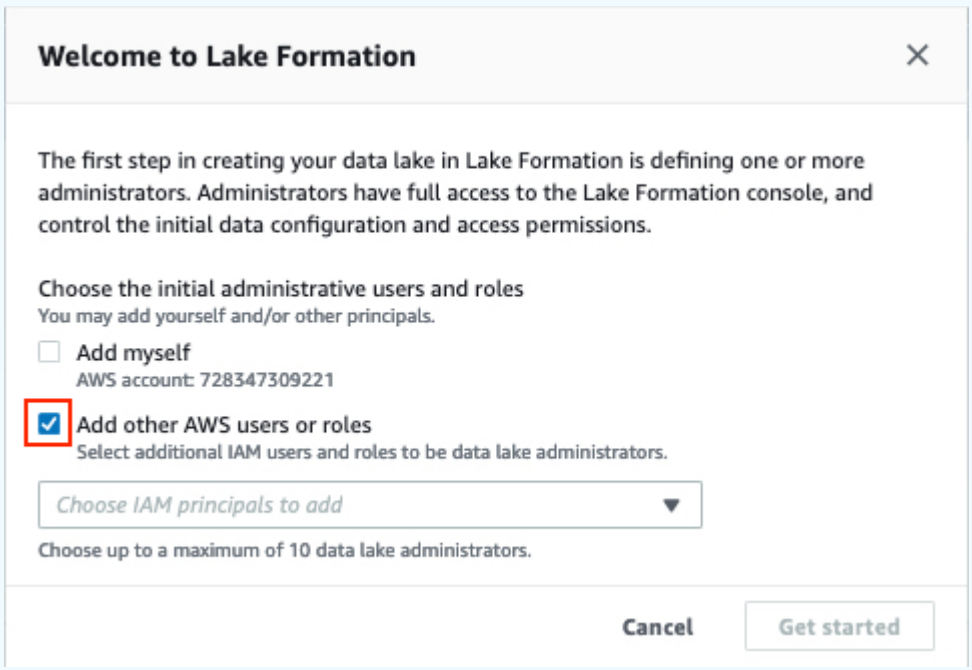
Aggiungere un utente o un ruolo come amministratore del Data Lake in Lake Formation (IAMamministratore)

Successivamente, l'IAMamministratore deve aggiungere l'utente o il ruolo creato nel passaggio 1 come amministratore del data lake in Lake Formation.

Per aggiungere un IAM utente o un ruolo come amministratore del data lake

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

> **Note**
>
> Se è la prima volta che visiti Lake Formation, viene visualizzata una finestra di dialogo Welcome to Lake Formation che ti chiede di definire un amministratore di Lake Formation.



2. Assegna al nuovo utente o ruolo l'amministratore del data AWS lake di Lake Formation.
   - Opzione 1: se hai ricevuto la finestra di dialogo Welcome to Lake Formation.
     1. Scegli Aggiungi altri AWS utenti o ruoli.
     2. Scegli la freccia rivolta verso il basso (▼).
     3. Scegli l' HealthLake amministratore di cui vorresti che diventasse anche amministratore di Lake Formation.
     4. Scegli Avvia.
   - Opzione 2: utilizzare il pannello di navigazione (☰).
     1. Scegli il pannello di navigazione (☰).
     2. In Autorizzazioni, scegli Ruoli e attività amministrative.
     3. Nella sezione Amministratori di Data lake, seleziona Scegli amministratori.
     4. Nella finestra di dialogo Gestisci gli amministratori del data lake, scegli la freccia rivolta verso il basso (▼).
     5. Successivamente, seleziona o cerca gli HealthLake amministratori, gli utenti o i ruoli che desideri siano anche amministratori di Lake Formation.
     6. Seleziona Salva.
3. Modifica le impostazioni di sicurezza predefinite che devono essere gestite da Lake Formation. Le risorse del HealthLake data store non devono essere gestite da Lake FormationIAM. Per

aggiornare, consulta Modifica del modello di autorizzazione predefinito nella AWS Lake Formation Developer Guide.

Creazione di un archivio dati in AWS HealthLake

Al termineConfigurazione delle autorizzazioni per iniziare a utilizzare AWS HealthLake, sei pronto per creare un archivio dati. In AWS HealthLake, si utilizza un data store per archiviare i dati in formato HL7 FHIR (R4). Gli argomenti di questo capitolo descrivono come creare un data store.

Per creare archivi dati abilitati all'analisi e per concedere l'accesso ad essi in Athena, aggiungi la policy `AWSLakeFormationDataAdmin` gestita al tuo IAM utente, gruppo o ruolo. La `AWSLakeFormationDataAdmin` policy consente di creare amministratori di data lake e di concedere l'accesso ai data store in Athena. Per informazioni sull'impostazione delle autorizzazioni, consulta. Configurazione delle autorizzazioni per iniziare a utilizzare AWS HealthLake

HealthLake è inoltre integrato con AWS CloudTrail. È possibile utilizzare CloudTrail per fornire un registro delle azioni intraprese da un utente, un ruolo o un AWS servizio in HealthLake. CloudTrail registra tutte le API chiamate e le azioni della console relative HealthLake agli eventi. Per ulteriori informazioni, consulta Chiamate AWS HealthLake API di registrazione con AWS CloudTrail.

Per ulteriori informazioni sui tipi di risorse Fast Healthcare Interoperability Resources (FHIR) supportati da HealthLake, vedi. Tipi di FHIR risorse supportati in AWS HealthLake

> ⚠ Compatibilità con Amazon Athena
>
> HealthLake gli archivi dati creati prima del 14 novembre 2022 non possono eseguire SQL query utilizzando Athena. Per utilizzare le funzionalità di ricerca di Athena sul tuo data store preesistente, devi prima migrare i dati in un nuovo data store. Per ulteriori informazioni sulla migrazione di archivi dati preesistenti, consulta. Migrazione di un data store esistente per utilizzare Amazon Athena

Dopo aver creato un data store, puoi ottenerne le proprietà, incluso lo stato, con le operazioni API_DescribeFHIRDatastoreAPI_ListFHIRDatastoreso.html. API In alternativa, puoi trovare gli stati dei data store e altri dettagli nella pagina Data stores della HealthLake console.

Un HealthLake data store può avere i seguenti stati:

- Creazione: il tuo data store è in fase di creazione.
- Attivo: il tuo data store è attivo. È possibile importare ed esportare dati da esso. Puoi anche gestire e cercare le FHIR risorse che hai archiviato nell'archivio dati.

- Eliminazione: il tuo archivio dati viene eliminato.
- Eliminato: il tuo archivio dati è stato eliminato.

Argomenti

Creazione di un archivio dati (AWS Management Console)

> **HealthLake differenze tra console**
>
> La HealthLake console non supporta la creazione di un SMART archivio dati non FHIR abilitato. Per creare un SMART data store non FHIR abilitato, è necessario utilizzare il AWS CLI o uno dei supporti AWS supportatiSDKS. Per ulteriori informazioni, consulta Integrazione SMART con FHIR AWS HealthLake. Inoltre, la console non fa distinzione tra i due tipi di data store supportati da HealthLake quando si visualizza la pagina dei dettagli di un singolo data store.

Per creare un HealthLake data store

1. Apri la HealthLake console a https://console.aws.amazon.com//healthlake/casa.
2. Apri il pannello di navigazione (≡).
3. Quindi, scegli Data Stores.
4. Quindi, scegli Crea Data Store.
5. Nella sezione delle impostazioni del Data Store, per il nome del Data Store, specifica un nome.
6. (Facoltativo) Nella sezione Impostazioni Data Store, per Precarica dati di esempio seleziona la casella di controllo per precaricare i dati Synthea.
   - I dati Synthea sono un set di dati di esempio precaricato. Per ulteriori informazioni, consulta Tipi di dati precaricati.
7. Nella sezione Crittografia Data Store, scegli Usa chiave AWS proprietaria (impostazione predefinita) o Scegli una AWS KMS chiave diversa (avanzata).
8. Nella sezione Tag - opzionale, puoi aggiungere tag al tuo data store.

- Per ulteriori informazioni sull'etichettatura del data store, consultaAggiungere un tag a un archivio dati.

9. Quindi, scegli Crea Data Store. Lo stato dei tuoi archivi dati è disponibile nella pagina Data stores.

Creazione di un archivio dati (AWS CLI e AWS SDKs)

È possibile utilizzare i seguenti esempi di codice per creare un HealthLake data store.

AWS CLI

L'esempio seguente dimostra l'utilizzo dell'`CreateFHIRDatastore`operazione con. AWS CLI Per eseguire l'esempio, è necessario installare. AWS CLI Quando crei il tuo archivio dati, la crittografia a riposo utilizza per impostazione predefinita una KMS chiave di AWS proprietà, se non diversamente specificato. Per ulteriori informazioni sulla crittografia, vedere,REST. HealthLake Crittografia presso for REST AWS HealthLake

L'esempio è formattato per Unix, Linux e macOS. Per Windows, sostituite il carattere di continuazione Unix backslash (\) alla fine di ogni riga con un carattere caret (). ^

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

In caso di successo, si ottiene la seguente risposta. JSON Quando l'archivio dati è pronto per l'acquisizione dei dati, lo stato cambia in`ACTIVE`. Per ulteriori informazioni sull'importazione di dati nel data store, HealthLake consulta. Importazione di file in un archivio HealthLake dati

```
{
    "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/"
}
```

Per visualizzare un elenco di tutti gli archivi di dati e gli archivi dati, è possibile utilizzare l'operazione. `ListFHIRDataStore` È inoltre possibile visualizzare un elenco di archivi dati attivi nella HealthLake console.

Python (boto3)

L'esempio seguente mostra come creare un archivio HealthLake dati utilizzando l'`create_fhir_datastore`operazione. Quando si crea il Data Store Encryption at Rest utilizza per impostazione predefinita una AWS KMS chiave di AWS proprietà, salvo diversa indicazione. Per ulteriori informazioni sulla crittografia, vedere,REST. HealthLake Crittografia presso for REST AWS HealthLake

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

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

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

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

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

    return True
```

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

Un archivio dati può avere uno dei quattro stati. `list_fhir_datastores`Utilizzalo per visualizzare un elenco dei tuoi archivi HealthLake dati indipendentemente dallo stato. Questo esempio mostra come filtrare in base allo stato di un data store.

```
import boto3

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

Per ulteriori informazioni, consulta la `list_fhir_datastore`documentazione di Boto3.

Importazione di file in un archivio HealthLake dati

Al termineCreazione di un archivio dati in AWS HealthLake, puoi importare i file nell'archivio dati da un bucket Amazon Simple Storage Service (Amazon S3). Per importare i file, si avvia un processo di importazione con la HealthLake console o l'`StartFHIRImportJob`APIoperazione.

Quando crei un processo di importazione, specifichi la posizione dei dati di input in Amazon S3, una posizione del bucket Amazon S3 per i file di log di output, IAM un ruolo che consente l'accesso ai bucket e HealthLake una chiave di proprietà o di proprietà del cliente. AWS AWS Key Management Service HealthLake utilizza questa chiave per crittografare i dati nella posizione di origine e verrà utilizzata per decrittografarli per consentirne l'importazione. HealthLake Per informazioni sulla configurazione delle autorizzazioni per i lavori di importazione, consulta. Impostazione delle autorizzazioni per i lavori di importazione Per ulteriori informazioni sulla creazione e l'utilizzo AWS KMS delle chiavi, consulta Creating keys nella AWS Key Management Service Developer Guide.

HealthLake accetta file di input in formato delimitato da nuova riga JSON (`.ndjson`), in cui ogni riga è costituita da una risorsa validaFHIR. È possibile utilizzare le API operazioni `ListFHIRImportJobs` per descrivere `DescribeFHIRImportJob` ed elencare i lavori di importazione in corso.

Per ogni processo di importazione, HealthLake genera un `manifest.json` file. Questo registro descrive sia i successi che gli errori di un processo di importazione. HealthLake invia il file nel bucket Amazon S3 specificato quando crei un processo di importazione. Per ulteriori informazioni, consulta File manifesto JSON.

Puoi mettere in coda i lavori di importazione o esportazione. Questi lavori di importazione o esportazione asincroni vengono elaborati in modo FIFO (First In First Out). È possibile creare, leggere, aggiornare o eliminare FHIR risorse mentre è in corso un processo di importazione o esportazione.

Dopo aver popolato un data store con dati precaricati o importati, puoi iniziare a interrogare il tuo data store utilizzando Amazon SQL Athena. Per ulteriori informazioni, consulta Esegui query negli archivi AWS HealthLake dati utilizzando SQL Amazon Athena.

Argomenti

- Esempio: avvio e monitoraggio dei processi di importazione con AWS CLI

Impostazione delle autorizzazioni per i lavori di importazione

Prima di importare i file in un data store, devi concedere l' HealthLake autorizzazione per accedere ai tuoi bucket di input e output in Amazon S3. Per concedere HealthLake l'accesso, devi creare un ruolo di IAM servizio HealthLake, aggiungere una policy di fiducia al ruolo per concedere le autorizzazioni per HealthLake assumere il ruolo e allegare una policy di autorizzazione al ruolo che gli conceda l'accesso ai tuoi bucket Amazon S3.

Quando crei un processo di importazione, specifichi l'Amazon Resource Name (ARN) di questo ruolo per`DataAccessRoleArn`. Per ulteriori informazioni sui IAM ruoli e sulle politiche di fiducia, consulta IAMRoles.

Dopo aver impostato l'autorizzazione, sei pronto per importare i file nel tuo data store con un processo di importazione. Per ulteriori informazioni, consulta Avvio di un processo di importazione in HealthLake.

Per configurare le autorizzazioni di importazione

1. Se non l'hai già fatto, crea un bucket Amazon S3 di destinazione per i file di log di output. Il bucket Amazon S3 deve trovarsi nella stessa AWS regione del servizio e l'opzione Block Public Access deve essere attivata per tutte le opzioni. Per ulteriori informazioni, consulta Usare Amazon S3 per bloccare l'accesso pubblico. Per la crittografia deve essere utilizzata anche una KMS chiave di proprietà di Amazon o di proprietà del cliente. Per ulteriori informazioni sull'uso KMS delle chiavi, consulta Amazon Key Management Service.
2. Crea un ruolo di servizio di accesso ai dati HealthLake e concedi al HealthLake servizio l'autorizzazione ad assumerlo con la seguente politica di fiducia. HealthLake lo usa per scrivere il bucket Amazon S3 di output.

```
{
    "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. Aggiungi una politica di autorizzazioni al ruolo di accesso ai dati che gli consenta di accedere al bucket Amazon S3. Sostituiscilo `amzn-s3-demo-bucket` con il nome del tuo 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"
    }]
}
```

Avvio di un processo di importazione in HealthLake

Dopo aver creato un archivio dati e impostato le autorizzazioni per i processi di importazione (Impostazione delle autorizzazioni per i lavori di importazione), puoi iniziare a importare file con un processo di importazione. È possibile avviare un processo di importazione utilizzando la AWS HealthLake console o l' AWS HealthLake importazioneAPI,. start-fhir-import-jobAPI

Argomenti

- Importazione di file con operazioni API
- Avvio di un processo di importazione (console)

Importazione di file con operazioni API

Prerequisiti

Quando si utilizzano le AWS HealthLake API operazioni, è necessario innanzitutto creare una politica AWS Identity and Access Management (IAM) e associarla a un IAM ruolo. Per ulteriori informazioni sui IAM ruoli e sulle politiche di fiducia, consulta IAMPolitiche e autorizzazioni. I clienti devono inoltre utilizzare una KMS chiave per la crittografia. Per ulteriori informazioni sull'uso KMS delle chiavi, consulta Amazon Key Management Service.

Per importare file (API), usa i seguenti passaggi.

1. Carica i tuoi dati in un bucket Amazon S3.
2. Usa l'operazione. `start-fhir-import-job API` API All'avvio del processo, specifica il nome del bucket Amazon S3 che contiene i file di input, la KMS chiave che desideri utilizzare per la crittografia e la configurazione dei dati di output.
3. Per ulteriori informazioni su un processo di FHIR importazione, utilizza l'describe-fhir-import-joboperazione per ottenere l'ID, il nome, ARN l'ora di inizio, l'ora di fine e lo stato corrente del lavoro. list-fhir-import-jobDa utilizzare per mostrare tutti i processi di importazione e i relativi stati.

Avvio di un processo di importazione (console)

Per importare file con la console, carichi i dati in un bucket Amazon S3,

Per importare file, segui i seguenti passaggi.

1. Carica i tuoi dati in un bucket Amazon S3.
2. Apri la HealthLake console a casa https://console.aws.amazon.com//healthlake/.
3. Vai alla pagina dei dettagli del data store del tuo data store e scegli Importa.
4. Specificate il vostro bucket Amazon S3 e create o identificate il IAM ruolo e la KMS chiave che desiderate utilizzare.
5. Scegli Import data (Importa dati).

File manifesto JSON

Per ogni processo di importazione, HealthLake genera un `manifest.json` file. HealthLake invia il file nel bucket Amazon S3 specificato quando crei un processo di importazione.

Il `manifest.json` file descrive sia i successi che gli errori di un processo di importazione. I file di registro sono organizzati in due cartelle, `SUCCESS` denominate e. `FAILURE` Un file di output può contenere informazioni sensibili, pertanto, quando crei un processo di importazione devi fornire sia un bucket Amazon S3 di output che una AWS KMS chiave per la crittografia.

Di seguito è riportato un esempio del file di output`manifest.json`. Si consiglia di utilizzare questo file come primo passo per la risoluzione dei problemi relativi a un processo di importazione non riuscito. Fornisce dettagli su ciascun file e sulle cause del fallimento del processo di importazione.

```
  {
  "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
}
```

Esempio: avvio e monitoraggio dei processi di importazione con AWS CLI

L'esempio seguente mostra come utilizzare per AWS Command Line Interface avviare e monitorare un processo di importazione. Puoi anche utilizzare l'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
```

All'inizio del processo di importazione, riceverai la seguente conferma.

```
{
```

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

Per monitorare lo stato di un processo di importazione o per conoscerne le proprietà di configurazione, utilizzate il AWS CLI comando describe-fhir-import-jobAPIo, come illustrato nell'esempio seguente.

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

In risposta, riceverete le seguenti informazioni.

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

Per visualizzare un elenco di tutti i processi di importazione, utilizzate il AWS CLI comando list-fhir-import-jobsAPIo, come illustrato nell'esempio seguente. È possibile aggiungere uno o più filtri per limitare i risultati.

```
    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)
```

Riceverai le seguenti informazioni in risposta.

```
{
    "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
```

Esportazione di file da un archivio HealthLake dati

Dopo aver creato un archivio dati e importato i dati (o se utilizzi dati di esempio precaricati), puoi esportare i dati in un bucket Amazon S3. Per esportare i dati dal tuo HealthLake data store, usa le seguenti operazioni.

- Effettua una richiesta di esportazione utilizzando l'`StartFHIRExportJob`APIoperazione che utilizza AWS SDKs and HealthLake.
  - Questa operazione supporta solo la creazione di una richiesta di esportazione a livello di sistema.
- Effettua una richiesta di esportazione utilizzando la `export` sintassi utilizzando. HealthLake FHIR REST API
  - Questa operazione supporta l'invio di richieste di esportazione a livello di sistema, di pazienti e di gruppo. È inoltre possibile applicare parametri per filtrare ulteriormente i dati nella richiesta di esportazione.

> **Important**
>
> HealthLake SDKle richieste di esportazione che utilizzano `StartFHIRExportJob` API l'operazione e le richieste di FHIR REST API esportazione che utilizzano `StartFHIRExportJobWithPost` API l'operazione hanno IAM azioni separate. Per ogni IAM azione, SDK esporta con `StartFHIRExportJob` ed FHIR REST API esporta con`StartFHIRExportJobWithPost`, le autorizzazioni di consentimento/rifiuto possono essere gestite separatamente. Se vuoi che entrambe le SDK FHIR REST API esportazioni siano limitate, assicurati di negare le autorizzazioni per ogni azione. IAM

Entrambe queste operazioni supportano solo l'esportazione dei file in un bucket Amazon S3 (S3). Tutti i file del tuo HealthLake data store vengono esportati come file delimitati da nuova riga JSON (`.ndjson`), in cui ogni riga è costituita da una risorsa valida. FHIR

Entrambe queste operazioni richiedono un ruolo di servizio. In esso, HealthLake deve essere definito come responsabile del servizio ed è necessario definire un bucket Amazon Simple Storage Service (S3) in cui esportare i file. Per ulteriori informazioni, consulta Impostazione delle autorizzazioni per i lavori di esportazione.

Puoi mettere in coda i lavori di importazione o esportazione. Questi lavori di importazione o esportazione asincroni vengono elaborati in modo FIFO (First In First Out). È possibile creare, leggere, aggiornare o eliminare FHIR risorse mentre è in corso un processo di importazione o esportazione.

Per esportare file dal tuo archivio HealthLake dati, consulta le seguenti sezioni.

- Impostazione delle autorizzazioni per i lavori di esportazione
- Esportazione di file dal tuo data store con la HealthLake console o AWS SDKs
- Esportazione di dati dal tuo archivio HealthLake dati con operazioni FHIR REST API

Impostazione delle autorizzazioni per i lavori di esportazione

Prima di esportare file da un data store, devi concedere l' HealthLake autorizzazione per accedere al tuo bucket di output in Amazon S3. Per concedere HealthLake l'accesso, devi creare un ruolo di IAM servizio HealthLake, aggiungere una policy di fiducia al ruolo per concedere le autorizzazioni per HealthLake assumere il ruolo e allegare una policy di autorizzazione al ruolo che gli conceda l'accesso al tuo bucket Amazon S3.

Se hai già creato un ruolo per HealthLake inImpostazione delle autorizzazioni per i lavori di importazione, puoi riutilizzarlo e concedergli le autorizzazioni aggiuntive per il tuo bucket di esportazione Amazon S3 elencate in questo argomento. Per ulteriori informazioni sui IAM ruoli e sulle politiche di fiducia, consulta IAM Politiche e autorizzazioni.

> **Important**
>
> HealthLake SDKle richieste di esportazione che utilizzano `StartFHIRExportJob` API l'operazione e le richieste di FHIR REST API esportazione che utilizzano `StartFHIRExportJobWithPost` API l'operazione hanno IAM azioni separate. Per ogni IAM azione, SDK esporta con `StartFHIRExportJob` ed FHIR REST API esporta con`StartFHIRExportJobWithPost`, le autorizzazioni di consentimento/rifiuto possono essere gestite separatamente. Se vuoi che entrambe le SDK FHIR REST API esportazioni siano limitate, assicurati di negare le autorizzazioni per ogni azione. IAM Se concedi agli utenti l'accesso completo a HealthLake, non è richiesta alcuna modifica delle autorizzazioni IAM utente.

L'utente o il ruolo che imposta le autorizzazioni deve disporre dell'autorizzazione a creare ruoli, creare politiche e allegare politiche ai ruoli. La seguente IAM politica concede queste autorizzazioni.

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

Per impostare le autorizzazioni di esportazione

1. Se non l'hai già fatto, crea un bucket Amazon S3 di destinazione per i dati che esporterai dal tuo data store. Il bucket Amazon S3 deve trovarsi nella stessa AWS regione del servizio e l'opzione Block Public Access deve essere attivata per tutte le opzioni. Per ulteriori informazioni, consulta Usare Amazon S3 per bloccare l'accesso pubblico. Per la crittografia deve essere utilizzata anche una KMS chiave di proprietà di Amazon o di proprietà del cliente. Per ulteriori informazioni sull'uso KMS delle chiavi, consulta Amazon Key Management Service.
2. Se non l'hai già fatto, crea un ruolo del servizio di accesso ai dati HealthLake e concedi al HealthLake servizio l'autorizzazione ad assumerlo con la seguente politica di fiducia. HealthLake lo usa per scrivere il bucket Amazon S3 di output. Se ne hai già creato unoImpostazione delle autorizzazioni per i lavori di importazione, puoi riutilizzarlo e concedergli le autorizzazioni per il tuo bucket Amazon S3 nel passaggio successivo.

   ```
   {
       "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. Aggiungi una politica di autorizzazioni al ruolo di accesso ai dati che gli consenta di accedere al tuo bucket Amazon S3 di output. Sostituiscilo `amzn-s3-demo-bucket` con il nome del bucket.

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

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

Esportazione di file dal tuo data store con la HealthLake console o AWS SDKs

Al termineImpostazione delle autorizzazioni per i lavori di esportazione, puoi esportare i file dal tuo data store in un bucket Amazon Simple Storage Service (Amazon S3). Per esportare file da un data store, devi avviare un processo di esportazione in. HealthLake Un processo di esportazione esporta i file dall'archivio dati in formato delimitato da nuova riga JSON (`.ndjson`), in cui ogni riga è costituita da una risorsa validaFHIR. Quando si avvia un processo di esportazione, è necessario specificare una AWS KMS chiave per la crittografia. Per ulteriori informazioni sulla creazione di una KMS chiave, consulta Creating keys nella AWSKey Management Service Developer Guide.

I seguenti argomenti spiegano come avviare un processo di esportazione con la AWS HealthLake console e poi AWS SDKs con l'start-fhir-export-jobAPIoperazione.

Argomenti

- Esportazione di file dal tuo archivio dati (console)
- Esportazione di file dal tuo archivio dati ()AWS SDKs

Esportazione di file dal tuo archivio dati (console)

Per esportare file (console), utilizza la procedura seguente.

1. Crea un bucket S3 di output nella stessa regione di. HealthLake
2. Per iniziare un nuovo processo di esportazione, identifica il bucket Amazon S3 di output e crea o identifica il IAM ruolo che desideri utilizzare. Per ulteriori informazioni sui IAM ruoli e sulle politiche di fiducia, consulta IAM ruoli. Utilizza anche una crittografia a KMS chiave. Per ulteriori informazioni sull'uso KMS delle chiavi, consulta Amazon Key Management Service.

3. Per vedere lo stato del tuo processo di esportazione, usa `ListFHIRExportJobs` API operation.

Esportazione di file dal tuo archivio dati ()AWS SDKs

Per esportare file dal tuo archivio dati con AWS SDKs, usa l'start-fhir-export-joboperazione. Il codice seguente mostra come avviare un processo di esportazione con 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'])
```

Per ottenere l'ID, il nomeARN, l'ora di inizio, l'ora di fine e lo stato corrente di un processo di FHIR esportazione, usa. describe-fhir-export-job list-fhir-export-jobsDa utilizzare per elencare tutti i lavori di esportazione e i relativi stati.

Il codice seguente mostra come ottenere le proprietà di uno specifico processo di esportazione con 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'])
```

Esportazione di dati dal tuo archivio HealthLake dati con operazioni FHIR REST API

Al termineImpostazione delle autorizzazioni per i lavori di esportazione, è possibile esportare i dati dall'archivio HealthLake dati con FHIR REST API operazioni. Per effettuare una richiesta di esportazione utilizzando il FHIR RESTAPI, è necessario disporre di un IAM utente, gruppo o ruolo con le autorizzazioni richieste, specificarlo `$export` come parte della `POST` richiesta e includere i parametri della richiesta nel corpo della richiesta. In base alle FHIR specifiche, il FHIR server deve supportare `GET` le richieste e può supportarle`POST`. Per supportare parametri aggiuntivi, è necessario un body per avviare l'esportazione, quindi HealthLake supporta POST le richieste.

> **Important**
>
> HealthLake gli archivi di dati creati prima del 1° giugno 2023 supportano solo le richieste di lavori di esportazione FHIR REST API basate sull'esportazione per le esportazioni a livello di sistema.
> HealthLake gli archivi di dati creati prima del 1° giugno 2023 non supportano l'acquisizione dello stato di un'esportazione utilizzando una `GET` richiesta sull'endpoint di un data store.

Tutte le richieste di esportazione effettuate utilizzando il FHIR REST API vengono restituite in `ndjson` formato ed esportate in un bucket Amazon S3. Ogni oggetto S3 conterrà un solo tipo di risorsa. FHIR

È possibile mettere in coda le richieste di esportazione in base alle quote dell' AWS account. Per ulteriori informazioni sui Service Quotas associati a HealthLake, vedere. AWS HealthLake endpoint e quote

HealthLake supporta i seguenti tre tipi di richieste endpoint di esportazione in blocco.

Tipo	Descrizioni	Sintassi
Esportazione del sistema	Esporta tutti i dati dal HealthLake FHIR server.	`POST https://healthlake.` **`your-region`** `.amazonaws.com/datastore/` **`your-datastore-id`** `/r4/$export`

Tipo	Descrizioni	Sintassi
Tutti i pazienti	Esporta tutti i dati relativi a tutti i pazienti, compresi i tipi di risorse associati al tipo di risorsa Paziente.	`POST https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/Patient/$export`
Gruppo di pazienti	Esporta tutti i dati relativi a un gruppo di pazienti specificato con un ID di gruppo.	`POST https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/Group/ID/$export`

Prima di iniziare

Soddisfa i seguenti requisiti per effettuare una richiesta di esportazione utilizzando il FHIR REST API modulo HealthLake.

- È necessario aver impostato un utente, un gruppo o un ruolo con le autorizzazioni necessarie per effettuare la richiesta di esportazione. Per ulteriori informazioni, consulta Autorizzazione di una richiesta `export`.
- Devi aver creato un ruolo di servizio che garantisca HealthLake l'accesso al bucket Amazon S3 in cui desideri esportare i tuoi dati. Il ruolo di servizio deve inoltre essere specificato HealthLake come principale del servizio. Per ulteriori informazioni sulla configurazione delle autorizzazioni, vedereImpostazione delle autorizzazioni per i lavori di esportazione.

Autorizzazione di una richiesta **`export`**

Per effettuare correttamente una richiesta di esportazione utilizzando il FHIR RESTAPI, autorizza il tuo utente, gruppo o ruolo utilizzando uno dei due IAM o OAuth2 .0. È inoltre necessario avere un ruolo di servizio.

Autorizzazione di una richiesta utilizzando IAM

Quando si effettua una `$export` richiesta, l'utente, il gruppo o il ruolo devono `StartFHIRExportJobWithPost` includere `CancelFHIRExportJobWithDelete` IAM azioni e azioni nella politica. `DescribeFHIRExportJobWithGet`

> **Important**
>
> HealthLake SDKle richieste di esportazione che utilizzano `StartFHIRExportJob` API l'operazione e le richieste di FHIR REST API esportazione che utilizzano `StartFHIRExportJobWithPost` API l'operazione hanno IAM azioni separate. Per ogni IAM azione, SDK esporta con `StartFHIRExportJob` ed FHIR REST API esporta con`StartFHIRExportJobWithPost`, le autorizzazioni di consentimento/rifiuto possono essere gestite separatamente. Se vuoi che entrambe le SDK FHIR REST API esportazioni siano limitate, assicurati di negare le autorizzazioni per ogni azione. IAM

Autorizzazione di una richiesta utilizzando SMART on FHIR (2.0) OAuth

Quando si effettua una `$export` richiesta SMART su un HealthLake data store FHIR abilitato, è necessario che vengano assegnati gli ambiti appropriati. Per ulteriori informazioni sugli ambiti supportati, consulta. HealthLake ambiti specifici FHIR delle risorse dell'archivio dati

Effettuare una richiesta **`export`**

Questa sezione descrive i passaggi necessari da eseguire quando si effettua una richiesta di esportazione utilizzando il FHIR RESTAPI.

Per evitare addebiti accidentali sul tuo AWS account, ti consigliamo di testare le tue richieste effettuando una `POST` richiesta senza fornire la `export` sintassi.

Per effettuare la richiesta, devi fare quanto segue:

1. Specificare `export` nella `POST` URL richiesta un endpoint supportato.
2. Specificate i parametri di intestazione richiesti.
3. Specificate un corpo della richiesta che definisca i parametri richiesti.

Fase 1: Specificare **`export`** nella **`POST`** URL richiesta un endpoint supportato

HealthLake supporta tre tipi di richieste endpoint di esportazione in blocco. Per effettuare una richiesta di esportazione in blocco, è necessario effettuare una richiesta `POST` basata su uno dei tre endpoint supportati. Gli esempi seguenti mostrano come specificare `export` nella richiesta. URL

- `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`

In quella stringa di `POST` richiesta, è possibile utilizzare i seguenti parametri di ricerca supportati.

Parametri di ricerca supportati

HealthLake supporta i seguenti modificatori di ricerca nelle richieste di esportazione in blocco.

Questi esempi includono caratteri speciali che devono essere codificati prima di inviare la richiesta.

Nome	Obbligatorio?	Descrizione	Esempio
`_outputFormat`	No	Il formato per i file Bulk Data richiesti da generare. I valori accettati sono`application/fhir+ndjson` ,`application/ndjson` ,`ndjson`.	
`_type`	No	Una stringa di tipi di FHIR risorse delimitati da virgole che desideri includere nel processo di esportazione. Ti consigliamo di includerla `_type` perché ciò può avere un impatto sui costi quando tutte le risorse vengono esportate.	`&_type=MedicationStatement,Observation`

Nome	Obbligatorio?	Descrizione	Esempio
`_since`	No	Tipi di risorse modificati in o dopo la data e l'ora. Se un tipo di risorsa non ha l'ora dell'ultimo aggiornamento, verranno inclusi nella risposta.	`&_since=2024-05-09T00%3A00%3A00Z`

Passaggio 2: Specificare i parametri di intestazione richiesti

Per effettuare una richiesta di esportazione utilizzando FHIR RESTAPI, è necessario specificare i due parametri di intestazione seguenti.

- Tipo di contenuto: `application/fhir+json`
- Preferisco: `respond-async`

Successivamente, è necessario specificare gli elementi richiesti nel corpo della richiesta.

Fase 3: Specificare un corpo della richiesta e definire i parametri richiesti.

La richiesta di esportazione richiede anche un corpo in `JSON` formato. Il corpo può includere i seguenti parametri.

Chiave	Obbligatorio?	Descrizione	Valore
DataAccessRoleArn	Sì	Un ruolo ARN di HealthLake servizio. Il ruolo di servizio utilizzato deve essere specificato HealthLake come principale del servizio.	`arn:aws:iam:: 444455556666 :role/your-healthlake-service-role`

Chiave	Obbligatorio?	Descrizione	Valore
JobName	No	Il nome della richiesta di esportazione.	**your-export-job-name**
S3Uri	Sì	Parte di una OutputDataConfig chiave. L'S3 URI del bucket di destinazione in cui verranno scaricati i dati esportati.	s3://DOC-EXAMPLE-DESTINATION-BUCKET/ **EXPORT-JOB** /
KmsKeyId	Sì	Parte di una chiave. OutputDataConfig La ARN AWS KMS chiave utilizzata per proteggere il bucket Amazon S3.	arn:aws:kms: **region-of-bucket:123456789012** :key/**1234abcd-12ab-34cd-56ef-1234567890ab**

Example — Corpo di una richiesta di esportazione effettuata utilizzando il FHIR REST API

Per effettuare una richiesta di esportazione utilizzando il FHIR RESTAPI, è necessario specificare un corpo, come illustrato di seguito.

```
{
  "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 la richiesta avrà esito positivo, riceverai la seguente risposta.

Intestazione della risposta

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

Corpo di risposta

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

Gestione della richiesta di esportazione

Dopo aver effettuato una richiesta di esportazione corretta, puoi gestirla descrivendo `export` lo stato di una richiesta di esportazione corrente e `export` annullando una richiesta di esportazione corrente.

Quando annulli una richiesta di esportazione utilizzando il RESTAPI, ti verrà addebitata solo la parte dei dati che è stata esportata fino al momento in cui hai inviato la richiesta di annullamento.

I seguenti argomenti descrivono come visualizzare lo stato di una richiesta di esportazione corrente o annullarla.

Annullamento di una richiesta di esportazione

Per annullare una richiesta di esportazione, effettua una `DELETE` richiesta e fornisci l'ID del lavoro nella richiesta. URL

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

Quando la richiesta ha esito positivo, riceverai quanto segue.

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

Quando la tua richiesta non va a buon fine, ricevi quanto segue.

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

Descrizione di una richiesta di esportazione

Per conoscere lo stato di una richiesta di esportazione, effettua una `GET` richiesta utilizzando `export` and your**`export-request-job-id`**.

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

La JSON risposta conterrà un `ExportJobProperties` oggetto. Può contenere le seguenti coppie chiave:valore.

Nome	Obbligatorio?	Descrizione	Valore
DataAccessRoleArn	No	Un ARN ruolo di HealthLake servizio. Il ruolo di servizio utilizzato deve essere specificato HealthLake come principale del servizio.	`arn:aws:iam:: 444455556666 :role/your-healthlake-service-role`
SubmitTime	No	Data e ora in cui è stato inviato un processo di esportazione.	`Apr 21, 2023 5:58:02`

Nome	Obbligatorio?	Descrizione	Valore
EndTime	No	L'ora in cui è stato completato un processo di esportazione.	`Apr 21, 2023 6:00:08 PM`
JobName	No	Il nome della richiesta di esportazione.	**`your-export-job-name`**
JobStatus	No		I valori validi sono: `SUBMITTED | IN_PROGRESS | COMPLETED_WITH_ERRORS | COMPLETED | FAILED`
S3Uri	Sì	Parte di un OutputDataConfigoggetto. L'Amazon S3 URI del bucket di destinazione in cui verranno scaricati i dati esportati.	`s3://DOC-EXAMPLE-DESTINATION-BUCKET/` **`EXPORT-JOB /`**
KmsKeyId	Sì	Parte di un oggetto. OutputDataConfig La ARN AWS KMS chiave utilizzata per proteggere il bucket Amazon S3.	`arn:aws:kms:` **`region-of-bucket:123456789012`** `:key/`**`1234abcd-12ab-34cd-56ef-1234567890ab`**

Example : Corpo di una richiesta di descrizione dell'esportazione effettuata utilizzando il FHIR REST API

In caso di successo, riceverai la seguente JSON risposta.

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

Eliminazione di un archivio dati in HealthLake

L'eliminazione di un archivio dati è un'operazione asincrona. Una volta avviata, lo stato cambia in Eliminazione. Un data store mantiene lo stato di Eliminazione fino alla rimozione di tutti FHIR i dati dal data store e dell'infrastruttura sottostante necessaria.

Dopo la rimozione dei dati e dell'infrastruttura, lo stato del HealthLake data store cambia in Eliminato. Dopo l'eliminazione, i dettagli sugli archivi di dati sono disponibili solo utilizzando le `ListFHIRDataStores` operazioni `DescribeFHIRDataStore` e per sette giorni. Dopo sette giorni, l'archivio dati eliminato non verrà visualizzato nei risultati.

Per eliminare correttamente un data store, è necessario `glue:DeleteDatabase` aggiungere l'IAMazione alla IAM policy dell'utente, del gruppo o del ruolo che effettua la richiesta. Questa IAM azione non è inclusa nella politica AWS gestita,`AmazonHealthLakeFullAccess`.

È possibile eliminare un archivio dati con AWS Management Console AWS SDKs, o AWS CLI.

Argomenti

Eliminazione di un archivio dati (console)

Per eliminare un data store con la console, scegli il tuo data store nella pagina Data Stores e scegli elimina.

Per eliminare un HealthLake data store

1. Apri la HealthLake console a https://console.aws.amazon.com//healthlake/casa.
2. Apri il pannello di navigazione (≡).
3. Quindi, scegli Data Stores.
4. Nella pagina Data Stores, scegli l'opzione accanto al data store che desideri eliminare.
5. Quindi, scegli Elimina
6. Nella finestra di dialogo digitate **`delete`** per confermare che desiderate eliminare l'archivio dati selezionato.
7. Quindi, scegli Elimina. Quindi lo stato del tuo data store cambierà da Attivo a Eliminazione.

Eliminazione di un archivio dati (AWS SDKse) AWS CLI

È possibile utilizzare gli esempi di codice riportati di seguito per eliminare un HealthLake data store.

AWS CLI

Negli esempi seguenti viene illustrato l'utilizzo dell'`DeleteFHIRDatastore`operazione con. AWS CLI Per eseguire l'esempio, è necessario installare. AWS CLI

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

In caso di successo, si ottiene la seguente JSON risposta.

```
{
    "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)

Il AWS SDK for Python supporta il `describe_fhir_datastore` metodo che accetta un singolo parametro. `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)
```

In caso di successo, restituisce un dizionario 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}}
```

Per restituire dettagli su più di un archivio dati alla volta, usa `ListFHIRDatastore`

utilizzate il `DeleteFHIRDataStore` comando utilizzando il AWS CLI come illustrato nell'esempio seguente. È inoltre possibile eliminare un archivio dati utilizzando delete-fhir-datastore APIo la console. L'eliminazione di un data store rimuove tutte le versioni FHIR delle risorse contenute nel data store e nell'infrastruttura sottostante. I log relativi a un data store eliminato vengono conservati all'interno dell'account del servizio in conformità alle linee guida. HIPAA

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

Come illustrato nella JSON risposta di esempio seguente, lo stato cambia in "`DELETING`" per confermare che l'archivio dati e il relativo contenuto sono in fase di eliminazione.

```
{
```

```
  "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)"
}
```

Utilizzo FHIR REST API delle interazioni con un archivio HealthLake dati

In AWS HealthLake, utilizzi le REST API interazioni Fast Healthcare Interoperability Resources (FHIR) per gestire e cercare FHIR risorse nel tuo archivio dati. FHIRRESTAPIle interazioni vengono utilizzate per eseguire interazioni di creazione, lettura, aggiornamento ed eliminazione (CRUD) sulle risorse di un archivio dati. È inoltre possibile creare stringhe di ricerca complesse utilizzando una `POST` HTTP richiesta `GET` o, poiché HealthLake supporta un sottoinsieme di operazioni di ricerca FHIR supportate.

Ai fini della conformità, i tipi di risorse vengono convalidati in base alla FHIR risorsa R4. HL7 FHIR StructureDefinition Per trovare le funzionalità FHIR relative di un archivio HealthLake dati attivo, effettua una `GET` richiesta dove `metadata` è specificato in, come segue. URL

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

In caso di successo, riceverai un codice di `200` HTTP risposta e la dichiarazione di capacità per il tuo HealthLake data store. Per ulteriori informazioni, consulta `CapabilityStatement`la documentazione di HL7 FHIR R4.

La tabella seguente elenca le FHIR interazioni supportate da AWS HealthLake.

FHIRinterazioni supportate da AWS HealthLake

FHIRinterazione	Descrizione
Interazioni dell'intero sistema	
`capabilities`	Ottieni una dichiarazione di capacità per il sistema
`batch/transaction`	Aggiorna, crea o elimina un set di risorse in una singola interazione
Interazioni a livello di tipo	
`create`	Crea una nuova risorsa con un ID assegnato dal server
`search`	Cerca un tipo di risorsa in base ad alcuni criteri di filtro

FHIRinterazione	Descrizione
`history`	Recupera la cronologia delle modifiche per un particolare tipo di risorsa
Interazioni a livello di istanza	
`read`	Leggi lo stato attuale di una risorsa
`history`	Leggi la cronologia delle modifiche per una particolare risorsa
`vread`	Leggi lo stato di una versione specifica della risorsa
`update`	Aggiorna una risorsa in base al suo ID (o creala se è nuova)
`delete`	Eliminazione di una risorsa

Argomenti

- Tipi di FHIR risorse supportati in AWS HealthLake
- Esecuzione di operazioni di creazione, lettura, aggiornamento ed eliminazione (CRUD) sugli archivi HealthLake dati
- Ricerca nell'archivio HealthLake dati utilizzando le FHIR REST API operazioni
- Leggere FHIR la cronologia delle risorse
- Acquisizione dei dati dei pazienti con l'operazione Patient $everything FHIR REST API
- Esportazione di dati dal tuo HealthLake data store usando $export

Tipi di FHIR risorse supportati in AWS HealthLake

La tabella seguente elenca i tipi di risorse FHIR R4 supportati da AWS HealthLake. Per ulteriori informazioni, consulta l'Indice delle risorse nella documentazione di HL7FHIRR4.

FHIRTipi di risorse R4 supportati da HealthLake

Account	DetectedIssue	Fattura	Professionista
ActivityDefinition	Dispositivo	Libreria	PractitionerRole
AdverseEvent	DeviceDefinition	Collegamento	Procedura

AllergyIntolerance	DeviceMetric	Elenco	Provenienza
Appuntamento	DeviceUseStatement	Ubicazione	Questionario
AppointmentResponse	DeviceRequest	Misura	QuestionnaireResponse
AuditEvent- Vedi nota	DiagnosticReport	MeasureReport	RelatedPerson
Binario	DocumentManifest	Media	RequestGroup
BodyStructure	DocumentReference	Farmaco	ResearchStudy
Pacchetto - Vedi nota	EffectEvidenceSynthesis	MedicationAdministration	ResearchSubject
CapabilityStatement	Incontro	MedicationDispense	RiskAssessment
CarePlan	Endpoint	MedicationKnowledge	RiskEvidenceSynthesis
CareTeam	EpisodeOfCare	MedicationRequest	Pianificazione
ChargeItem	EnrollmentRequest	MedicationStatement	ServiceRequest
ChargeItemDefinition	EnrollmentResponse	MessageHeader	Slot
Richiedi	ExplanationOfBenefit	MolecularSequence	Esemplare
ClaimResponse	FamilyMemberHistory	NutritionOrder	StructureDefinition
Communication	Flag	Osservazione	StructureMap
CommunicationRequest	Obiettivo	OperationOutcome	Sostanza
Composizione	Group (Gruppo)	Organizzazione	SupplyDelivery
ConceptMap	GuidanceResponse	OrganizationAffiliation	SupplyRequest
Condizione	HealthcareService	Parametri	Attività

Consenso	ImagingStudy	Paziente	ValueSet
Contratto	Immunizzazione	PaymentNotice	VisionPrescription
Copertura	ImmunizationEvaluation	PaymentReconciliation	VerificationResult
CoverageEligibilityRequest	ImmunizationRecommendation	Person	
CoverageEligibilityResponse	InsurancePlan	PlanDefinition	

⚠ FHIRspecifiche e HealthLake

- Non è possibile effettuare `GET` o `POST` richiedere con questi tipi di risorse: Binary, Bundle e Parameters. OperationOutcome
- AuditEvent— Una AuditEvent risorsa può essere creata o letta, ma non può essere aggiornata o eliminata.
- Bundle: esistono diversi modi per HealthLake gestire le richieste Bundle. Per ulteriori dettagli, consulta Gestione di più FHIR risorse tramite Bundle.
- VerificationResult— Questo tipo di risorsa è supportato solo per gli archivi dati creati dopo il 09 dicembre 2023.

Esecuzione di operazioni di creazione, lettura, aggiornamento ed eliminazione (CRUD) sugli archivi HealthLake dati

Sebbene si utilizzino AWS azioni native per gestire gli archivi dati, importare dati ed esportare dati, si utilizzano quattro FHIR HTTP operazioni principali per creare (`POST`), leggere (`GET`), aggiornare (`PUT`) ed eliminare (`DELETE`) FHIR all'interno di un HealthLake data store. I seguenti argomenti descrivono come eseguire le operazioni di creazione, lettura, aggiornamento ed eliminazione (CRUD) sul HealthLake data store utilizzando i FHIR REST API servizi. È necessario utilizzare un processo di firma Signature Version 4 per autenticare HealthLake API le richieste inviate tramite un HTTP

client. Per ulteriori informazioni, consulta la procedura di firma di Signature Version 4 nel Riferimenti generali di AWS.

Argomenti

Creazione di una risorsa con **`POST`**

Si utilizza una `POST` richiesta per creare una nuova risorsa in un archivio HealthLake dati. `POST`le richieste non richiedono la fornitura di un `id` elemento. I HealthLake server restituiscono un codice di HTTP stato `201` Created quando una risorsa è stata creata con successo.

> **Note**
>
> Quando si effettua una `POST` richiesta sul tipo di `DocumentReference` risorsa, le estensioni esistenti non vengono modificate. Al contrario, AWS HealthLake aggiunge le nuove estensioni con quelle esistenti al tuo archivio dati. Per ulteriori dettagli su come HealthLake utilizza l'elaborazione del linguaggio naturale (NLP) sul tipo di `DocumentReference` risorsa per estrarre dati medici preziosi, consultaUtilizzo della generazione automatizzata di risorse basata sull'elaborazione del linguaggio naturale (NLP) del tipo di FHIR DocumentReference risorsa in AWS HealthLake.

Example Creazione di una **`Patient`** risorsa utilizzando una **`POST`** richiesta.

Per creare una `POST` richiesta di HealthLake data store, usa l'endpoint del tuo data store e fornisci un corpo della JSON richiesta. Per trovare l'endpoint di un data store, cerca nella HealthLake console sotto Data Stores o utilizzando l'escribeFHIRDatastoreoperazione D nella AWS HealthLake API pagina di riferimento.

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"
}
```

Risposta JSON

Per confermare la creazione della risorsa per il paziente, riceverai un codice di HTTP stato 201 creato e la seguente JSON risposta.

```
{
  "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"
  }
}
```

Leggere una risorsa con **GET**

Questo esempio mostra come leggere una FHIR risorsa per un paziente utilizzando una GET richiesta.

Example Leggere una **Patient** risorsa specifica utilizzando una **GET** richiesta.

Per creare una GET richiesta di HealthLake data store, usa l'endpoint del tuo data store. Per trovare l'endpoint di un data store, cerca nella HealthLake console sotto Data Stores o utilizzando l'escribeFHIRDatastoreoperazione D nella AWS HealthLake API pagina di riferimento.

È inoltre necessario includere il tipo di risorsa **Patient** e un identificatore valido,. **2de04858-ba65-44c1-8af1-f2fe69a977d9**

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

Risposta JSON

In caso di successo, riceverai un codice di 200 HTTP stato e la seguente JSON risposta.

```
{
    "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"
}
```

Aggiornamento di una risorsa utilizzando **PUT**

L'esempio seguente mostra come utilizzare per PUT aggiornare i dettagli su un paziente nel tipo di FHIR risorsa paziente. Inoltre, quando si effettua una PUT richiesta su una risorsa non ancora creata, verrà creata una versione iniziale.

La richiesta restituirà un codice di `200` HTTP stato se la risorsa è stata aggiornata o restituirà un codice di `201` HTTP stato se è stata creata una nuova risorsa.

> **Note**
>
> Quando si effettua una PUT richiesta sul tipo di `DocumentReference` risorsa, le estensioni esistenti non vengono modificate. Al contrario, AWS HealthLake aggiunge le nuove estensioni con quelle esistenti al tuo archivio dati. Per ulteriori dettagli su come HealthLake utilizza l'elaborazione del linguaggio naturale (NLP) sul tipo di `DocumentReference` risorsa per estrarre dati medici preziosi, consultaUtilizzo della generazione automatizzata di risorse basata sull'elaborazione del linguaggio naturale (NLP) del tipo di FHIR DocumentReference risorsa in AWS HealthLake.

Example Aggiornamento di un tipo di **`Patient`** risorsa utilizzando una **`PUT`** richiesta

Quando effettui una PUT richiesta, avrai bisogno dell'endpoint del data store, del nome del tipo di risorsa che desideri aggiornare, un identificatore e un corpo della JSON richiesta.

Se si utilizza PUT per creare una nuova risorsa, utilizza l'identificatore fornito per creare la nuova risorsa.

PUT Request

Esempio di struttura di una PUT richiesta valida:

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

JSON Request Body

Un JSON corpo di esempio utilizzato per aggiornare la risorsa paziente specificata.

```
{
  "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"
}
```

Risposta JSON

Riceverai quanto segue JSON in risposta per confermare la modifica:

```
{
 "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"
 }
}
```

Aggiornamento condizionale

L'aggiornamento condizionale consente di aggiornare una risorsa esistente in base ad alcuni criteri di ricerca di identificazione, anziché in base all'ID logico. Quando il server elabora questo aggiornamento, esegue una ricerca utilizzando le sue funzionalità di ricerca standard per il tipo di risorsa, con l'obiettivo di risolvere un singolo ID logico per questa richiesta.

L'azione intrapresa dipende dal numero di corrispondenze trovate:

- Nessuna corrispondenza, nessun id fornito nel corpo della richiesta: il server crea la risorsa.
- Nessuna corrispondenza, ID fornito e la risorsa non esiste già con l'id: il server considera l'interazione come un'interazione Update as Create.
- Nessuna corrispondenza, id fornito ed esiste già: il server rifiuta l'aggiornamento con un `409 Conflict` errore.
- Una corrispondenza, nessun ID di risorsa fornito OPPURE (id della risorsa fornito e corrisponde alla risorsa trovata): il server esegue l'aggiornamento rispetto alla risorsa corrispondente come sopra dove, se la risorsa è stata aggiornata, il server SHALL restituisce un`200 OK`;

- One Match, id della risorsa fornito ma non corrisponde alla risorsa trovata: il server restituisce un `409 Conflict` errore che indica che la specifica dell'ID del client era un problema, preferibilmente con un OperationOutcome
- Corrispondenze multiple: il server restituisce un `412 Precondition Failed` errore che indica che i criteri del client non erano sufficientemente selettivi, preferibilmente con un OperationOutcome

Example — Aggiorna una risorsa per il paziente il cui nome è Peter, la data di nascita è il 1° gennaio 2000 e il numero di telefono 1234567890:

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

Eliminazione di una risorsa utilizzando **DELETE**

Per eliminare una risorsa dal tuo archivio HealthLake dati, devi fare una `DELETE` HTTP richiesta.

Example Eliminazione di un tipo di **`Patient`** risorsa specifico utilizzando una **`DELETE`** richiesta.

Per creare una `DELETE` richiesta, utilizza l'endpoint del data store. Per trovare l'endpoint di un data store, cerca nella HealthLake console sotto Data Stores o usando l'escribeFHIRDatastoreoperazione D disponibile nel AWS HealthLake API Reference.

È inoltre necessario includere il tipo di risorsa e un identificatore valido.

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

Risposta HTTP

In caso di successo, riceverai un codice di `204` HTTP stato che conferma che la risorsa non è più presente nell'archivio dati. Quando una richiesta di eliminazione ha esito negativo, riceverai un codice di HTTP stato della serie 400 che indica il motivo per cui la `DELETE` richiesta non è riuscita.

Gestione di più FHIR risorse tramite Bundle

Nella specifica HL7 FHIR R4, i bundle sono semplicemente una raccolta di risorse. HealthLake supporta la creazione di un tipo di risorsa Bundle in una FHIR REST API richiesta e l'utilizzo di una transazione bundle per eseguire più CRUD operazioni in una singola richiesta. FHIR REST API In

una transazione di bundle, è necessario specificare il tipo di pacchetto come `batch` nella richiesta. FHIR REST API

Tutte le richieste di pacchetti vengono registrate da. AWS CloudTrail Per ulteriori informazioni sull'utilizzo CloudTrail con HealthLake, consultaChiamate AWS HealthLake API di registrazione con AWS CloudTrail.

HL7FHIRRisorse R4 (esterne)

- Per leggere le specifiche complete, vedere Tipo di risorsa: pacchetto nell'indice della FHIRdocumentazione.
- Per ulteriori informazioni sulle interazioni in batch utilizzando il FHIR RESTAPI, vedere Interazioni in batch utilizzando il FHIR REST API nell'Indice della FHIR documentazione.

Le sezioni seguenti descrivono come strutturare una FHIR REST API richiesta per creare una nuova risorsa Bundle o per elaborare le risorse singolarmente utilizzando transazioni in bundle.

> ⚠ Differenze tra la HealthLake console AWS CLI, la e AWS SDKs
>
> La HealthLake console supporta solo le operazioni di tipo Bundle in cui il tipo di risorsa Bundle è specificato nella FHIR REST API richiesta. URL

Esecuzione di più CRUD operazioni utilizzando i bundle FHIR

Quando nella richiesta non è specificato alcun tipo di risorsaURL, la FHIR REST API richiesta viene analizzata come singole transazioni del Data Store. Ogni CRUD operazione fornita nel JSON corpo viene valutata e viene restituito un codice di HTTP stato specifico. HealthLake supporta il tipo Bundle. `batch`

Per eseguire più CRUD operazioni in una singola FHIR REST API richiesta, procedi come segue:

L'elenco seguente mostra le parti troncate del corpo della richiesta utilizzate nella richiesta di pacchetto. FHIR REST API Per un corpo completo della richiesta, vedere Creazione di una richiesta di pacchetto che coinvolge più operazioni. CRUD

1. Non specificate un tipo di risorsa nella POST richiesta:

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

2. Nel corpo della richiesta, specifica il tipo di pacchetto come `"type": "batch"`
3. Nel corpo della richiesta, specifica i dati specifici della risorsa per ogni CRUD interazione a partire dalla chiave. `resource`
4. Ogni CRUD operazione è specificata come segue `request` nel corpo della richiesta:

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

Nella JSON risposta, si ottiene un codice di HTTP stato per ogni CRUD operazione specificata nella richiesta.

HealthLake limita le transazioni Bundle

- Per ulteriori informazioni sui limiti imposti HealthLake ai pacchetti, consulta. AWS HealthLake endpoint e quote

Di seguito è riportato un esempio di operazione Bundle contenente più CRUD operazioni.

Example — Creazione di una richiesta Bundle che include più CRUD operazioni.

Per effettuare una FHIR REST API richiesta che esegua più CRUD operazioni, è necessario effettuare una `POST` richiesta utilizzando l'endpoint del Data Store e fornire un corpo della JSON richiesta.

Puoi trovare l'endpoint del tuo data store nella HealthLake console sotto Data Stores o utilizzando l'escribeFHIRDatastoreoperazione D nella AWS HealthLake API pagina di riferimento.

POST Request

Effettua una `POST` richiesta utilizzando l'endpoint del tuo data store. Usa la scheda successiva, JSONRequest Body per visualizzare gli elementi richiesti del corpo della richiesta.

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

JSON Request Body

Nel corpo della richiesta, è necessario fornire le seguenti coppie chiave:valore insieme a qualsiasi altro FHIR dato specifico della risorsa relativo alle singole richieste. CRUD Il primo esempio mostra un corpo di richiesta JSON troncato che evidenzia gli elementi richiesti. Il secondo esempio mostra un corpo completo della richiesta. JSON

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

Ecco un esempio completo che mostra la creazione di un nuovo `Patient` tipo di `Medication` risorsa.

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

Risposta JSON

Per confermare la creazione delle risorse specificate nella transazione bundle di esempio, viene visualizzato il codice di HTTP stato 201 Created per ogni CRUD operazione inclusa. Quando un'CRUDoperazione fallisce, viene visualizzato HTTP lo stato della serie 400 che indica il motivo per cui la singola richiesta non è riuscita.

```
{
  "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"
      }
    }
```

```
  ]
}
```

Raggruppamento delle risorse come tipo di risorsa Bundle

Per creare un nuovo tipo di risorsa Bundle, è necessario specificare `Bundle` nella FHIR REST API richiesta e fornire un JSON corpo valido contenente le risorse che si desidera raggruppare.

Quando nella richiesta viene specificato BundleURL, il contenuto del corpo della JSON richiesta viene salvato nell'archivio HealthLake dati così com'è. Pertanto, non è possibile CRUD eseguire alcuna operazione sui singoli tipi di risorse. Ai pacchetti di questo tipo viene assegnato un unico nuovo ID di risorsa. Poiché le risorse vengono salvate così come sono, non è possibile effettuare `GET` o `POST` richiedere informazioni su singole risorse salvate nel tipo di risorsa Bundle.

> **Note**
>
> La specifica HL7 FHIR R4 supporta anche il raggruppamento di risorse utilizzando Group, Composition e List. Quando si creano questi tipi di risorse, le singole risorse non vengono contenute direttamente. Utilizzano invece l'`Reference`elemento per indicare le singole risorse. L'utilizzo di questi tipi di risorse consente quindi di modificare le singole risorse in essi contenute.

Per creare un tipo di `Bundle` risorsa, è necessario specificarlo nella `POST` richiesta e fornire un'JSONenumerazione delle risorse che si desidera includere.

Example — Creazione di una risorsa Bundle utilizzando una richiesta **`POST`**

Per creare una `bundle` risorsa, procedi come segue

1. Formatta una FHIR REST API richiesta come segue:

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

2. Fornisci un JSON corpo che specifica le risorse che desideri raggruppare. Questo esempio raggruppa due risorse per i pazienti.

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

Ricerca nell'archivio HealthLake dati utilizzando le FHIR REST API operazioni

HealthLake supporta la ricerca nell'archivio dati utilizzando le REST API operazioni fornite come parte dello FHIR standard. In questa sezione, troverai esempi di come effettuare `GET` `POST` richieste su diversi tipi di risorse.

> **Note**
>
> Per le domande che riguardano informazioni di identificazione personale (PII) o Informazioni sanitarie protette (PHI), si consiglia di utilizzare `POST` le richieste. In una `POST` richiesta, PII o PHI viene aggiunto come parte del corpo della richiesta e viene crittografato durante il transito.

La FHIR specifica supporta più tipi di parametri di ricerca, ma HealthLake supporta solo un sottoinsieme. Per ulteriori informazioni, consulta Tipi di parametri di ricerca supportati e Parametri di ricerca avanzati supportati da HealthLake.

Ricerca nell'archivio dati utilizzando le FHIR REST API operazioni.

- Tipi di parametri di ricerca supportati
- Parametri di ricerca avanzati supportati da HealthLake
  - _include
  - _revinclude
  - _summary

- _elements
- _total
- _sort
- _count
- Chaining and Reverse Chaining(_has)

- Modificatori di ricerca supportati
- Comparatori di ricerca supportati
- Parametri di ricerca non supportati da HealthLake
- Cerca con POST esempi
- Cerca con GET esempi

Tipi di parametri di ricerca supportati

La tabella seguente mostra i tipi di parametri di ricerca supportati in HealthLake.

Tipi di parametri di ricerca supportati

Parametro di ricerca	Descrizione
_id	ID della risorsa (non completoURL)
_lastUpdated	Data ultimo aggiornamento. Il server ha discrezione sulla precisione dei limiti.
_etichetta	Cerca in base a un tag di risorsa.
_profilo	Cerca tutte le risorse contrassegnate con un profilo.
_sicurezza	Cerca sulle etichette di sicurezza applicate a questa risorsa.
_fonte	Cerca da dove proviene la risorsa.
_testo	Cerca nella narrazione della risorsa.
createdAt	Cerca su estensione personalizzata -createdAt.

> **Note**
>
> I seguenti parametri di ricerca sono supportati solo per i datastore creati dopo il 09 dicembre 2023: _security, _source, _text,. createdAt

La tabella seguente mostra esempi di come modificare le stringhe di query in base a tipi di dati specificati per un determinato tipo di risorsa. Per maggiore chiarezza, i caratteri speciali nella colonna degli esempi non sono stati codificati. Per eseguire correttamente una query, assicuratevi che la stringa di query sia stata codificata correttamente.

Tipi di parametri di ricerca	Informazioni	Esempi

Numero	Cerca un valore numerico in una risorsa specificata. Si osservano cifre significative.
	
	Il numero di cifre significative è specifico in base al valore del parametro di ricerca, esclusi gli zeri iniziali.
	
	I prefissi di confronto sono consentiti.
		`[parameter]=100`
		
		`[parameter]=1e2`
		
		`[parameter]=lt100`

Data/ DateTime	Cerca una data o un'ora specifica. Il formato previsto è `yyyy-mm-ddThh:mm:ss[Z|(+|-)hh:mm]` ma può variare.
	
	Accetta i seguenti tipi di dati: `datedateTime`,`instant`,`PeriodTiming`. Per maggiori dettagli sull'utilizzo di questi tipi di dati nelle ricerche, consulta la data nell'indice della FHIR documentazione.
		`[parameter]=eq2013-01-14`
		
		`[parameter]=gt2013-01-14T10:00`
		
		`[parameter]=ne2013-01-14`

Tipi di parametri di ricerca	Informazioni	Esempi
	I prefissi di confronto sono consentiti.	
Stringa	Cerca una sequenza di caratteri con distinzione tra maiuscole e minuscole. Supporta entrambi i tipiHumanName . Address Per ulteriori dettagli, consultate la voce relativa HumanName ai tipi di Address dati e le voci relative ai tipi di dati nell'Indice della FHIR documentazione. La ricerca avanzata è supportata tramite :text modificatori.	[base]/Patient?given=eve [base]/Patient?given:contains=eve
Token	Cerca una close-to-exact corrispondenza in base a una stringa di caratteri, spesso confrontata con un paio di valori di codice medico. La distinzione tra maiuscole e minuscole è collegata al sistema di codice utilizzato durante la creazione di una query.Le query basate su Subsumption possono aiutare a ridurre i problemi legati alla distinzione tra maiuscole e minuscole. Per chiarezza non è stato codificato. |	[parameter]=[system]|[code] : Qui [system] si riferisce a un sistema di codifica e si [code] riferisce al valore di codice trovato all'interno di quel sistema specifico. [parameter]=[code] : Qui il tuo input corrisponderà a un codice o a un sistema. [parameter]=|[code] : Qui il tuo input corrisponderà a un codice e la proprietà del sistema non ha un identificatore.

Tipi di parametri di ricerca	Informazioni	Esempi

Composita	Cerca più parametri all'interno di un singolo tipo di risorsa, utilizzando i modificatori e l'operazione$. ,
	
	I prefissi di confronto sono consentiti.
		`/Patient?language=FR,NL&language=EN`
		
		`Observation?component-code-value-quantity=http://loinc.org|8480-6$lt60`
		
		`[base]/Group?characteristic-value=gender$mixed`

Quantità	Cerca un numero, un sistema e un codice come valori. È richiesto un numero, ma il sistema e il codice sono facoltativi. In base al tipo di dati sulla quantità. Per maggiori dettagli, consulta la sezione Quantità nell'indice della FHIR documentazione.
	
	Utilizza la seguente sintassi presunta `[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`

Documentazione di riferimento	Cerca riferimenti ad altre risorse.	`[base]/Observation?subject=Patient/23`
		
		`test`

Tipi di parametri di ricerca	Informazioni	Esempi
URI	Cerca una stringa di caratteri che identifichi in modo inequivocabile una particolare risorsa.	`[base]/ValueSet?url=http://acme.org/fhir/ValueSet/123`
Speciale	Ricerche basate su estensioni mediche integrate. NLP	

Parametri di ricerca avanzati supportati da HealthLake

HealthLake supporta i seguenti parametri di ricerca avanzata.

Nome	Descrizione	Esempio	Funzionalità
`_include`	Utilizzato per richiedere la restituzione di risorse aggiuntive in una richiesta di ricerca. Restituisce risorse a cui fa riferimento l'istanza della risorsa di destinazione.	`Encounter?_include=Encounter:subject`	
`_revinclude`	Utilizzato per richiedere la restituzione di risorse aggiuntive in una richiesta di ricerca. Restituisce risorse che fanno riferimento all'istanza della risorsa principale.	`Patient?_id=` **`patient-identifier`** `&_revinclude=Encounter:patient`	
`_summary`	Il riepilogo può essere utilizzato per richiedere un sottoinsieme della risorsa.	`Patient?_summary=text`	Sono supportati i seguenti parametri di riepilogo:`_summary=true` ,`_summary=false` ,`_summary=text` ,`_summary=data` .

Nome	Descrizione	Esempio	Funzionalità
`_elemen`	Richiedi la restituzione di un set specifico di elementi come parte di una risorsa nei risultati della ricerca.	`Patient?_elements=identifier,active,link`	
`_total`	Restituisce il numero di risorse che corrispondono ai parametri di ricerca.	`Patient?_total=accurate`	Support`_total=accurate` ,`_total=none` .
`_sort`	Indica l'ordinamento dei risultati di ricerca restituiti utilizzando un elenco separato da virgole. Il `-` prefisso può essere utilizzato per qualsiasi regola di ordinamento nell'elenco separato da virgole per indicare l'ordine decrescente.	`Observation?_sort=status,-date`	Supporta l'ordinamento per campi con tipi`Number, String, Quantity, Token, URI, Reference` . L'ordinamento per `Date` è supportato solo per i datastore creati dopo il 09 dicembre 2023. Supporta fino a 5 regole di ordinamento.
`_count`	Controlla quante risorse vengono restituite per pagina del pacchetto di ricerca.	`Patient?_count=100`	La dimensione massima della pagina è 100.
`chainin`	Elementi di ricerca delle risorse referenziate. `.`Indirizza la ricerca concatenata all'elemento all'interno della risorsa referenziata.	`DiagnosticReport?subject:Patient.name=peter`	
`reverse chainin (_has)`	Cerca una risorsa in base agli elementi delle risorse che le fanno riferimento.	`Patient?_has:Observation:patient:code=1234-5`	

`_include`

L'utilizzo `_include` in una query di ricerca consente di restituire anche FHIR risorse specifiche aggiuntive. `_include`Da utilizzare per includere risorse collegate in avanti.

Example — Da utilizzare **`_include`** per trovare i pazienti o il gruppo di pazienti a cui è stata diagnosticata la tosse

È possibile eseguire la ricerca in base al tipo di `Condition` risorsa specificando il codice diagnostico per la tosse e quindi utilizzando `_include` specificare che si desidera che venga restituita anche la `subject` diagnosi. Nel tipo di `Condition` risorsa `subject` si riferisce al tipo di risorsa per il paziente o al tipo di risorsa del gruppo.

Per maggiore chiarezza, i caratteri speciali dell'esempio non sono stati codificati. Per eseguire correttamente una query, assicuratevi che la stringa di query sia stata codificata correttamente.

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

`_revinclude`

L'utilizzo `_revinclude` in una query di ricerca consente di restituire anche FHIR risorse aggiuntive specificate. `_revinclude`Da utilizzare per includere risorse collegate all'indietro.

Example — Da utilizzare **`_revinclude`** per includere tipi di risorse correlate all'incontro e all'osservazione collegate a un paziente specifico

Per effettuare questa ricerca, è necessario innanzitutto definire la persona `Patient` specificando il suo identificatore nel parametro di `_id` ricerca. Quindi dovresti specificare FHIR risorse aggiuntive utilizzando la struttura `Encounter:patient` e. `Observation:patient`

Per maggiore chiarezza, i caratteri speciali dell'esempio non sono stati codificati. Per eseguire correttamente una query, assicuratevi che la stringa di query sia stata codificata correttamente.

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

`_summary`

L'utilizzo `_summary` in una query di ricerca consente all'utente di richiedere un sottoinsieme della FHIR risorsa. Può contenere uno dei seguenti valori:`true, text, data, false`. Qualsiasi altro valore verrà considerato non valido. Le risorse restituite verranno contrassegnate con `'SUBSETTED'` meta.tag, per indicare che le risorse sono incomplete.

- `true`: Restituisce tutti gli elementi supportati contrassegnati come «riepilogo» nella definizione di base delle risorse.
- `text`: restituisce solo gli elementi 'text', 'id', 'meta' e solo gli elementi obbligatori di primo livello.
- `data`: restituisce tutte le parti tranne l'elemento 'text'.
- `false`: restituisce tutte le parti delle risorse

In una singola richiesta di ricerca, `_summary=text` non può essere combinato con i `_include` nostri parametri `_revinclude` di ricerca.

Example — Ottieni l'elemento «testuale» delle risorse per i pazienti in un datastore.

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

`_elements`

L'utilizzo `_elements` in una query di ricerca consente di richiedere elementi di FHIR risorsa specifici. Le risorse restituite verranno contrassegnate con `'SUBSETTED'` meta.tag, per indicare che le risorse sono incomplete.

Il `_elements` parametro è costituito da un elenco separato da virgole di nomi di elementi di base, come gli elementi definiti al livello principale della risorsa. Devono essere restituiti solo gli elementi elencati. Se i valori dei `_elements` parametri contengono elementi non validi, il server li ignorerà e restituirà elementi obbligatori e elementi validi.

`_elements`non sarà applicabile alle risorse incluse (risorse restituite la cui modalità di ricerca è`include`).

In una singola richiesta di ricerca, `_elements` non può essere combinato con i parametri `_summary` di ricerca.

Example — Ottieni elementi «identificativi», «attivi» e «link» delle risorse per i pazienti nel tuo HealthLake datastore.

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

_total

L'utilizzo `_total` in una query di ricerca restituirà il numero di risorse che corrispondono ai parametri di ricerca richiesti. HealthLake restituirà il numero totale di risorse corrispondenti (risorse restituite la cui modalità di ricerca è`match`) nella risposta `Bundle.total` di ricerca.

`_total`supporta i valori dei none parametri`accurate`,. `_total=estimate`non è supportato. Tutti gli altri valori verranno considerati non validi. `_total`non è applicabile alle risorse incluse (risorse restituite la cui modalità di ricerca è`include`).

Example — Ottieni il numero totale di risorse per i pazienti in un datastore:

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

_sort

L'utilizzo `_sort` nella query di ricerca ordina i risultati in un ordine specifico. I risultati vengono ordinati in base all'elenco di regole di ordinamento separate da virgole in ordine di priorità. Le regole di ordinamento devono essere parametri di ricerca validi. Tutti gli altri valori verranno considerati non validi.

In una singola richiesta di ricerca, puoi utilizzare fino a 5 parametri di ricerca di ordinamento. Facoltativamente, puoi utilizzare un `-` prefisso per indicare l'ordine decrescente. Per impostazione predefinita, il server ordinerà in ordine crescente.

I tipi di parametri di ricerca di ordinamento supportati sono:`Number, String, Date, Quantity, Token, URI, Reference`. Se un parametro di ricerca si riferisce a un elemento annidato, questo parametro di ricerca non è supportato per l'ordinamento. Ad esempio, la ricerca in base al «nome» del tipo di risorsa Patient si riferisce a Patient.name. L'elemento con tipo di HumanName dati è considerato annidato. Pertanto, l'ordinamento delle risorse del paziente per «nome» non è supportato.

Example — Ottieni le risorse per i pazienti in un datastore e ordinale per data di nascita in ordine crescente:

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

_count

Il parametro `_count` è definito come un'istruzione al server relativa a quante risorse devono essere restituite in una singola pagina.

La dimensione massima della pagina è 100. Qualsiasi valore superiore a 100 non è valido. `_count=0`non è supportato.

Example — Cerca la risorsa Patient e imposta la dimensione della pagina di ricerca su 25:

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

Chaining and Reverse Chaining(_has)

Il concatenamento e il concatenamento inverso FHIR offrono un modo più efficiente e compatto per ottenere dati interconnessi, riducendo la necessità di più interrogazioni separate e rendendo il recupero dei dati più comodo per sviluppatori e utenti.

Se un livello di ricorsione restituisce più di 100 risultati, HealthLake restituirà 4xx per proteggere il datastore dal sovraccarico e dalla causa di più impaginazioni.

Example — Concatenamento: ottiene tutto ciò che si riferisce a un paziente il DiagnosticReport cui nome del paziente è peter.

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

Example — Reverse Chaining - Get Patient Resources, dove la risorsa relativa al paziente viene indicata da almeno un'osservazione, dove l'osservazione ha il codice 1234 e dove l'osservazione si riferisce alla risorsa del paziente nel parametro di ricerca del paziente.

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

Modificatori di ricerca supportati

I modificatori di ricerca vengono utilizzati con campi basati su stringhe. Tutti i modificatori di ricerca utilizzano una logica basata su booleani. HealthLake Ad esempio, è possibile specificare di `:contains` specificare che il campo stringa più grande deve includere una stringa piccola per poterla includere nei risultati della ricerca.

Modificatori di ricerca supportati

Modificatore di ricerca	Tipo
: mancante	Tutti i parametri tranne `Composite`
: esatto	Stringa
:contiene	Stringa
:non	Token
:testo	Token
:identificatore	Documentazione di riferimento

Comparatori di ricerca supportati

È possibile utilizzare i comparatori di ricerca per controllare la natura della corrispondenza in una ricerca. È possibile utilizzare i comparatori durante la ricerca nei campi relativi a numeri, date e quantità. La tabella seguente elenca i comparatori di ricerca e le relative definizioni supportati da. HealthLake

Comparatori di ricerca supportati

Comparatore di ricerca	Descrizione
eq	Il valore del parametro nella risorsa è uguale al valore fornito.
uno	Il valore del parametro nella risorsa non è uguale al valore fornito.

Comparatore di ricerca	Descrizione
gt	Il valore del parametro nella risorsa è maggiore del valore fornito.
lt	Il valore del parametro nella risorsa è inferiore al valore fornito.
età	Il valore del parametro nella risorsa è maggiore o uguale al valore fornito.
le	Il valore del parametro nella risorsa è inferiore o uguale al valore fornito.
come	Il valore del parametro nella risorsa inizia dopo il valore fornito.
eb	Il valore del parametro nella risorsa termina prima del valore fornito.

Parametri di ricerca non supportati da HealthLake

Per un elenco completo dei parametri di ricerca supportati, consultate il registro dei parametri FHIR di ricerca. HealthLake supporta tutti i parametri di ricerca ad eccezione di quelli elencati nella tabella.

Parametri di ricerca non supportati

Composizione del pacchetto	Ubicazione: vicino
Identificatore del pacchetto	C onsent-source-reference
Messaggio del pacchetto	Paziente a contratto
Tipo di pacchetto	Contenuto delle risorse
Timestamp del pacchetto	Interrogazione delle risorse

Cerca con POST esempi

È possibile effettuare ricerche in un archivio HealthLake dati effettuando POST richieste. È possibile fornire i parametri di interrogazione nel URI o in un corpo della richiesta, ma non è possibile utilizzarli entrambi in una singola richiesta.

Gli esempi in questo argomento seguono questa procedura consigliata.

> **Note**
>
> Per le domande che riguardano informazioni di identificazione personale (PII) o Informazioni sanitarie protette (PHI), si consiglia di utilizzare `POST` le richieste. In una `POST` richiesta, PII o PHI viene aggiunto come parte del corpo della richiesta e viene crittografato durante il transito.

Quando si effettua una POST richiesta con un parametro nel corpo della richiesta, `Content-Type: application/x-www-form-urlencoded` utilizzalo come parte dell'intestazione.

Questo argomento fornisce esempi di come effettuare ricerche POST utilizzando i seguenti tipi di risorse.

- Età: L'età non è un tipo di risorsa definito inFHIR. L'età viene invece acquisita come parte del tipo di risorsa Patient. Per cercare un gruppo di pazienti in base a un'età o una fascia di età specifiche, usa athe section called “Comparatori di ricerca supportati”. Per maggiori dettagli, vedere Tipo di risorsa: Paziente nell'indice della FHIR documentazione.
- Condizione: questo tipo di risorsa memorizza i dettagli relativi a concetti clinici come diagnosi, situazioni, condizioni cliniche e problemi che hanno raggiunto un livello di preoccupazione. Per ulteriori informazioni, vedere Tipo di risorsa: condizione nell'indice della FHIR documentazione. HealthLake crea nuove condizioni basate sui documenti trovati in DocumentReference. Queste aggiunte sono escluse per impostazione predefinita quando si effettua una `POST` richiesta. Per includerle, è necessario specificare un identificatore valido per una risorsa condizionale nella ricerca.
- DocumentReferencePer includerli, è necessario HealthLake specificare un identificatore valido per una risorsa condizionale nella ricerca. ----sep----:Questo tipo di risorsa è supportato da. Questo tipo di risorsa supporta documenti di riferimento di qualsiasi tipo. Per ulteriori informazioni, consulta Tipo di risorsa: DocumentReference nell'indice della FHIR documentazione. HealthLake fornisce anche l'elaborazione integrata del linguaggio naturale (NLP) dei documenti presenti in

DocumentReference. Per ulteriori informazioni, consulta Utilizzo della generazione automatizzata di risorse basata sull'elaborazione del linguaggio naturale (NLP) del tipo di FHIR DocumentReference risorsa in AWS HealthLake.

- Ubicazione: questo tipo di risorsa include sia sedi accessorie (un luogo utilizzato per l'assistenza sanitaria senza previa designazione o autorizzazione) sia sedi dedicate e nominate formalmente. Per maggiori dettagli, vedere Tipo di risorsa: Ubicazione nell'indice della FHIR documentazione.
- Osservazione: misurazioni e semplici affermazioni fatte su un paziente, un dispositivo o un altro soggetto. HealthLake crea nuove risorse di osservazione sulla base dei documenti presenti nella DocumentReference risorsa. Per ulteriori informazioni su come HealthLake crea nuove risorse, consultaUtilizzo della generazione automatizzata di risorse basata sull'elaborazione del linguaggio naturale (NLP) del tipo di FHIR DocumentReference risorsa in AWS HealthLake. Queste aggiunte sono escluse per impostazione predefinita quando si effettua una `POST` richiesta. Per includerle, è necessario specificare un identificatore valido per una risorsa di osservazione nella ricerca. Per ulteriori informazioni, consulta Tipo di risorsa: osservazione nell'indice della FHIR documentazione.

Ogni scheda mostra esempi di come effettuare ricerche in base al tipo di risorsa specificato. Include un esempio di come specificare la richiesta nel corpo della richiesta.

Age

Utilizzare quanto segue per effettuare una richiesta di ricerca `POST` basata sul tipo di `Patient` risorsa. Questa ricerca utilizza il `eq` comparatore di ricerca per cercare persone nate nel 1997.

È necessario specificare una richiesta URL e un corpo della richiesta. Ecco un esempio di richiestaURL.

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

Per specificare l'anno 1997 nella ricerca, è necessario aggiungere il seguente elemento al corpo della richiesta.

```
birthdate=eq1997
```

Risposta JSON

In caso di successo, riceverai un codice di `200` HTTP risposta e una JSON risposta simile.

Condition

Utilizzando quanto segue per effettuare una `POST` richiesta sul tipo di `Condition` risorsa. Questa ricerca trova le posizioni nel tuo archivio HealthLake dati che contengono il codice medico`72892002`.

È necessario specificare una richiesta URL e un corpo della richiesta. Ecco un esempio di richiestaURL.

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

Per specificare il codice medico che desideri cercare, aggiungi questo JSON elemento al corpo della richiesta.

```
code=72892002
```

Risposta JSON

In caso di successo, riceverai un codice di `200` HTTP risposta. La seguente JSON risposta è stata troncata per motivi di chiarezza.

```
{
 "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

Per vedere i risultati dell'elaborazione integrata HealthLake del linguaggio naturale (NLP) quando si effettua una POST richiesta sul tipo di `DocumentReference` risorsa, formattare una richiesta è il seguente.

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

Per specificare l' DocumentReference elemento a cui vuoi fare riferimento, vediParametri di ricerca. Li specificherai nel corpo della richiesta comeJSON.

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

Questa stringa di query utilizza più parametri di ricerca per effettuare ricerche nelle operazioni di Amazon Comprehend API Medical utilizzate per generare i risultati NLP medici integrati.

Location

Usa quanto segue per effettuare una `POST` richiesta sul tipo di `Location` risorsa. Questa ricerca trova le posizioni nel tuo archivio HealthLake dati che contengono il nome della città Boston come parte dell'indirizzo.

È necessario specificare una richiesta URL e un corpo della richiesta. Ecco un esempio di richiestaURL.

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

Per specificare Boston nella ricerca, aggiungi il seguente elemento al corpo della richiesta:

```
address=Boston
```

Risposta JSON

In caso di successo, riceverai un codice di `200` HTTP risposta. La JSON risposta è stata troncata per motivi di chiarezza.

```
{
 "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

Utilizzare quanto segue per effettuare una richiesta di ricerca POST basata sul tipo di risorsa. `Observation` Questa ricerca utilizza il parametro `value-concept` di ricerca per cercare il codice medico,`266919005`. Questo stato indica`Never smoker`.

È necessario specificare una richiesta URL e un corpo della richiesta. Ecco un esempio di richiestaURL.

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

Per specificare lo stato`Never smoker`, impostate `value-concept=266919005` nel corpo diJSON.

```
value-concept=266919005
```

Risposta JSON

In caso di successo, riceverai un codice di `200` HTTP risposta. La seguente JSON risposta è stata troncata per motivi di chiarezza.

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

Cerca con GET esempi

È possibile effettuare ricerche in un archivio HealthLake dati effettuando GET richieste. HealthLake supporta solo la fornitura di parametri di query come parte del URI corpo della richiesta e non come parte di esso.

> **Note**
>
> Per le domande che riguardano informazioni di identificazione personale (PII) o Informazioni sanitarie protette (PHI), si consiglia di utilizzare `POST` le richieste. In una `POST` richiesta, PII o PHI viene aggiunto come parte del corpo della richiesta e viene crittografato durante il transito.

L'argomento fornisce esempi di come eseguire ricerche `GET` utilizzando i tipi di risorse supportati in HealthLake.

- Età: L'età non è un tipo di risorsa definito inFHIR. L'età viene invece rilevata come parte del tipo di risorsa per il paziente. Per cercare un gruppo di pazienti in base a un'età o una fascia di età specifiche, è necessario utilizzare unthe section called “Comparatori di ricerca supportati”. Per maggiori dettagli, vedere Tipo di risorsa: Paziente nell'indice della FHIR documentazione.
- Condizione: questo tipo di risorsa memorizza i dettagli relativi a concetti clinici come diagnosi, situazioni, condizioni cliniche e problemi che hanno raggiunto un livello di preoccupazione. Per ulteriori informazioni, vedere Tipo di risorsa: condizione nell'indice della FHIR documentazione. HealthLake crea nuove condizioni basate sui documenti trovati in DocumentReference. Queste aggiunte sono escluse per impostazione predefinita quando si effettua una `POST` richiesta. Per includerle, è necessario specificare un identificatore valido per una risorsa condizionale nella ricerca.
- DocumentReferencePer includerli, è necessario HealthLake specificare un identificatore valido per una risorsa condizionale nella ricerca. ----SEP----:Questo tipo di risorsa è supportato da.

Questo tipo di risorsa supporta documenti di riferimento di qualsiasi tipo. Per ulteriori informazioni, consulta Tipo di risorsa: DocumentReference nell'indice della FHIR documentazione. HealthLake fornisce anche l'elaborazione integrata del linguaggio naturale (NLP) dei documenti presenti in DocumentReference. Per ulteriori informazioni, consulta Utilizzo della generazione automatizzata di risorse basata sull'elaborazione del linguaggio naturale (NLP) del tipo di FHIR DocumentReference risorsa in AWS HealthLake.

- Ubicazione: questo tipo di risorsa include sia sedi accessorie (un luogo utilizzato per l'assistenza sanitaria senza previa designazione o autorizzazione) sia sedi dedicate e nominate formalmente. Per maggiori dettagli, vedere Tipo di risorsa: Ubicazione nell'indice della FHIR documentazione.
- Osservazione: misurazioni e semplici affermazioni fatte su un paziente, un dispositivo o un altro soggetto. HealthLake crea nuove risorse di osservazione sulla base dei documenti presenti nella DocumentReference risorsa. Per ulteriori informazioni su come HealthLake creare nuove risorse, consultaUtilizzo della generazione automatizzata di risorse basata sull'elaborazione del linguaggio naturale (NLP) del tipo di FHIR DocumentReference risorsa in AWS HealthLake. Queste aggiunte sono escluse per impostazione predefinita quando si effettua una `POST` richiesta. Per includerle, è necessario specificare un identificatore valido per una risorsa di osservazione nella ricerca. Per ulteriori informazioni, consulta Tipo di risorsa: osservazione nell'indice della FHIR documentazione.

Ogni scheda mostra un esempio di come effettuare una ricerca in base al tipo di risorsa specificato. Include un esempio di come specificare la richiesta nella URI e la relativa JSON risposta.

Age

Utilizzare quanto segue per effettuare una richiesta di ricerca `GET` basata sul tipo di `Patient` risorsa. Questa ricerca utilizza il `eq` comparatore di ricerca per cercare persone nate nel 1997.

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

Risposta JSON

In caso di successo, riceverai un codice di `200` HTTP risposta.

Condition

Usa quanto segue per fare una `GET` richiesta sul tipo di `Condition` risorsa. Questa ricerca trova le posizioni nel tuo archivio HealthLake dati che contengono il codice medico`72892002`.

È necessario specificare una richiesta URL e un corpo della richiesta. Ecco un esempio di richiestaURL.

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

Risposta JSON

In caso di successo, riceverai un codice di 200 HTTP risposta. La seguente JSON risposta è stata troncata per motivi di chiarezza.

```
{
 "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

Questo esempio mostra come creare una richiesta di ricerca sul tipo di `DocumentReference` risorsa per i pazienti con diagnosi di streptococco e ai quali è stata prescritta anche amoxicillina.

```
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
```

In caso di successo si otterrà la seguente risposta. JSON

```
{
    "resourceType": "Bundle",
    "type": "searchset",
    "entry": [
    {
       "resource": {
       "resourceType": "DocumentReference",
       "id": "985c3e94-4219-4c79-97a1-c94694525e24",
       "meta": {
          "lastUpdated": "2020-11-23T06:09:10.719Z"
        },
        "extension": [
          {
            "url": "http://healthlake.amazonaws.com/aws-cm/",
            "extension": [
              {
                "url": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/",
                "extension": [
                  {
```

```
                    "url": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/raw-
response",
                    "valueString": "{Entities: [{Id: 0,Text: otitis media,Category:
 MEDICAL_CONDITION,Type: DX_NAME,Score: 0.9815994,BeginOffset: 151,EndOffset:
 163,Attributes: [],Traits: [{Name: DIAGNOSIS,Score: 0.95042425}],ICD10CMConcepts:
 [{Description: Otitis media, unspecified, unspecified ear,Code: H66.90,Score:
 0.7176407}, {Description: Otitis media, unspecified,Code: H66.9,Score:
 0.6930445}, {Description: Otitis media, unspecified, left ear,Code: H66.92,Score:
 0.688161}, {Description: Otitis media, unspecified, bilateral,Code: H66.93,Score:
 0.6748094}, {Description: Otitis media, unspecified, right ear,Code:
 H66.91,Score: 0.6645618}]}, {Id: 1,Text: streptococcal sore throat,Category:
 MEDICAL_CONDITION,Type: DX_NAME,Score: 0.92208487,BeginOffset: 461,EndOffset:
 486,Attributes: [],Traits: [],ICD10CMConcepts: [{Description: Streptococcal
 pharyngitis,Code: J02.0,Score: 0.55638546}, {Description: Acute streptococcal
 tonsillitis, unspecified,Code: J03.00,Score: 0.53159785}, {Description:
 Streptococcal sepsis, unspecified,Code: A40.9,Score: 0.51865804}, {Description:
 Acute pharyngitis, unspecified,Code: J02.9,Score: 0.45085955}, {Description:
 Streptococcal infection, unspecified site,Code: A49.1,Score: 0.41550553}]},
 {Id: 3,Text: disorder,Category: MEDICAL_CONDITION,Type: DX_NAME,Score:
 0.9191257,BeginOffset: 488,EndOffset: 496,Attributes: [],Traits: [{Name:
 DIAGNOSIS,Score: 0.93372077}],ICD10CMConcepts: [{Description: Parkinson's
 disease,Code: G20,Score: 0.6959145}, {Description: Illness, unspecified,Code:
 R69,Score: 0.68428487}, {Description: Disorder of bone, unspecified,Code:
 M89.9,Score: 0.6542605}, {Description: Unspecified mental disorder due to known
 physiological condition,Code: F09,Score: 0.6240179}, {Description: Mental disorder,
 not otherwise specified,Code: F99,Score: 0.61046}]}],ModelVersion: 0.1.0}"
                  },
                  {
                    "url": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/
model-version",
                    "valueString": "0.1.0"
                  },
                  {
                    "url": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/aws-
cm-icd10-entity",
                    "extension": [
                      {
                        "url": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/
aws-cm-icd10-entity-id",
                        "valueInteger": 0
                      },
                      {
                        "url": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/
aws-cm-icd10-entity-text",
```

```
                        "valueString": "otitis media"
                      },
                      {
                        "url": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/
aws-cm-icd10-entity-begin-offset",
                        "valueInteger": 151
                      },
                      {
                        "url": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/
aws-cm-icd10-entity-end-offset",
                        "valueInteger": 163
                      },
                      {
                        "url": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/
aws-cm-icd10-entity-score",
                        "valueDecimal": 0.9815994
                      },
                      {
                        "url": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/
aws-cm-icd10-entity-ConceptList",
                        "extension": [
                          {
                            "url": "http://healthlake.amazonaws.com/aws-cm/infer-
icd10/aws-cm-icd10-entity-Concept",
                            "extension": [
                              {
                                "url": "http://healthlake.amazonaws.com/aws-cm/
infer-icd10/aws-cm-icd10-entity-Concept-Code",
                                "valueString": "H66.90"
                              },
                              {
                                "url": "http://healthlake.amazonaws.com/aws-cm/
infer-icd10/aws-cm-icd10-entity-Concept-Description",
                                "valueString": "Otitis media, unspecified,
 unspecified ear"
                              },
                              {
                                "url": "http://healthlake.amazonaws.com/aws-cm/
infer-icd10/aws-cm-icd10-entity-Concept-Score",
                                "valueDecimal": 0.7176407
                              }
                            ]
                          },
                          {
```

```
                            "url": "http://healthlake.amazonaws.com/aws-cm/infer-
icd10/aws-cm-icd10-entity-Concept",
                            "extension": [
                              {
                                "url": "http://healthlake.amazonaws.com/aws-cm/
infer-icd10/aws-cm-icd10-entity-Concept-Code",
                                "valueString": "H66.9"
                              },
                              {
                                "url": "http://healthlake.amazonaws.com/aws-cm/
infer-icd10/aws-cm-icd10-entity-Concept-Description",
                                "valueString": "Otitis media, unspecified"
                              },
                              {
                                "url": "http://healthlake.amazonaws.com/aws-cm/
infer-icd10/aws-cm-icd10-entity-Concept-Score",
                                "valueDecimal": 0.6930445
                              }
                            ]
                          },
                          {
                            "url": "http://healthlake.amazonaws.com/aws-cm/infer-
icd10/aws-cm-icd10-entity-Concept",
                            "extension": [
                              {
                                "url": "http://healthlake.amazonaws.com/aws-cm/
infer-icd10/aws-cm-icd10-entity-Concept-Code",
                                "valueString": "H66.92"
                              }
                            ]
                          }
```

Location

Usa quanto segue per fare una `GET` richiesta sul tipo di `Location` risorsa. Questa ricerca trova le posizioni nel tuo archivio HealthLake dati che contengono il nome della città Boston come parte dell'indirizzo.

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

Risposta JSON

In caso di successo, riceverai un codice di 200 HTTP risposta. La JSON risposta è stata troncata per motivi di chiarezza.


```
{
    "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

Usa quanto segue per effettuare una richiesta di ricerca GET basata sul tipo di risorsa. `Observation` Questa ricerca utilizza il parametro `value-concept` di ricerca per cercare il codice medico,`266919005`. Questo stato indica`Never smoker`.

È necessario specificare una richiesta URL e una stringa di query. Ecco un esempio di richiestaURL.

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

Per specificare lo stato`Never smoker`, imposta `value-concept=266919005` come stringa di query.

Risposta JSON

In caso di successo, riceverai un codice di `200` HTTP risposta. La seguente JSON risposta è stata troncata per motivi di chiarezza.

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

Leggere FHIR la cronologia delle risorse

L'FHIR`history`interazione recupera la cronologia di una particolare FHIR risorsa in un archivio HealthLake dati. Utilizzando questa interazione, è possibile determinare in che modo il contenuto di

una FHIR risorsa è cambiato nel tempo. È anche utile, in coordinamento con i registri di controllo, per vedere lo stato di una risorsa prima e dopo la modifica.

> **Note**
>
> FHIRla risorsa `history` è abilitata per impostazione predefinita in tutti gli archivi HealthLake dati creati dopo il 25/10/2024. Se il tuo data store è stato creato prima di questa data, puoi inviare un ticket di supporto per abilitare l'interazione. FHIR `history` Crea un caso utilizzando AWS Support Center Console. Per creare il tuo caso, accedi al tuo Account AWS e scegli Crea caso.

L'`history`interazione viene eseguita utilizzando il `HTTP GET` comando. Le FHIR interazioni `create` danno come `delete` risultato una versione storica della risorsa da salvare. `update` HealthLake supporta i seguenti parametri di ricerca per l'FHIR`history`interazione.

HealthLake parametri di ricerca supportati per FHIR **`history`** l'interazione

parametro di ricerca	Descrizione
`_count : integer`	Il numero massimo di risultati di ricerca in una pagina. Il server restituirà il numero richiesto o il numero massimo di risultati di ricerca consentit o per impostazione predefinita per l'archivio dati, a seconda di quale sia inferiore.
`_since : instant`	Include solo le versioni delle risorse create in un determinato istante o successivamente.
`_at : date(Time)`	Includi solo le versioni delle risorse che erano correnti in un determinato momento durante il periodo di tempo specificato nel valore di data e ora. Per ulteriori informazioni, `date`consulta la HL7FHIRRESTfulAPIdocumentazione.

L'esempio seguente restituisce 100 risultati di ricerca storici per pagina per una FHIR `Patient` risorsa in HealthLake. Per visualizzare l'intero URL percorso, scorri il pulsante Copia. Il URL è del modulo:

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

Il contenuto restituito da un'interazione cronologica è contenuto in una FHIR risorsa `Bundle`, con tipo impostato su`history`. Contiene la cronologia delle versioni specificata, ordinata in base alle ultime versioni meno recenti e include le risorse eliminate. Per ulteriori informazioni sull'`history`interazione, consulta `history`la HL7 FHIR RESTful API documentazione.

> **Note**
>
> Puoi disattivare tipi di `history` FHIR risorse specifici. Per annullare l'iscrizione, crea un caso utilizzando AWS Support Center Console. Per creare il tuo caso, accedi al tuo Account AWS e scegli Crea caso.

Leggere la cronologia delle risorse specifiche della versione FHIR

L'FHIR`vread`interazione esegue una lettura specifica della versione di una risorsa in un data store. HealthLake Utilizzando questa interazione, è possibile visualizzare il contenuto di una FHIR risorsa com'era in un determinato momento del passato.

HealthLake dichiara il supporto per il controllo delle versioni `CapabilityStatement.rest.resource.versioning`per ogni risorsa supportata. Tutti gli archivi HealthLake dati includono `Resource.meta.versionId` (`vid`) su tutte le risorse.

Quando FHIR `history` l'interazione è abilitata (per impostazione predefinita per gli archivi dati creati dopo il 25/10/2024 o su richiesta per gli archivi dati più vecchi), la `Bundle` risposta include `vid` come parte di. `location` Nell'esempio seguente, `vid` viene visualizzato come numero. 1 Per visualizzare l'esempio completo, vedete Example bundle/bundle-response (). JSON

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

L'`vread`interazione viene eseguita utilizzando il comando. `HTTP GET` L'`vread`interazione seguente restituisce una singola istanza con il contenuto specificato per la FHIR `Patient` risorsa per la versione dei metadati della `vid` risorsa specificata da. Per visualizzare l'intero URL percorso nell'esempio seguente, scorri il pulsante Copia. Il URL è del modulo:

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

> **Note**
>
> Se si utilizza l'`history`interazione senza `vread` durante la lettura di una FHIR risorsa, restituisce HealthLake sempre la versione più recente dei metadati della risorsa.

Per ulteriori informazioni sull'`vread`interazione, consulta la `vread`documentazione HL7FHIRRestful. API

Acquisizione dei dati dei pazienti con l'operazione Patient $everything FHIR REST API

L'operazione Patient $everything viene utilizzata per interrogare una risorsa FHIR Patient insieme a qualsiasi altra risorsa correlata a quel paziente. Questa operazione può essere utilizzata per fornire a un paziente l'accesso all'intera cartella clinica o per consentire a un fornitore di eseguire un download di massa di dati relativi a un paziente. HealthLake supporta $everything per un ID paziente specifico.

> **Note**
>
> L'operazione Patient $everything è attualmente supportata negli archivi dati creati dopo il 27 febbraio 2024.

Ottieni tutte le risorse relative a un paziente

Patient $everything è un'RESTAPIoperazione che può essere richiamata come mostrato negli esempi seguenti.

GET Request

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

> **Note**
>
> Le risorse in risposta sono ordinate per tipo di risorsa e ID di risorsa.
> La risposta viene sempre compilata con Bundle.total.

Parametri Patient $everything

HealthLake supporta i seguenti parametri di interrogazione

Parametro	Informazioni
rapida	Recupera tutti i dati del paziente dopo una data di inizio specificata.
end	Ottieni tutti i dati del paziente prima di una data di fine specificata.
since	Aggiorna tutti i dati dei pazienti dopo una data specificata.
_tipo	Ottieni i dati dei pazienti per tipi di risorse specifici.
_conta	Ottieni i dati del paziente e specifica le dimensioni della pagina.

Example - Ottieni tutti i dati del paziente dopo una data di inizio specificata

Patient $everything può utilizzare il `start` filtro per interrogare i dati solo dopo una data specifica.

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

Example - Ottieni tutti i dati del paziente prima di una data di fine specificata

Patient $everything può utilizzare il end filtro solo per interrogare i dati prima di una data specifica.

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

Example - Aggiorna tutti i dati dei pazienti dopo una data specificata

Patient $everything può utilizzare il `since` filtro solo per interrogare i dati aggiornati dopo una data specifica.

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

Example - Ottieni i dati dei pazienti per tipi di risorse specifici

Patient $everything può utilizzare il `_type` filtro per specificare tipi di risorse specifici da includere nella risposta. È possibile specificare più tipi di risorse in un elenco separato da virgole.

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

Example - Ottieni i dati del paziente e specifica le dimensioni della pagina

Patient $everything può utilizzare il `_count` per impostare la dimensione della pagina.

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

Patient $everything **`start`** e attributi **`end`**

HealthLake supporta i seguenti attributi di risorsa per i parametri di inizio e fine della query.

Risorsa	Elemento risorsa
Account	Account. servicePeriod.inizio
AdverseEvent	AdverseEvent.data
AllergyIntolerance	AllergyIntolerance.recordedDate

Risorsa	Elemento risorsa
Appuntamento	Appuntamento. Inizio
AppointmentResponse	AppointmentResponse.inizio
AuditEvent	AuditEvent.periodo.inizio
Base	Di base. Creato
BodyStructure	NO_ DATE
CarePlan	CarePlan.periodo.inizio
CareTeam	CareTeam.periodo.inizio
ChargeItem	ChargeItem. occurrenceDateTime, ChargeItem. occurrencePeriod.inizio, ChargeItem. occurrenceTiming.evento
Richiedi	Reclamo. billablePeriod.iniziare
ClaimResponse	ClaimResponse.creato
ClinicalImpression	ClinicalImpression.data
Communication	Comunicazione. Inviata
CommunicationRequest	CommunicationRequest. occurrenceDateTime, CommunicationRequest. occurrencePeriod.inizio
Composizione	Composizione.data

Risorsa	Elemento risorsa
Condizione	Condizione. recordedDate
Consenso	Consenso. dateTime
Copertura	Copertura.Period.Start
CoverageEligibilityRequest	CoverageEligibilityRequest.creato
CoverageEligibilityResponse	CoverageEligibilityResponse.creato
DetectedIssue	DetectedIssue.identificato
DeviceRequest	DeviceRequest.authoredOn
DeviceUseStatement	DeviceUseStatement.recordedOn
DiagnosticReport	DiagnosticReport.efficace
DocumentManifest	DocumentManifest.creato
DocumentReference	DocumentReference.contesto.periodo.inizio
Incontro	Incontro.period.start
EnrollmentRequest	EnrollmentRequest.creato

Risorsa	Elemento risorsa
EpisodeOfCare	EpisodeOfCare.periodo.inizio
ExplanationOfBenefit	ExplanationOfBenefit. billablePeriod.inizio
FamilyMemberHistory	NO_ DATE
Flag	flag. Period.start
Obiettivo	Obiettivo. statusDate
Group (Gruppo)	NO_ DATE
ImagingStudy	ImagingStudy.iniziato
Immunizzazione	Immunizzazione. Registrata
ImmunizationEvaluation	ImmunizationEvaluation.data
ImmunizationRecommendation	ImmunizationRecommendation.data
Fattura	Data della fattura
Elenco	Elenca. Data
MeasureReport	MeasureReport.periodo.inizio

Risorsa	Elemento risorsa
Media	Media. Emesso
MedicationAdministration	MedicationAdministration.efficace
MedicationDispense	MedicationDispense.whenPrepared
MedicationRequest	MedicationRequest.authoredOn
MedicationStatement	MedicationStatement.dateAsserted
MolecularSequence	NESSUNO_ DATE
NutritionOrder	NutritionOrder.dateTime
Osservazione	Osservazione. Efficace
Paziente	NO_ DATE
Person	NESSUNO_ DATE
Procedura	Procedura. Eseguita
Provenienza	Provenienza occurredPeriod.start, Provenienza. occurredDateTime
QuestionnaireResponse	QuestionnaireResponse.scritto

Risorsa	Elemento risorsa
RelatedPerson	NESSUNO_ DATE
RequestGroup	RequestGroup.authoredOn
ResearchSubject	ResearchSubject.periodo
RiskAssessment	RiskAssessment. occurrenceDateTime, RiskAssessment. occurrencePeriod.inizio
Pianificazione	Pianificazione. planningHorizon
ServiceRequest	ServiceRequest.authoredOn
Esemplare	Esemplare. receivedTime
SupplyDelivery	SupplyDelivery. occurrenceDateTime, SupplyDelivery. occurrencePeriod.inizio, SupplyDelivery. occurrenceTiming.evento
SupplyRequest	SupplyRequest.authoredOn
VisionPrescription	VisionPrescription.dateWritten

Esportazione di dati dal tuo HealthLake data store usando $export

Per effettuare una richiesta di esportazione utilizzando il FHIR REST API comando Specificare `$export` come parte della `POST` richiesta e includere i parametri della richiesta nel corpo della richiesta. In base alle FHIR specifiche, il FHIR server deve supportare `GET` le richieste e può supportare `POST` le richieste. Per supportare parametri aggiuntivi, è necessario un body per avviare l'esportazione, quindi HealthLake supporta POST le richieste.

> **Important**
>
> HealthLake gli archivi di dati creati prima del 1° giugno 2023 supportano solo le richieste di lavori di esportazione FHIR REST API basate sull'esportazione per le esportazioni a livello di sistema.
> HealthLake gli archivi di dati creati prima del 1° giugno 2023 non supportano l'acquisizione dello stato di un'esportazione utilizzando una `GET` richiesta sull'endpoint di un data store.

Tutte le richieste di esportazione effettuate utilizzando il FHIR REST API vengono restituite in `ndjson` formato ed esportate in un bucket Amazon S3. Ogni oggetto S3 conterrà un solo tipo di risorsa. FHIR

Puoi effettuare una singola richiesta di esportazione per ogni AWS account alla volta. Per ulteriori informazioni sui Service Quotas associati a HealthLake, vedere. AWS HealthLake endpoint e quote

Per ulteriori informazioni su come effettuare una richiesta di esportazione utilizzando il FHIR RESTAPI, consulta. Esportazione di dati dal tuo archivio HealthLake dati con operazioni FHIR REST API

Esegui query negli archivi AWS HealthLake dati utilizzando SQL Amazon Athena

Quando HealthLake crei un data store, la struttura FHIR dati altamente annidata viene inserita in Amazon Athena e trasformata automaticamente in tabelle Iceberg con cui è possibile interrogare. SQL La concessione dell'accesso a questa nuova risorsa viene gestita utilizzando AWS Lake Formation. Ogni tipo di FHIR risorsa è rappresentato come una tabella individuale in Athena.

Important

Per gli archivi dati creati prima del 14 novembre 2022, è necessario migrare l'archivio dati esistente in uno nuovo per interrogarlo. SQL Per assistenza, consulta Migrazione di un data store esistente per utilizzare Amazon Athena.

Note

Dopo il 20 febbraio 2023, gli archivi HealthLake dati non utilizzano l'elaborazione integrata del linguaggio naturale (NLP) per impostazione predefinita. Se sei interessato ad attivare questa funzionalità sul tuo data store, consulta il Come posso attivare la funzionalità integrata HealthLake di elaborazione del linguaggio naturale? capitolo Risoluzione dei problemi.

Per creare un HealthLake data store, è necessario aggiungere IAM criteri aggiuntivi e un ruolo di servizio IAM all'utente o al ruolo di HealthLake amministratore. Per ulteriori informazioni sulla configurazione delle autorizzazioni, consultaConfigurazione delle autorizzazioni per iniziare a utilizzare AWS HealthLake.

HealthLake gli archivi di dati vengono inseriti in Athena come tabelle Iceberg. Per saperne di più sul funzionamento delle tabelle Iceberg in Athena, consulta Using Iceberg tables nella Athena User Guide.

HealthLake supporta `READ` le operazioni dei tuoi archivi HealthLake dati, archivi dati in Athena. Per ulteriori informazioni sulle operazioni di creazione, lettura, aggiornamento ed eliminazione (CRUD) che utilizzano le FHIR REST API operazioni, consulta Utilizzo FHIR REST API delle interazioni con un archivio HealthLake dati ulteriori informazioni su come CRUD le operazioni influiscono sui dati in Athena.

Gli argomenti di questo capitolo descrivono come connettere l'archivio HealthLake dati ad Athena, come interrogarlo utilizzando SQL e come collegare i risultati con altri AWS servizi per ulteriori analisi.

Indice

Connessione del tuo data store ad Amazon Athena

> **Important**
>
> Dopo il 14 novembre 2022, i IAM requisiti di accesso HealthLake sono cambiati. Per creare archivi dati e concedere l'accesso ad essi in Athena, devi aggiungere la policy `AWSLakeFormationDataAdmin` gestita al tuo IAM utente, gruppo o ruolo. Puoi utilizzare la `AWSLakeFormationDataAdmin` policy per creare amministratori di data lake e concedere l'accesso ai data store in Athena.

Questo argomento descrive i passaggi necessari per creare un utente, un gruppo o un ruolo Athena e concedere loro l'accesso FHIR alle risorse presenti in HealthLake un archivio dati.

- Concessione a un utente, gruppo o ruolo dell'accesso a un HealthLake data store (AWS Lake Formation Console)
- Configurazione di un account Athena

Concessione a un utente, gruppo o ruolo dell'accesso a un HealthLake data store (AWS Lake Formation Console)

> Persona: amministratore HealthLake
>
> L' HealthLake amministratore è un amministratore del data lake di AWS Lake Formation. Consentono l'accesso agli archivi di HealthLake dati in Lake Formation.

Per ogni data store creato, ci sono due voci visibili nella console di AWS Lake Formation. Una voce è un collegamento a una risorsa. I nomi dei link alle risorse sono sempre visualizzati in corsivo. Ogni collegamento alla risorsa viene visualizzato con il nome e il proprietario della relativa risorsa condivisa collegata. Per tutti gli archivi HealthLake dati, il proprietario della risorsa condivisa è l'account HealthLake di servizio. L'altra voce è l'archivio HealthLake dati nell'account del HealthLake servizio. I passaggi di questa procedura utilizzano l'archivio dati che è il collegamento alla risorsa.

Per ulteriori informazioni sui collegamenti alle risorse, consulta Come funzionano i collegamenti alle risorse in Lake Formation nella AWS Lake Formation Developer Guide.

Affinché un utente, un gruppo o un ruolo possa interrogare i dati in Athena, è necessario concedere l'autorizzazione Descrivi sul database delle risorse. Quindi, è necessario concedere Select e Descrivi sulle tabelle.

STEP1: Per concedere DESCRIBEle autorizzazioni su un database HealthLake Data Store Resource Link

1. Apri la console AWS Lake Formation: https://console.aws.amazon.com/lakeformation/
2. Nella barra di navigazione principale, scegli Database.
3. Nella pagina Database, scegli il pulsante di opzione accanto al nome del data store in corsivo.
4. Scegli Azioni (▾).
5. Scegli Concessione.
6. Nella pagina Concedi le autorizzazioni per i dati, in Principali, scegli IAMutenti o ruoli.
7. In IAMUtenti o ruoli, usa la freccia rivolta verso il basso (▾) o cerca l'IAMutente, il ruolo o il gruppo su cui desideri poter effettuare interrogazioni in Athena.
8. In LF-Tags o nella scheda delle risorse del catalogo, scegliete l'opzione Named data catalog resources.

9. In Database, utilizzate la freccia rivolta verso il basso (▼) per scegliere il database del HealthLake data store a cui desiderate condividere l'accesso.
10. Nella scheda Autorizzazioni Resource link, in Autorizzazioni Resource link, scegli Descrivi.

Quando la concessione ha esito positivo, viene visualizzato il banner Concedi autorizzazione con successo. Per visualizzare l'autorizzazione appena concessa, scegli Autorizzazioni Data lake. Trova l'utente, il gruppo e il ruolo nella tabella. Nella colonna Autorizzazioni, vedrai l'elenco Descrivi.

Ora devi usare Grant on target per concedere Select e Descrivi su tutte le tabelle del database.

STEP2: Concedi l'accesso a tutte le tabelle in un collegamento alle risorse del HealthLake data store

1. Apri la console AWS Lake Formation: https://console.aws.amazon.com/lakeformation/
2. Nella barra di navigazione principale, scegli Database.
3. Nella pagina Database, scegli il pulsante di opzione accanto al nome del data store in corsivo.
4. Scegli Azioni (▼).
5. Scegli Grant on target.
6. Nella pagina Concedi le autorizzazioni per i dati, in Principali, scegli IAMutenti o ruoli.
7. In IAMUtenti o ruoli, usa la freccia rivolta verso il basso (▼) o cerca l'IAMutente, il gruppo o il ruolo su cui desideri poter effettuare interrogazioni in Athena.
8. In LF-Tags o nella scheda delle risorse del catalogo, scegliete l'opzione Named data catalog resources.
9. In Database, utilizzate la freccia rivolta verso il basso (▼) per scegliere il database del HealthLake data store a cui desiderate concedere l'accesso.
10. In Tabelle, scegli Tutte le tabelle per condividere tutte le tabelle con un HealthLake utente.
11. Nella scheda Autorizzazioni della tabella, in Autorizzazioni della tabella, scegli Descrivi e seleziona.
12. Scegli Concessione.

Dopo aver scelto Concedi, viene visualizzato il banner Concedi autorizzazioni di successo. L'utente specificato può ora effettuare interrogazioni su un archivio HealthLake dati in Athena.

Guida introduttiva ad Athena

> **HealthLake utente**
>
> L' HealthLake utente utilizzerà la console Athena o AWS SDKs interrogherà un HealthLake data store condiviso con lui dall' HealthLake amministratore. AWS CLI

Per interrogare un data store utilizzando Athena, è necessario eseguire le tre operazioni seguenti.

- Concedi all'IAMutente o al ruolo l'accesso al HealthLake data store tramite Lake Formation. Per ulteriori informazioni, consulta Concessione a un utente, gruppo o ruolo dell'accesso a un HealthLake data store (AWS Lake Formation Console).
- Crea un gruppo di lavoro per il tuo HealthLake data store.
- Imposta un bucket Amazon S3 per archiviare i risultati delle query.

Per iniziare a usare Athena, aggiungi le AmazonAthenaFullAccesspolicy FullAccess AWS gestite da AmazonS3 al tuo utente, gruppo o ruolo. L'utilizzo di una politica AWS gestita è un ottimo modo per iniziare a utilizzare un nuovo servizio. Ricorda: le policy gestite di AWS potrebbero non concedere autorizzazioni con privilegi minimi per i tuoi casi d'uso specifici perché sono disponibili per l'uso da parte di tutti i clienti AWS. Quando imposti le autorizzazioni con IAM le politiche, concedi solo le autorizzazioni necessarie per eseguire un'attività. Per ulteriori informazioni IAM e sull'applicazione dei privilegi minimi, consulta Applica le autorizzazioni con privilegi minimi nella Guida per l'utente. IAM

> **Important**
>
> Per interrogare un archivio HealthLake dati in Athena, è necessario utilizzare la versione 3 del motore Athena.

I gruppi di lavoro sono risorse e pertanto è possibile utilizzare politiche IAM basate per controllare l'accesso a gruppi di lavoro specifici. Per ulteriori informazioni, consulta Utilizzo dei gruppi di lavoro per controllare l'accesso alle query e i costi nella Guida per l'utente di Athena.

Per ulteriori informazioni sulla configurazione dei gruppi di lavoro, consulta https://docs.aws.amazon.com/athena/latest/ug/workgroups-procedure.html la Guida per l'utente di Athena.

> **Note**
>
> La regione in cui si trova il bucket Amazon S3 e la console Athena devono corrispondere.

Prima di poter eseguire una query, è necessario specificare una posizione del bucket dei risultati delle query in Amazon S3, o utilizzare un gruppo di lavoro che ha specificato un bucket e la cui configurazione sostituisce le impostazioni del client. I file di output vengono salvati automaticamente per ogni query eseguita.

Per ulteriori dettagli sulla specificazione delle posizioni dei risultati delle query nella console Athena, consulta Specificare una posizione dei risultati delle query utilizzando la console Athena nella Amazon Athena User Guide.

Per vedere esempi di come interrogare il tuo archivio HealthLake dati in Athena, consulta. Interroga il tuo archivio HealthLake dati utilizzando SQL

Interroga il tuo archivio HealthLake dati utilizzando SQL

> **Note**
>
> Dopo il 20 febbraio 2023, gli archivi HealthLake dati non utilizzano l'elaborazione integrata del linguaggio naturale (NLP) per impostazione predefinita. Se sei interessato ad attivare questa funzionalità sul tuo data store, consulta il Come posso attivare la funzionalità integrata HealthLake di elaborazione del linguaggio naturale? capitolo Risoluzione dei problemi.
> Tutti gli esempi di questo argomento utilizzano dati fittizi creati con Synthea. Per ulteriori informazioni sulla creazione di un data store precaricato con dati Synthea, consulta. Creazione di un archivio dati in AWS HealthLake

Quando importi il tuo HealthLake data store in Athena, ogni tipo di risorsa dal tuo HealthLake data store viene convertito in una tabella. Queste tabelle possono essere interrogate singolarmente o in gruppo utilizzando SQL interrogazioni basate. A causa della struttura degli archivi dati, i dati vengono importati in Athena come diversi tipi di dati. Per ulteriori informazioni sulla creazione di SQL query in grado di accedere a questi tipi di dati, consulta Interrogare gli array con tipi complessi e strutture annidate nella Amazon Athena User Guide.

Per ogni elemento di un tipo di risorsa, la specifica definisce una cardinalità. FHIR La cardinalità di un elemento definisce i limiti inferiore e superiore di quante volte questo elemento può apparire. Quando si crea un'SQLinterrogazione, è necessario tenerne conto. Ad esempio, esaminiamo alcuni elementi in Tipo di risorsa: Paziente.

- Elemento: Nome La FHIR specifica imposta la cardinalità come`0..*`.

  L'elemento viene acquisito come matrice.

```
[{
 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
}]
```

  In Athena, per vedere come è stato importato un tipo di risorsa, cercalo in Tabelle e viste. Per accedere agli elementi di questo array, puoi usare la notazione a punti. Ecco un semplice esempio che consentirebbe di accedere ai valori di `given` e`family`.

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

- Elemento: MaritalStatus La FHIR specifica imposta la cardinalità come`0..1`.

  Questo elemento viene catturato comeJSON.

```
{
```

```
 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
}
```

In Athena, per vedere come è stato importato un tipo di risorsa, cercalo in Tabelle e viste. Per accedere alle coppie chiave-valore inJSON, puoi usare la notazione a punti. Poiché non è un array, non è richiesto alcun indice di matrice. Ecco un semplice esempio che consentirebbe di accedere al valore di`text`.

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

Per ulteriori informazioni sull'accesso e sulla ricercaJSON, consulta Interrogazione JSON nella Guida per l'utente di Athena.

Le istruzioni di interrogazione di Athena Data Manipulation Language (DML) si basano su Trino. Athena non supporta tutte le funzionalità di Trino e presenta differenze significative. Per ulteriori informazioni, consulta DMLle interrogazioni, le funzioni e gli operatori nella Guida per l'utente di Amazon Athena.

Inoltre, Athena supporta diversi tipi di dati che potresti incontrare durante la creazione di query sul tuo HealthLake archivio dati. Per ulteriori informazioni sui tipi di dati in Athena, consulta Tipi di dati in Amazon Athena nella Amazon Athena User Guide.

Per ulteriori informazioni su come funzionano SQL le query in Athena, SQLconsulta il riferimento per Amazon Athena nella Amazon Athena User Guide.

Ogni scheda mostra esempi di come effettuare ricerche sui tipi di risorse specificati e sugli elementi associati utilizzando Athena.

Element: Extension

L'elemento `extension` viene utilizzato per creare campi personalizzati in un archivio dati.

Questo esempio mostra come accedere alle funzionalità dell'`extension`elemento presente nel tipo di `Patient` risorsa.

Quando l'archivio HealthLake dati viene importato in Athena, gli elementi di un tipo di risorsa vengono analizzati in modo diverso. Poiché la struttura della variabile `element` is, non può essere specificata completamente nello schema. Per gestire tale variabilità, gli elementi all'interno dell'array vengono passati come stringhe.

Nella descrizione della tabella di`Patient`, è possibile vedere l'elemento `extension` descritto come`array<string>`, il che significa che è possibile accedere agli elementi dell'array utilizzando un valore di indice. Per accedere agli elementi della stringa, tuttavia, è necessario utilizzare`json_extract`.

Ecco una singola voce dell'`extension`elemento che si trova nella tabella dei pazienti.

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

Anche se questo è validoJSON, Athena lo tratta come una stringa.

Questo esempio di SQL query dimostra come creare una tabella che contenga gli `patient-mothersMaidenName` elementi and. `patient-birthPlace` Per accedere a questi elementi, è necessario utilizzare diversi indici di matrice 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
```

Per ulteriori informazioni sulle query che coinvolgonoJSON, consulta Estrazione di dati da JSON nella Guida per l'utente di Amazon Athena.

Element: birthDate (Age)

L'età non è un elemento del tipo di risorsa Patient in. FHIR Di seguito sono riportati due esempi di ricerche che filtrano in base all'età.

Poiché l'età non è un elemento, utilizziamo il `birthDate` per le SQL domande. Per vedere come è stato inserito un elementoFHIR, cerca il nome della tabella in Tabelle e viste. Puoi vedere che è di tipo stringa.

Esempio 1: calcolo di un valore per l'età

In questa SQL query di esempio, utilizziamo uno SQL strumento integrato `year` per estrarre tali componenti. `current_date` Quindi, li sottraiamo per restituire l'età effettiva del paziente, sotto forma di colonna denominata`age`.

```
SELECT
```

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

Esempio 2: Filtraggio per pazienti nati prima `2019-01-01` e che lo sono ancora. `male`

La SQL query mostra come utilizzare la `CAST` funzione per inserire l'`birthDate`elemento come tipo `DATE` e come filtrare in base a due criteri della `WHERE` clausola. Poiché l'elemento viene inserito come stringa di tipo per impostazione predefinita, è necessario che `CAST` sia di tipo. `DATE` Quindi puoi usare l'<operatore per confrontarlo con una data diversa,. `2019-01-01` Utilizzando`AND`, è possibile aggiungere un secondo criterio alla `WHERE` clausola.

```
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

Questo esempio mostra le ricerche di località all'interno del tipo di risorsa Location in cui il nome della città è Attleboro.

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

Element: Age

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

Resource type: Condition

La condizione relativa al tipo di risorsa memorizza i dati di diagnosi relativi a problemi che hanno raggiunto un livello di preoccupazione. HealthLake(), l'elaborazione medica integrata del linguaggio naturale (NLP) genera nuove `Condition` risorse sulla base dei dettagli presenti nel tipo di DocumentReference risorsa. Quando viene generata una nuova risorsa, HealthLake aggiunge il tag `SYSTEM_GENERATED` all'`meta`elemento. Questa SQL query di esempio

dimostra come è possibile cercare nella tabella delle condizioni e restituire risultati laddove i SYSTEM_GENERATED risultati sono stati rimossi.

Per ulteriori informazioni sull'elaborazione integrata HealthLake del linguaggio naturale (NLP), vedereUtilizzo della generazione automatizzata di risorse basata sull'elaborazione del linguaggio naturale (NLP) del tipo di FHIR DocumentReference risorsa in AWS HealthLake.

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

Puoi anche cercare all'interno di un elemento di stringa specificato per filtrare ulteriormente la tua query. L'modifierextensionelemento contiene dettagli su quale DocumentReference risorsa è stata utilizzata per generare un insieme di condizioni. Ancora una volta, è necessario utilizzare json_extract per accedere agli JSON elementi annidati che vengono portati in Athena come stringa.

Questa SQL query di esempio dimostra come è possibile cercare tutto ciò Condition che è stato generato in base a uno specifico. DocumentReference CASTUtilizzatelo per impostare l'JSONelemento come stringa in modo da poterlo utilizzare LIKE per il confronto.

```
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

Il tipo di risorsa, Observation, memorizza le misurazioni e le semplici asserzioni fatte su un paziente, un dispositivo o un altro argomento. HealthLake(), l'elaborazione integrata del linguaggio naturale (NLP) genera nuove Observation risorse sulla base dei dettagli presenti in una DocumentReference risorsa. Questa SQL query di esempio include WHERE meta.tag[1] is NULL commenti, il che significa che i SYSTEM_GENERATED risultati sono inclusi.

```
SELECT valueCodeableConcept.coding[1].code
```

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

Questa colonna è stata importata come `struct`. Pertanto, è possibile accedere agli elementi al suo interno utilizzando la notazione a punti.

Resource type: MedicationStatement

MedicationStatement è un tipo di FHIR risorsa che è possibile utilizzare per memorizzare i dettagli sui farmaci che un paziente ha assunto, sta assumendo o assumerà in futuro. HealthLake(), l'elaborazione medica integrata del linguaggio naturale (NLP) genera nuove MedicationStatement risorse sulla base dei documenti presenti nel tipo di DocumentReference risorsa. Quando vengono generate nuove risorse, HealthLake aggiunge il tag `SYSTEM_GENERATED` all'`meta`elemento. Questa SQL query di esempio dimostra come creare un'interrogazione che filtra in base a un singolo paziente utilizzando il relativo identificatore e trova le risorse che sono state aggiunte da HealthLake NLP

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

Per ulteriori informazioni sulla HealthLake medicina integrataNLP, vedere. Utilizzo della generazione automatizzata di risorse basata sull'elaborazione del linguaggio naturale (NLP) del tipo di FHIR DocumentReference risorsa in AWS HealthLake

SQLInterrogazioni di esempio con filtri complessi

> **Note**
>
> Dopo il 20 febbraio 2023, gli archivi HealthLake dati non utilizzano l'elaborazione integrata del linguaggio naturale (NLP) per impostazione predefinita. Se sei interessato ad attivare questa funzionalità sul tuo data store, consulta il Come posso attivare la funzionalità integrata HealthLake di elaborazione del linguaggio naturale? capitolo Risoluzione dei problemi.

Gli esempi in questo argomento includono SQL query per HealthLake l'integrazione con Athena che utilizzano filtri complessi.

Example Creazione di criteri di filtro basati su dati demografici

L'identificazione dei dati demografici corretti dei pazienti è importante quando si crea una coorte di pazienti. Questa query di esempio dimostra come utilizzare la notazione a punti Trino e `json_extract` filtrare i dati nell'archivio dati. HealthLake

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

Con la console Athena, puoi ordinare e scaricare ulteriormente i risultati.

Example Creazione di filtri per un paziente e le relative condizioni

Questa query di esempio dimostra come trovare e ordinare tutte le condizioni correlate per i pazienti presenti in un archivio HealthLake dati.

```
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;
```

Puoi utilizzare la console Athena per ordinare ulteriormente questi risultati o scaricarli per ulteriori analisi.

Example Creazione di filtri per i pazienti e le relative osservazioni

Questa query di esempio dimostra come trovare e ordinare tutte le osservazioni correlate per i pazienti presenti in un archivio HealthLake dati.

```
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 Creazione di condizioni di filtraggio per un paziente e le relative procedure

Collegare le procedure ai pazienti è un aspetto importante dell'assistenza sanitaria. Questa SQL interrogazione dimostra come è possibile utilizzare i tipi di risorse relative al paziente e alla procedura per eseguire questa operazione in Athena. Questa SQL query restituirà tutti i pazienti e le relative procedure presenti nell'archivio HealthLake dati.

```
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;
```

Ora puoi utilizzare la console Athena per scaricare i risultati per ulteriori analisi o ordinarli per comprenderli meglio.

Example Creazione di condizioni di filtraggio per un paziente e relative prescrizioni

È importante consultare un elenco aggiornato dei farmaci che i pazienti assumono. Utilizzando Athena, è possibile scrivere una SQL query che utilizza sia il tipo di paziente che quello di MedicationRequest risorse presenti nell'archivio HealthLake dati.

Questa SQL interrogazione unisce il paziente e le MedicationRequest tabelle importate in Athena. Inoltre, organizza le prescrizioni nelle rispettive voci utilizzando la notazione a punti.

```
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
```

Puoi utilizzare la console Athena per ordinare i risultati o scaricarli per ulteriori analisi.

Example Visualizzazione dei farmaci presenti nel tipo di MedicationStatement risorsa

La query di esempio mostra come organizzare il nido JSON importato in SQL Athena utilizzando. La query utilizza l'`meta`elemento per indicare quando è stato aggiunto un farmaco tramite l'elaborazione integrata HealthLake del linguaggio naturale ()NLP. Per ulteriori informazioni sull' HealthLakeintegrazione con Amazon Comprehend Medical, Utilizzo della generazione automatizzata di risorse basata sull'elaborazione del linguaggio naturale (NLP) del tipo di FHIR DocumentReference

risorsa in AWS HealthLake consulta. Viene inoltre utilizzato per `json_extract` cercare dati all'interno dell'array di JSON stringhe.

```
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;
```

Puoi usare la console Athena per scaricare questi risultati o ordinarli.

Example Filtra per un tipo di malattia specifico

L'esempio mostra come trovare un gruppo di pazienti, di età compresa tra 18 e 75 anni, a cui è stato diagnosticato il diabete.

```
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%');
```

Ora puoi usare la console Athena per ordinare i risultati o scaricarli per ulteriori analisi.

AWS HealthLake e VPC endpoint di interfaccia ()AWS PrivateLink

Puoi stabilire una connessione privata tra il tuo VPC e il tuo dispositivo AWS HealthLake creando un VPCendpoint di interfaccia. Gli VPC endpoint di interfaccia sono alimentati da AWS PrivateLink, una tecnologia che è possibile utilizzare per accedere HealthLake in modo privato, APIs senza un gateway Internet, un NAT dispositivo, una connessione o una VPN connessione. AWS Direct Connect Le istanze in uso VPC non necessitano di indirizzi IP pubblici con cui comunicare;. HealthLake APIs Il traffico tra il tuo VPC e HealthLake; non esce dalla rete Amazon.

Ogni endpoint dell'interfaccia è rappresentato da una o più interfacce di rete elastiche nelle tue sottoreti.

Per ulteriori informazioni, consulta Interface VPC endpoints (AWS PrivateLink) nella Amazon VPC User Guide.

Considerazioni sugli endpoint HealthLake VPC

Prima di configurare un VPC endpoint di interfaccia per HealthLake, assicurati di consultare le proprietà e le limitazioni degli endpoint dell'interfaccia nella Amazon VPC User Guide.

HealthLake supporta l'esecuzione di chiamate a tutte le sue API azioni dal tuo. VPC

Creazione di un VPC endpoint di interfaccia per HealthLake;

Puoi creare un VPC endpoint per il servizio HealthLake; utilizzando la VPC console Amazon o il AWS Command Line Interface (AWS CLI). Per ulteriori informazioni, consulta Creazione di un endpoint di interfaccia nella Amazon VPC User Guide.

Crea un VPC endpoint per HealthLake; utilizzando il seguente nome di servizio:

- com.amazonaws. *region*.salutelake

Se attivi la modalità privata DNS per l'endpoint, puoi effettuare API richieste HealthLake utilizzando il DNS nome predefinito per la regione. Ad esempio *healthlake.us-east-1.amazonaws.com*.

Per ulteriori informazioni, consulta Accedere a un servizio tramite un endpoint di interfaccia nella Amazon VPC User Guide.

Creazione di una policy VPC sugli endpoint per HealthLake

Puoi allegare una policy per gli endpoint al tuo VPC endpoint che controlli l'accesso a. HealthLake La policy specifica le informazioni riportate di seguito:

- Il principale che può eseguire operazioni.
- Le azioni che possono essere eseguite.
- Le risorse sui cui si possono eseguire azioni.

Per ulteriori informazioni, consulta Controllare l'accesso ai servizi con VPC endpoint nella Amazon VPC User Guide.

Esempio: policy per le azioni relative agli VPC endpoint HealthLake

Di seguito è riportato un esempio di policy sugli endpoint per. HealthLake Se collegata a un endpoint, questa policy garantisce l'accesso all' HealthLake`CreateFHIRDatastore`azione a tutti i principali su tutte le risorse.

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

Etichettare le risorse in AWS HealthLake

Puoi assegnare i metadati alle risorse AWS sotto forma di tag. Ogni tag è un'etichetta composta da una chiave e da un valore definiti dall'utente. Con i tag è possibile a gestire, identificare, organizzare, cercare e filtrare le risorse.

In questo argomento vengono descritte le categorie e le strategie di tagging comunemente utilizzate per implementare una strategia di tagging coerente ed efficace. Nelle sezioni seguenti si presuppone una conoscenza di base delle AWS risorse, dei tag, della fatturazione dettagliata e AWS dell'Identity and Access Management ()IAM.

Ogni tag è costituito da due parti:

- Una chiave di tag (ad esempio CostCenter, Ambiente o Progetto). Le chiavi dei tag prevedono una distinzione tra lettere maiuscole e minuscole.
- Un valore di tag (ad esempio, 111122223333 o Production). Analogamente alle chiavi dei tag, i valori dei tag prevedono una distinzione tra lettere maiuscole e minuscole.

È possibile utilizzare i tag per suddividere le risorse in categorie in base allo scopo, al proprietario, all'ambiente o ad altri criteri. Per ulteriori informazioni, consulta Strategie di tagging di AWS.

È possibile aggiungere, modificare o rimuovere i tag una risorsa alla volta dalla console di servizio, dal servizio API o dal. AWS CLI

Per abilitare i tag, assicurati che TagResources siano autorizzati. Puoi autorizzare TagResources allegando una IAM politica come nell'esempio seguente.

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

```
    ]
}
```

Avviso importante

AWS HealthLake protegge i dati dei clienti secondo le politiche del modello di responsabilità AWS condivisa. Ciò significa che tutti i dati dei clienti sono crittografati sia in fase di transizione che in fase di archiviazione. Tuttavia, non tutti i nomi immessi dai clienti per gli archivi dati o le operazioni basate sul lavoro sono crittografati. Non devono mai contenere informazioni di identificazione personale o informazioni sanitarie protette. Per ulteriori informazioni, consulta il capitolo AWS HealthLake Sicurezza.

Best practice

Quando crei una strategia di tagging per le risorse AWS, segui le procedure consigliate:

- Non archiviare informazioni di identificazione personale (PII), informazioni sulla salute personale (PHI) o altre informazioni sensibili nei tag.
- Utilizza un formato standardizzato con distinzione tra maiuscole e minuscole per i tag e applicalo in modo coerente a tutti i tipi di risorse.
- Prendi in considerazione le linee guida per i tag che supportano più scopi, ad esempio la gestione del controllo dell'accesso alle risorse, il monitoraggio dei costi, l'automazione e l'organizzazione.
- Utilizza strumenti automatizzati per aiutare a gestire i tag delle risorse. AWS Resource Groups e Resource Groups Tagging API consentono il controllo programmatico dei tag, rendendo possibile la gestione, la ricerca e il filtraggio automatici di tag e risorse.
- L'etichettatura è più efficace quando si utilizzano più tag.
- I tag possono essere modificati o modificati in base alle esigenze degli utenti, tuttavia per aggiornare i tag di controllo dell'accesso, è necessario aggiornare anche le politiche che fanno riferimento a tali tag per controllare l'accesso alle risorse.

Requisiti per il tagging

I tag hanno i requisiti seguenti:

- Le chiavi non possono avere come prefisso aws:.
- Le chiavi devono essere univoche per un set di tag.

- Una chiave deve essere costituita da un numero di caratteri compreso tra 1 e 128.
- Un valore deve essere costituito da un numero di caratteri compreso tra 0 e 256.
- Non è necessario che i valori siano univoci per un set di tag.
- I caratteri consentiti per le chiavi e i valori sono lettere Unicode, cifre, spazi e uno qualsiasi dei simboli seguenti: _ . : / = + - @.
- Per chiavi e valori viene fatta distinzione tra maiuscole e minuscole.

Aggiungere un tag a un archivio dati

L'aggiunta di tag a un data store può aiutarti a identificare e organizzare AWS le tue risorse e a gestirne l'accesso. Innanzitutto, aggiungi uno o più tag (coppie chiave-valore) a un Data Store. Puoi utilizzare fino a cinquanta tag per utente. Esistono anche restrizioni sui caratteri che è possibile utilizzare nei campi chiave e valore.

Dopo aver creato i tag, puoi creare IAM politiche per gestire l'accesso al data store in base a questi tag. È possibile utilizzare la HealthLake console o AWS CLI aggiungere tag a un data store. L'aggiunta di tag a un repository può avere impatto sull'accesso a tale repository. Prima di aggiungere un tag a un data store, assicurati di esaminare tutte IAM le politiche che potrebbero utilizzare i tag per controllare l'accesso a risorse come gli archivi dati.

Segui questi passaggi per utilizzare AWS CLI per aggiungere un tag a un HealthLake data store. Per aggiungere un tag a un data store quando lo crei, vediCreazione di un archivio dati in AWS HealthLake.

Nel terminale o nella riga di comando, esegui il comando tag-resource, specificando Amazon Resource Name (ARN) del data store a cui desideri aggiungere i tag e la chiave e il valore del tag che desideri aggiungere. Puoi aggiungere più di un tag a un data store. Esistono anche restrizioni sui caratteri che è possibile utilizzare nei campi chiave e valore, come elencato in Requisiti per il tagging Ad esempio, per aggiungere tag a un archivio dati durante la creazione, è necessario utilizzare il seguente comando in AWS CLI. Il nome del data store è Test_Data_Store e i due tag aggiunti con chiavi sono key1 e key2 con valori rispettivamente come value1 e value2 :

```
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
```

Per aggiungere tag a un archivio dati esistente, esegui il seguente comando di esempio:

```
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
```

In caso di successo, questo comando non restituisce alcuna risposta.

Elenco dei tag per un archivio dati

Segui questi passaggi per utilizzare AWS CLI per visualizzare un elenco dei AWS tag per un HealthLake Data Store. Se non sono stati aggiunti tag, l'elenco restituito è vuoto.

Nel terminale o nella riga di comando, esegui il list-tags-for-resource comando come illustrato nell'esempio seguente.

```
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"
    }
}
```

Rimozione di tag da un archivio dati

È possibile rimuovere uno o più tag associati a un data store. La rimozione di un tag non elimina il tag da altre risorse AWS associate a tale tag.

Nel terminale o nella riga di comando, esegui il comando untag-resource, specificando Amazon Resource Name ARN () del data store in cui desideri rimuovere i tag e la chiave del tag che desideri rimuovere.

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

In caso di successo, questo comando non restituisce una risposta. Per verificare i tag associati all'archivio dati, esegui il list-tags-for-resource comando.

Monitoraggio HealthLake

Il monitoraggio è un elemento importante per mantenere l'affidabilità, la disponibilità e le prestazioni delle HealthLake altre AWS soluzioni esistenti. AWSfornisce i seguenti strumenti di monitoraggio per osservare HealthLake, segnalare quando qualcosa non va e intraprendere azioni automatiche quando necessario:

- Amazon CloudWatch monitora AWS le tue risorse e le applicazioni su cui esegui AWS in tempo reale. Puoi raccogliere e tenere traccia dei parametri, creare dashboard personalizzati e impostare allarmi che ti avvisano o intraprendono azioni quando una determinata metrica raggiunge una soglia specifica. Ad esempio, puoi tenere CloudWatch traccia CPU dell'utilizzo o di altri parametri delle tue EC2 istanze Amazon e avviare automaticamente nuove istanze quando necessario. Per ulteriori informazioni, consulta la Amazon CloudWatch User Guide.
- AWS CloudTrailregistra le API chiamate e gli eventi correlati effettuati da o per conto del tuo AWS account. Distribuisce quindi i file di log a un bucket Amazon S3 specificato. È possibile identificare gli utenti e gli account chiamati AWS, l'indirizzo IP di origine di tali chiamate e il momento in cui si sono verificate. Per ulteriori informazioni, consulta la Guida per l'utente AWS CloudTrail.

Argomenti

Monitoraggio HealthLake con Amazon CloudWatch

È possibile monitorare HealthLake l'utilizzo CloudWatch, che raccoglie dati grezzi e li elabora in metriche leggibili e quasi in tempo reale. Queste statistiche vengono conservate per 15 mesi, quindi puoi utilizzare tali informazioni storiche e avere una prospettiva migliore sulle prestazioni della tua applicazione o del tuo servizio web. È anche possibile impostare allarmi che controllano determinate soglie e inviare notifiche o intraprendere azioni quando queste soglie vengono raggiunte. Per ulteriori informazioni, consulta la Amazon CloudWatch User Guide.

Le metriche vengono riportate per tutti HealthLake APIs, inclusi i seguenti.

- gestione dell'archivio datiAPIs: CreateFHIRDatastore, D, DeleteFHIRDatastore, L escribeFHIRDatastore istFHIRDatastores
- Importazione ed esportazione APIs —S tartFHIRImport Job, L istFHIRImport Jobs, D escribeFHIRImport Job, S tartFHIRExport Job, L istFHIRExport Jobs, D escribeFHIRExport Job

- HTTPRESTGestione dei clienti e delle risorse APIs — CreateResource DeleteResource, GetCapabilities, ReadResource,, SearchAll, SearchWithGet SearchWithPost, UpdateResource.
- Etichettatura APIs — ListTagsForResource, TagResource UntagResource

Le seguenti tabelle elencano i parametri e le dimensioni di HealthLake.

Vengono riportate le seguenti metriche. Ciascuna viene presentata come conteggio delle frequenze per un intervallo di dati specificato dall'utente.

Metriche

Metriche	Descrizione
Numero di chiamate	Il numero di chiamate versoAPIs. Questo può essere segnalato per l'account o per un archivio dati specificato. Unità: numero Statistiche valide: somma, conteggio Dimensioni: funzionamento, ID dell'archivio dati, tipo di archivio dati
Richieste riuscite	Il numero di API richieste andate a buon fine. Unità: numero Statistiche valide: Sum, Average Dimensioni: funzionamento, archivio dati, tipo di archivio dati
Errori dell'utente	Il numero di richieste non riuscite a causa di un errore dell'utente. Unità: numero Statistiche valide: Sum, Average

Metriche	Descrizione
	Dimensioni: funzionamento, ID dell'archivio dati, tipo di archivio dati
Errori del server	Il numero di richieste non riuscite a causa di un errore del server. Unità: numero Statistiche valide: Sum, Average Dimensioni: funzionamento, ID dell'archivio dati, tipo di archivio dati
Richieste con throttling	Il numero di richieste che sono state limitate. Questa metrica non è inclusa nel conteggio degli errori degli utenti o del server. Unità: numero Statistiche valide: Sum, Average Dimensioni: funzionamento, ID dell'archivio dati, tipo di archivio dati
Latenza	Il tempo impiegato in millisecondi per elaborare la richiesta dell'utente. Unità: millisecondi Statistiche valide: Minima, Massima, Media Dimensioni: funzionamento, ID dell'archivio dati, tipo di archivio dati

Vengono riportate le seguenti dimensioni.

Dimensioni

Dimensioni	Descrizione
Operazione	Quale API operazione è stata utilizzata
DataStoreID	L'archivio dati incluso nella API richiesta
DataStoreType	Il tipo di archivio dati (attualmente è supportato solo FHIR R4)

Puoi ottenere metriche per la console HealthLake di AWS gestione, il AWS CLI, o il. CloudWatch API Puoi utilizzarlo CloudWatch API tramite uno degli Amazon AWS Software Development Kit (SDKs) o gli CloudWatch API strumenti. La HealthLake console visualizza grafici basati sui dati grezzi di. CloudWatch API

È necessario disporre delle CloudWatch autorizzazioni appropriate con cui eseguire il monitoraggio HealthLake . CloudWatch Per ulteriori informazioni, consulta Authentication and Access Control for Amazon CloudWatch nella Amazon CloudWatch User Guide.

Visualizzazione delle HealthLake metriche

Per visualizzare le metriche (console) CloudWatch

1. Accedi alla console di gestione AWS e apri laconsole CloudWatch.
2. Scegli Metriche, scegli Tutte le metriche, quindi scegli/. AWS HealthLake
3. Scegliere la dimensione, selezionare un nome parametro e scegliere Add to graph (Aggiungi a grafico).
4. Seleziona un valore per l'intervallo di date. Il numero di parametri per l'intervallo di date selezionato è visualizzato nel grafico.

Creazione di un allarme utilizzando CloudWatch

Un CloudWatch allarme controlla una singola metrica in un periodo di tempo specificato ed esegue una o più azioni: inviare una notifica a un argomento di Amazon Simple Notification Service (AmazonSNS) o a una politica di Auto Scaling. L'azione o le azioni si basano sul valore della metrica relativo a una determinata soglia per un certo numero di periodi di tempo specificati. CloudWatch può anche inviarti un SNS messaggio Amazon quando l'allarme cambia stato.

CloudWatch gli allarmi attivano azioni solo quando lo stato cambia e persiste per il periodo specificato.

Per visualizzare le metriche (console) CloudWatch

1. Accedi alla console di gestione AWS e apri laconsole CloudWatch.
2. Seleziona Alarms (Allarmi), quindi Create Alarm (Crea allarme).
3. Scegli AWS/HealthLake, quindi scegli una metrica.
4. In Time Range (Intervallo di tempo), scegli un intervallo di tempo durante il quale eseguire il monitoraggio, quindi scegli Next (Successivo).
5. Immetti il Name (Nome) e la Description (Descrizione).
6. Per Whenever, scegliete >= e digitate un valore massimo.
7. Se desideri CloudWatch inviare un'e-mail quando viene raggiunto lo stato di allarme, nella sezione Azioni, per Ogni volta che questo allarme è ALARM, scegli Stato. Per Invia notifica a, scegli una mailing list o scegli Nuova lista e crea una nuova mailing list.
8. Visualizza un'anteprima dell'allarme nella sezione Alarm Preview (Anteprima allarme). Se sei soddisfatto dell'allarme, scegli Create Alarm (Crea allarme).

Integrazione SMART con FHIR AWS HealthLake

Applicazioni mediche sostituibili e tecnologie riutilizzabili (SMART) su un HealthLake data store FHIR abilitato consente SMART alle applicazioni FHIR conformi di accedere ai dati archiviati in un archivio dati. HealthLake HealthLake l'accesso ai dati avviene autenticando e autorizzando le richieste utilizzando un server di autorizzazione di terze parti e configurando risorse aggiuntive in. AWS

Per utilizzare l'SMARTon FHIR con il tuo HealthLake data store, devi fornire quanto segue nella tua richiesta C. reateFHIRDatastore API

- Imposta AuthorizationStrategyuguale a`SMART_ON_FHIR_V1`.
- Imposta il IdpLambdaArn valore uguale a quello AWS Lambda che hai creato per gestire la decodifica ARN dei token con il tuo server di autorizzazione.
- Definite gli elementi di metadati specificati nel vostro server di autorizzazione. Questi elementi di metadati vengono restituiti nel Discovery Document. Per ulteriori informazioni, consulta Recupero di un Discovery Document SMART su un HealthLake data store FHIR abilitato.
- Facoltativo: FineGrainedAuthorizationEnabledabilitalo se hai impostato un'autorizzazione granulare sul tuo server di autorizzazione.

Puoi creare un SMART archivio dati non FHIR abilitato utilizzando AWS Command Line Interface (AWS CLI) o tramite uno dei AWS supportiSDKs. La creazione di un SMART HealthLake data store non FHIR abilitato non è supportata utilizzando la HealthLake console. Per ulteriori informazioni, consulta Crea un SMART data store non FHIR abilitato.

Per inserire questi parametri nella richiesta, è necessario configurare le risorse in altri AWS servizi (AWS Secrets Manager and AWS Lambda), creare nuovi ruoli di IAM servizio e configurare un SMART server di autorizzazione FHIR conforme. Utilizza la sezione Configurazione delle risorse necessarie per implementare un data store FHIR non conforme per saperne di più SMART sulla configurazione delle risorse richieste e per visualizzare una panoramica di alto livello su come interagisce un'SMARTapplicazione on. FHIR HealthLake

Ciò significa che, anziché gestire le credenziali utente tramite l'utente, AWS Identity and Access Management si utilizza un SMART server di autorizzazione non conforme. FHIR

HealthLake supporta la versione 1.0SMART. FHIR Per ulteriori informazioni su questo framework, vedere SMARTApplication Launch Framework Implementation Guide Release 1.0.

Per autorizzare e autenticare le richieste di data store utilizzando SMART onFHIR, HealthLake supporta l'utilizzo di:

- Integrazione OpenID (AuthN): utilizzata per autenticare la persona o l'applicazione client è chi (o cosa) dichiara di essere.
- OAuthIntegrazione 2.0 (AuthZ): utilizzata per autorizzare FHIR le risorse dell'archivio HealthLake dati che una richiesta autenticata può anche leggere o scrivere dati. Questo è definito dagli ambiti configurati nel server di autorizzazione

Indice

- Recupero di un Discovery Document SMART su un HealthLake data store FHIR abilitato
- Effettuare una FHIR REST API richiesta su un HealthLake data store SMART abilitato
- Configurazione delle risorse necessarie per implementare un SMART data store non FHIR conforme
  - In che modo un'applicazione client avvia e richiede dati da un SMART data store FHIR onenable HealthLake

Requisiti di autenticazione per SMART on FHIR

Per accedere alle FHIR risorse in un SMART FHIR HealthLake data store, un'applicazione client deve essere autorizzata da un server di autorizzazione OAuth conforme alla versione 2.0 e presentare un token OAuth Bearer come parte di una richiesta. FHIR REST API Per trovare l'endpoint del server di autorizzazione, utilizzate l' HealthLake SMARTon FHIR Discovery Document tramite un Known Uniform Resource Identifier. Per ulteriori informazioni su questo processo, consultare Recupero di un Discovery Document SMART su un HealthLake data store FHIR abilitato.

Quando si crea un FHIR HealthLake data store SMART on, è necessario definire l'endpoint del server di autorizzazione e l'endpoint del token nell'`metadata`elemento della richiesta C. reateFHIRDatastore Per ulteriori informazioni sulla definizione dell'`metadata`elemento, consultaCreazione di un SMART HealthLake data store non FHIR abilitato.

Utilizzando gli endpoint del server di autorizzazione, l'applicazione client autenticherà un utente con il servizio di autorizzazione. Una volta autorizzato e autenticato, un JSON Web Token (JWT) viene generato dal servizio di autorizzazione e passato all'applicazione client. Questo token contiene ambiti di FHIR risorse che l'applicazione client può utilizzare, il che a sua volta limita i dati a cui l'utente è in grado di accedere. Facoltativamente, se è stato fornito l'ambito di avvio, la risposta conterrà tali dettagli. Per ulteriori informazioni SMART sugli FHIR onscope supportati da HealthLake, consulta. Supportato SMART sui FHIR OAuth cannocchiali da HealthLake

Utilizzando i JWT dati concessi dal server di autorizzazione, un'applicazione client effettua FHIR REST API chiamate a un SMART archivio HealthLake dati FHIR abilitato. Per convalidare e decodificareJWT, è necessario creare una funzione Lambda. HealthLake richiama questa funzione Lambda per tuo conto quando viene ricevuta una FHIR REST API richiesta. Per vedere un esempio di funzione Lambda di avvio, vedere. Utilizzo AWS Lambda per la convalida dei token con un SMART archivio HealthLake dati FHIR abilitato

Elementi del server di autorizzazione necessari per creare un archivio SMART dati FHIR abilitato HealthLake

Nella reateFHIRDatastore richiesta C, è necessario fornire l'endpoint di autorizzazione e l'endpoint del token come parte dell'`metadata`elemento nell'`IdentityProviderConfiguration`oggetto. Sono necessari sia l'endpoint di autorizzazione che l'endpoint del token. Per vedere un esempio di come questo è specificato nella reateFHIRDatastore richiesta C, vedi. Creazione di un SMART HealthLake data store non FHIR abilitato

Dichiarazioni obbligatorie per completare una FHIR REST API richiesta SMART su un archivio HealthLake dati non FHIR abilitato

La AWS Lambda funzione deve contenere le seguenti affermazioni affinché sia una FHIR REST API richiesta valida SMART su un HealthLake data store non FHIR abilitato.

- `nbf`: Reclamo (non prima): l'attestazione «nbf» (non precedente) identifica il momento entro il quale l'attestazione JWT MUST NOT deve essere accettata per l'elaborazione. L'elaborazione del reclamo «nbf» richiede che la corrente sia date/time MUST be after or equal to the not-before date/time elencata nel reclamo «nbf». La funzione Lambda di esempio che forniamo converte `iat` dalla risposta del server in. `nbf`
- `exp`: Richiesta (ora di scadenza): l'attestazione «exp» (ora di scadenza) identifica l'ora di scadenza in cui o dopo la quale non JWT deve essere accettata per l'elaborazione.
- `isAuthorized`: Un valore booleano impostato su. `True` Indica che la richiesta è stata autorizzata sul server di autorizzazione.
- `aud`: Dichiarazione (Audience) - L'attestazione «aud» (audience) identifica i destinatari a cui JWT è destinata. Questo deve essere un endpoint di SMART HealthLake data store non FHIR abilitato.
- `scope`: Questo deve essere almeno un ambito correlato alle FHIR risorse. Questo ambito è definito sul server di autorizzazione. Per ulteriori informazioni sugli ambiti relativi alle FHIR risorse accettati da HealthLake, vedereHealthLake ambiti specifici FHIR delle risorse dell'archivio dati.

Supportato SMART sui FHIR OAuth cannocchiali da HealthLake

HealthLake utilizza OAuth 2.0 come protocollo di autorizzazione. L'utilizzo di questo protocollo sul server di autorizzazione consente di definire a quali FHIR risorse del HealthLake data store un'applicazione client può avere accesso anche in lettura e/o scrittura.

Il FHIR framework SMART on definisce un insieme di ambiti che possono essere richiesti al server di autorizzazione. Per visualizzare le definizioni degli ambiti nel FHIR framework SMART on, consulta SMARTFHIRScopes nella HL7FHIRResource Guide.

Ad esempio, un'applicazione client progettata esclusivamente per consentire ai pazienti di visualizzare i risultati di laboratorio o visualizzare i propri dati di contatto dovrebbe essere autorizzata a richiedere (tramite FHIR REST richiesta) `read` gli ambiti. Per definirli come ambito, è necessario fornire una stringa come la seguente`patient/Observation.read`. Ciò consentirebbe all'applicazione client di richiedere l'accesso al tipo di `Observation` risorsa in modalità di sola lettura sul tipo di `Patient` risorsa.

Ambito di lancio autonomo

HealthLake supporta l'ambito della modalità di avvio autonoma. `launch/patient`

In modalità di avvio autonoma, un'applicazione client richiede l'accesso ai dati clinici del paziente perché l'utente e il paziente non sono noti all'applicazione client. Pertanto, la richiesta di autorizzazione dell'applicazione client richiede esplicitamente la restituzione dell'ambito del paziente. Una volta completata con successo l'autenticazione, il server di autorizzazione emette un token di accesso contenente l'ambito del paziente di avvio richiesto. Il contesto paziente necessario viene fornito insieme al token di accesso nella risposta del server di autorizzazione.

Ambiti della modalità di avvio supportati

Ambito	Descrizione
`launch/patient`	Un parametro in una richiesta di autorizzazione OAuth 2.0 che richiede la restituzione dei dati del paziente nella risposta di autorizzazione.

HealthLake ambiti specifici FHIR delle risorse dell'archivio dati

HealthLake definisce tre livelli di ambiti.

- Gli ambiti specifici del paziente garantiscono l'accesso a dati specifici su un singolo paziente. Quale paziente viene specificato nel contesto del lancio.
- Gli ambiti a livello utente garantiscono l'accesso a dati specifici a cui un utente può accedere.
- Gli ambiti a livello di sistema garantiscono l'accesso in lettura/scrittura a tutte le FHIR risorse presenti nell'archivio dati. HealthLake

La tabella seguente mostra la sintassi per la costruzione di ambiti relativi alle FHIR risorse supportati da. HealthLake Il formato generale è il seguente:

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

Ambiti di autorizzazione supportati negli archivi HealthLake dati

Sintassi dell'ambito	Ambito di esempio	Risultato
`patient/(fhir-resource | '*'). ('read' | 'write' | '*')`	`patient/A llergyInt olerance.*`	Un'applicazione client avrebbe accesso in lettura/scrittura alle allergie.
`user/(fhir-resource | '*').('re ad' | 'write' | '*')`	`user/Obse rvation.read`	Un'applicazione client avrebbe accesso in lettura a tutte le osservazioni registrate.
`system/('read' | 'write' | *)`	`system/*.*`	Un'applicazione client avrebbe accesso in lettura/scrittura a tutti i dati.

Utilizzo AWS Lambda per la convalida dei token con un SMART archivio HealthLake dati FHIR abilitato

Quando si crea un SMART HealthLake data store non FHIR abilitato, è necessario fornire ARN la AWS Lambda funzione nella `CreateFHIRDatastore` richiesta. La funzione Lambda ARN viene specificata nell'`IdentityProviderConfiguration`oggetto utilizzando il `IdpLambdaArn` parametro.

È necessario creare la funzione Lambda prima di creare il proprio SMART HealthLake data store FHIR abilitato. Una volta creato il data store, la Lambda ARN non può essere modificata. Per vedere la Lambda ARN specificata al momento della creazione del data store, usa l'`DescribeFHIRDatastore`APIoperazione.

Affinché una FHIR REST richiesta abbia esito positivo SMART su un HealthLake data store non FHIR abilitato, la funzione Lambda deve eseguire le seguenti operazioni:

- La funzione Lambda deve restituire una risposta in meno di 1 secondo all'endpoint del HealthLake data store.
- Decodifica il token di accesso fornito nell'intestazione di autorizzazione della REST API richiesta inviata dall'applicazione client.
- Assegna un ruolo IAM di servizio con autorizzazioni sufficienti per eseguire la richiesta. FHIR REST API
- Le seguenti affermazioni sono necessarie per completare una FHIR REST API richiesta. Per ulteriori informazioni, consulta Reclami obbligatori.
  - `nbf`
  - `exp`
  - `isAuthorized`
  - `aud`
  - `scope`

Quando lavori con Lambda, devi creare un ruolo di esecuzione e una policy basata sulle risorse oltre alla funzione Lambda. Il ruolo di esecuzione di una funzione Lambda è un IAM ruolo che concede alla funzione l'autorizzazione ad accedere ai AWS servizi e alle risorse necessari in fase di esecuzione. La politica basata sulle risorse fornita deve consentire di richiamare la funzione HealthLake per conto dell'utente.

Le sezioni di questo argomento descrivono una richiesta di esempio da un'applicazione client e una risposta decodificata, i passaggi necessari per creare una funzione AWS Lambda e come creare una politica basata sulle risorse che possa presupporre. HealthLake

- Parte 1: creazione di una funzione Lambda
- Parte 2: creazione di un ruolo HealthLake di servizio utilizzato dalla funzione AWS Lambda
- Parte 3: Aggiornamento del ruolo di esecuzione della funzione Lambda
- Parte 4: aggiunta di una politica delle risorse alla funzione Lambda
- Parte 5: Fornire la concorrenza per la funzione Lambda

Creazione di una funzione AWS Lambda

La funzione Lambda creata in questo argomento viene attivata quando HealthLake riceve una richiesta a un data store non SMART FHIR HealthLake abilitato. La richiesta proveniente dall'applicazione client contiene una REST API chiamata e un'intestazione di autorizzazione contenente un token di accesso.

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

L'esempio della funzione Lambda in questo argomento utilizza AWS Secrets Manager per oscurare le credenziali relative al server di autorizzazione. Consigliamo vivamente di non fornire i dettagli di accesso al server di autorizzazione direttamente in una funzione Lambda.

Example convalida di una FHIR REST richiesta contenente un token del portatore di autorizzazione

La funzione Lambda di esempio mostra come convalidare una FHIR REST richiesta inviata a un data store non SMART FHIR HealthLake abilitato. Per visualizzare step-by-steps le istruzioni su come implementare questa funzione Lambda, vedere. Creazione di una funzione Lambda utilizzando AWS Management Console

Se la FHIR REST API richiesta non contiene un endpoint, un token di accesso e un'RESToperazione di Data Store validi, la funzione Lambda avrà esito negativo. Per ulteriori informazioni sugli elementi richiesti del server di autorizzazione, consulta. Reclami obbligatori

```
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()
```

Creazione di una funzione Lambda utilizzando AWS Management Console

Questa procedura presuppone che tu abbia già creato il ruolo di servizio che desideri HealthLake assumere quando gestisci una FHIR REST API richiesta SMART su un HealthLake data store non FHIR abilitato. Se non hai creato il ruolo di servizio, puoi comunque creare la funzione Lambda. È necessario aggiungere il ruolo ARN of service prima che la funzione Lambda funzioni. Per ulteriori informazioni sulla creazione di un ruolo di servizio e sulla sua specificazione nella funzione Lambda, consulta, Creazione di un ruolo di HealthLake servizio da utilizzare nella funzione AWS Lambda utilizzata per decodificare un JWT

Per creare una funzione Lambda ()AWS Management Console

1. Aprire la pagina Funzioni della console Lambda.
2. Scegli Crea funzione.
3. Scegli Crea da zero.
4. In Informazioni di base, inserisci il nome di una funzione. In Runtime scegli un runtime basato su Python.
5. In Execution role (Ruolo di esecuzione), scegli Create a new role with basic Lambda permissions (Crea un nuovo ruolo con le autorizzazioni Lambda di base).

   Lambda crea un ruolo di esecuzione che concede alla funzione l'autorizzazione a caricare i log su Amazon. CloudWatch La funzione Lambda assume il ruolo di esecuzione quando si richiama la funzione e utilizza il ruolo di esecuzione per creare credenziali per. AWS SDK
6. Scegliete la scheda Codice e aggiungete la funzione Lambda di esempio.

   Se non hai ancora creato il ruolo di servizio per la funzione Lambda da utilizzare, dovrai crearlo prima che la funzione Lambda di esempio funzioni. Per ulteriori informazioni sulla creazione di un ruolo di servizio per la funzione Lambda, vedere. Creazione di un ruolo di HealthLake servizio da utilizzare nella funzione AWS Lambda utilizzata per decodificare un JWT

```
import base64
import boto3
import logging
import json
import os
from urllib import request, parse

logger = logging.getLogger()
logger.setLevel(logging.INFO)
```

```
## Uses Secrets manager to gain access to the access key ID and secret access key
 for the authorization server
client = boto3.client('secretsmanager', region_name="region-of-datastore")
response = client.get_secret_value(SecretId='name-specified-by-customer-in-
secretsmanager')
secret = json.loads(response['SecretString'])
client_id = secret['client_id']
client_secret = secret['client_secret']


unencoded_auth = f'{client_id}:{client_secret}'
headers = {
  'Authorization': f'Basic {base64.b64encode(unencoded_auth.encode()).decode()}',
  'Content-Type': 'application/x-www-form-urlencoded'
}

auth_endpoint = os.environ['auth-server-base-url'] # Base URL of the Authorization
 server
user_role_arn = os.environ['iam-role-arn'] # The IAM role client application will
 use to complete the HTTP request on the datastore

def lambda_handler(event, context):
    if 'datastoreEndpoint' not in event or 'operationName' not in event or
 'bearerToken' not in event:
    return {}

    datastore_endpoint = event['datastoreEndpoint']
    operation_name = event['operationName']
    bearer_token = event['bearerToken']
    logger.info('Datastore Endpoint [{}], Operation Name:
 [{}]'.format(datastore_endpoint, operation_name))

    ## To validate the token
    auth_response = auth_with_provider(bearer_token)
    logger.info('Auth response: [{}]'.format(auth_response))
    auth_payload = json.loads(auth_response)
    ## Required parameters needed to be sent to the datastore endpoint for the HTTP
 request to go through
    auth_payload["isAuthorized"] = bool(auth_payload["active"])
    auth_payload["nbf"] = auth_payload["iat"]
    return {"authPayload": auth_payload, "iamRoleARN": user_role_arn}

## Access the server
```

```
def auth_with_provider(token):
    data = {'token': token, 'token_type_hint': 'access_token'}
    req = request.Request(url=auth_endpoint + '/v1/introspect',
 data=parse.urlencode(data).encode(), headers=headers)
    with request.urlopen(req) as resp:
    return resp.read().decode()
```

Modifica del ruolo di esecuzione di una funzione Lambda

Dopo aver creato la funzione Lambda, è necessario aggiornare il ruolo di esecuzione per includere le autorizzazioni necessarie per chiamare Secrets Manager. In Secrets Manager, ogni segreto che crei ha unARN. Per applicare il privilegio minimo, il ruolo di esecuzione deve avere accesso solo alle risorse necessarie per l'esecuzione della funzione Lambda.

Puoi modificare il ruolo di esecuzione di una funzione Lambda cercandola nella IAM console o scegliendo Configurazione nella console Lambda. Per ulteriori informazioni sulla gestione del ruolo di esecuzione delle funzioni Lambda, consulta. Ruolo di esecuzione Lambda

Example Ruolo di esecuzione della funzione Lambda che concede l'accesso a **`GetSecretValue`**

L'aggiunta dell'IAMazione `GetSecretValue` al ruolo di esecuzione concede l'autorizzazione necessaria per il funzionamento della funzione Lambda di esempio.

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

A questo punto hai creato una funzione Lambda che può essere utilizzata per convalidare il token di accesso fornito come parte della FHIR REST richiesta inviata al tuo SMART data store FHIR abilitato HealthLake .

Creazione di un ruolo di HealthLake servizio da utilizzare nella funzione AWS Lambda utilizzata per decodificare un JWT

Persona: amministratore IAM

Un utente che può aggiungere o rimuovere IAM politiche e creare nuove IAM identità.

Ruolo del servizio

Un ruolo di servizio è un IAMruolo che un servizio assume per eseguire azioni per conto dell'utente. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio da IAM. Per ulteriori informazioni, consulta Creare un ruolo per delegare le autorizzazioni a un utente Servizio AWS nella Guida per l'IAMutente.

Dopo la decodifica del JSON Web Token (JWT), anche l'autorizzazione di cui Lambda deve restituire IAM un ruolo. ARN Questo ruolo deve disporre delle autorizzazioni necessarie per eseguire la REST API richiesta o fallirà a causa di autorizzazioni insufficienti.

Quando si imposta una politica personalizzata, è meglio IAM utilizzarla, concedere le autorizzazioni minime richieste. Per ulteriori informazioni, consulta Applica le autorizzazioni con privilegi minimi nella Guida per l'utente. IAM

La creazione di un ruolo di HealthLake servizio da designare nella funzione di autorizzazione Lambda richiede due passaggi.

- Innanzitutto, è necessario creare una IAM politica. La policy deve specificare l'accesso alle FHIR risorse per le quali sono stati forniti gli ambiti nel server di autorizzazione.
- In secondo luogo, è necessario creare il ruolo di servizio. Quando si crea il ruolo, si designa una relazione di fiducia e si allega la politica creata nella prima fase. La relazione di fiducia viene designata HealthLake come responsabile del servizio. In questo passaggio è necessario specificare un archivio HealthLake dati ARN e un ID AWS account.

Creazione di una nuova IAM politica

Gli ambiti definiti nel server di autorizzazione determinano le FHIR risorse a cui un utente autenticato ha accesso in un archivio HealthLake dati.

La IAM policy che crei può essere personalizzata in base agli ambiti che hai definito.

È possibile definire le seguenti azioni nell'`Action`elemento di una dichiarazione IAM politica. Per ogni `Action` elemento della tabella è possibile definire un`Resource types`. In HealthLake un data store è l'unico tipo di risorsa supportato che può essere definito nell'`Resource`elemento di una dichiarazione di politica di IAM autorizzazione.

Le singole FHIR risorse non sono risorse che è possibile definire come elemento di una politica di IAM autorizzazione.

Azioni definite da HealthLake

Azioni	Descrizione	Livello di acces	Tipo di risorsa (obbligatorio)
CreateResource	Concede l'autorizzazione a creare una risorsa	Scrittu	DatastoreARN: arn:aws:healthlake: ::datastore/fhir/ **your-region 111122223333 your-datastore-id**
DeleteResource	Concede l'autorizzazione per eliminare la risorsa	Scrittu	Archivio datiARN: arn:aws:healthlake: **your-region** ::datastore/fhir/ **111122223333 your-datastore-id**
ReadResource	Concede l'autorizzazione per leggere la risorsa	Lettur	Archivio datiARN: arn:aws:healthlake: **your-region** ::datastore/fhir/ **111122223333 your-datastore-id**
SearchWithGet	Concede il permesso di cercare risorse con metodo GET	Lettur	DatastoreARN: arn:aws:healthlake: ::datastore/fhir/ **your-region 111122223333 your-datastore-id**
SearchWithPost	Concede il permesso di cercare risorse con metodo POST	Lettur	DatastoreARN: arn:aws:healthlake: ::datastore/fhir/ **your-region 111122223333 your-datastore-id**
StartFHIRExportJobWithPost	Concede l'autorizzazione a iniziare un lavoro di FHIR esportazione con GET	Scrittu	DatastoreARN: arn:aws:healthlake: ::datastore/fhir/ **your-region 111122223333 your-datastore-id**

Azioni	Descrizione	Livello di accesso	Tipo di risorsa (obbligatorio)
UpdateResource	Concede l'autorizzazione per aggiornare la risorsa	Scrittura	Archivio datiARN: arn:aws:healthlake: **your-region** ::datastore/fhir/ **111122223333 your-datastore-id**

Per iniziare, puoi usare. `AmazonHealthLakeFullAccess` Questa politica consentirebbe la lettura, la scrittura, la ricerca e l'esportazione su tutte FHIR le risorse presenti in un archivio dati. Per concedere autorizzazioni di sola lettura su un data store, utilizzare. `AmazonHealthLakeReadOnlyAccess`

Per ulteriori informazioni sulla creazione di una politica personalizzata utilizzando, oppure AWS Management Console AWS CLI IAMSDKs, consulta Creazione di IAM politiche nella Guida per l'IAMutente.

Creazione di un ruolo di servizio per HealthLake (IAMconsole)

Utilizzare questa procedura per creare un ruolo di servizio. Quando si crea un servizio, è inoltre necessario definire una IAM politica.

Per creare il ruolo di servizio per HealthLake (IAMconsole)

1. Accedi a AWS Management Console e apri la IAM console all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione della console IAM seleziona Roles (Ruoli).
3. Quindi seleziona Create role (Crea ruolo).
4. Nella pagina Seleziona entità fiduciaria, scegli Politica di fiducia personalizzata.
5. Successivamente, in Politica di fiducia personalizzata, aggiorna la politica di esempio come segue. Sostituiscilo **`your-account-id`** con il tuo numero ARN di account e aggiungi il data store che desideri utilizzare nei processi di importazione o esportazione.

```
{
  "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. Quindi, seleziona Next (Successivo).
7. Nella pagina Aggiungi autorizzazioni, scegli la politica che desideri che il HealthLake servizio assuma. Per trovare la tua politica, cercala in Politiche di autorizzazione.
8. Quindi, scegli Allega politica.
9. Quindi nella pagina Nome, rivedi e crea in Nome del ruolo, inserisci un nome.
10. (Facoltativo) Quindi, in Descrizione, aggiungi una breve descrizione del tuo ruolo.
11. Se possibile, specifica un nome del ruolo o un suffisso del nome del ruolo per facilitare l'identificazione dello scopo del ruolo. I nomi dei ruoli devono essere univoci all'interno del tuo Account AWS. Non fanno distinzione tra maiuscole e minuscole. Ad esempio, non è possibile creare ruoli denominati sia **`PRODROLE`** che **`prodrole`**. Poiché varie entità possono fare riferimento al ruolo, non è possibile modificare il nome del ruolo dopo averlo creato.
12. Esamina i dettagli del ruolo, quindi scegli Crea ruolo.

Per informazioni su come specificare il ruolo ARN nella funzione Lambda di esempio, vedere. Creazione di una funzione AWS Lambda

Ruolo di esecuzione Lambda

Il ruolo di esecuzione di una funzione Lambda è un IAM ruolo che concede alla funzione il permesso di accedere a AWS servizi e risorse. Questa pagina fornisce informazioni su come creare, visualizzare e gestire il ruolo di esecuzione di una funzione Lambda.

Per impostazione predefinita, Lambda crea un ruolo di esecuzione con autorizzazioni minime quando si crea una nuova funzione Lambda utilizzando. AWS Management Console Per gestire le autorizzazioni concesse nel ruolo di esecuzione, consulta Creazione di un ruolo di esecuzione nella IAM console nella Lambda Developer Guide.

La funzione Lambda di esempio fornita in questo argomento utilizza Secrets Manager per oscurare le credenziali del server di autorizzazione.

Come per qualsiasi IAM ruolo creato, è importante seguire le best practice con privilegi minimi. Durante la fase di sviluppo, a volte potresti concedere autorizzazioni oltre a quelle richieste. Prima di pubblicare la funzione nell'ambiente di produzione, una best practice consiste nel modificare la policy in modo da includere solo le autorizzazioni richieste. Per ulteriori informazioni, consulta Apply least-privelege nella Guida per l'utente. IAM

Consenti HealthLake di attivare la funzione Lambda

Quindi HealthLake puoi invocare la funzione Lambda per tuo conto, devi fare quanto segue:

- È necessario impostare `IdpLambdaArn` uguale alla ARN funzione Lambda che si desidera HealthLake richiamare nella richiesta. `CreateFHIRDatastore`
- È necessaria una politica basata sulle risorse che consenta di HealthLake richiamare la funzione Lambda per conto dell'utente.

Quando HealthLake riceve una FHIR REST API richiesta SMART su un HealthLake data store non FHIR abilitato, ha bisogno delle autorizzazioni per richiamare la funzione Lambda specificata al momento della creazione del data store per tuo conto. Per concedere HealthLake l'accesso, utilizzerai una politica basata sulle risorse. Per ulteriori informazioni sulla creazione di una politica basata sulle risorse per una funzione Lambda, consulta Consentire AWS a un servizio di chiamare una funzione Lambda nella Developer Guide.AWS Lambda

Fornire la concorrenza per la funzione Lambda

> **Important**
>
> HealthLake richiede che il tempo di esecuzione massimo per la funzione Lambda sia inferiore a un secondo (1000 millisecondi).
> Se la funzione Lambda supera il limite di tempo di esecuzione, si ottiene un'eccezione. TimeOut

Per evitare che si verifichi questa eccezione, consigliamo di configurare la concorrenza fornita. Allocando la simultaneità fornita prima di un aumento delle chiamate, è possibile assicurarsi che tutte le richieste siano servite da istanze inizializzate con latenza bassa. Per ulteriori informazioni sulla configurazione della concorrenza fornita, consulta Configuring provisioned concurrency nella Lambda Developer Guide

Per vedere il tempo di esecuzione medio della tua funzione Lambda, utilizza attualmente la pagina Monitoraggio della funzione Lambda sulla console Lambda. Per impostazione predefinita, la console Lambda fornisce un grafico della durata che mostra il tempo medio, minimo e massimo impiegato dal codice della funzione per l'elaborazione di un evento. Per ulteriori informazioni sul monitoraggio delle funzioni Lambda, consulta Monitoring functions in Lambda console Lambda nella Lambda Developer Guide.

Se hai già predisposto la concorrenza per la tua funzione Lambda e desideri monitorarla, consulta Monitoring concurrency nella Lambda Developer Guide.

Creazione di un SMART HealthLake data store non FHIR abilitato

Per utilizzare il FHIR framework SMART on con HealthLake, crea un HealthLake data store con il `IdentityProviderConfiguration` parametro specificato nella reateFHIRDatastore richiesta C. Nel `IdentityProviderConfiguration` parametro si specificano le seguenti informazioni:

- Imposta il AuthorizationStrategyvalore uguale a`SMART_ON_FHIR_V1`.
- Imposta il IdpLambdaArnvalore uguale a quello AWS Lambda che hai creato per gestire la decodifica ARN dei token con il tuo server di autorizzazione.
- Definisci gli elementi di metadati specificati nel server di autorizzazione come JSON blocco. Questi elementi di metadati vengono restituiti nel Discovery Document.
- Facoltativo: Abilita. FineGrainedAuthorizationEnabled `True`Specificare di utilizzare l'autorizzazione granulare fornita da HealthLake

È possibile creare un SMART archivio dati non FHIR abilitato utilizzando AWS Command Line Interface (AWS CLI) o tramite uno dei AWS supportiSDKs. La creazione di un SMART HealthLake data store non FHIR abilitato non è supportata utilizzando la HealthLake console.

Utilizzo di AWS CLI per creare un SMART HealthLake data store non FHIR abilitato

È possibile utilizzare il seguente esempio di codice per creare un SMART HealthLake data store FHIR abilitato utilizzando AWS CLI. Quando si crea un SMART HealthLake data store non FHIR abilitato, è necessario specificare il `identity-provider-configuration`parametro.

Nel `identity-provider-configuration` parametro puoi facoltativamente abilitare l'autorizzazione granulare impostando uguale a. `FineGrainedAuthorizationEnabled` `True` Per ulteriori informazioni sull'autorizzazione granulare, consulta. Utilizzo di autorizzazioni granulari con un archivio dati SMART abilitato FHIR HealthLake L'esempio seguente contiene un carattere speciale \ per indicare interruzioni di riga o come carattere di escape. Questo è per fare chiarezza.

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

In caso di successo si ottiene la seguente JSON risposta:

```
{
```

```
  "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"
}
```

Utilizzo di autorizzazioni granulari con un archivio dati SMART abilitato FHIR HealthLake

Gli ambiti da soli non forniscono la specificità necessaria sui dati a cui un richiedente è autorizzato ad accedere in un data store. L'utilizzo di autorizzazioni granulari consente un livello di specificità più elevato quando si concede l'accesso a un data store abilitato. SMART FHIR HealthLake Per utilizzare l'autorizzazione granulare, imposta `FineGrainedAuthorizationEnabled` uguale a `True` nel parametro della richiesta C. `IdentityProviderConfiguration` reateFHIRDatastore

Se hai abilitato l'autorizzazione granulare, il tuo server di autorizzazione restituisce un `fhirUser` ambito insieme al `id_token` token di accesso. Ciò consente di recuperare le informazioni sull'utente dall'applicazione client. L'applicazione client deve trattare il `fhirUser` reclamo come se si trattasse URI di una FHIR risorsa che rappresenta l'utente corrente. Questo valore può essere `Patient`, `Practitioner` o `RelatedPerson`. La risposta del server di autorizzazione include anche un `user/` ambito che definisce a quali dati l'utente può accedere. Questo utilizza la sintassi definita per gli ambiti relativi agli ambiti specifici FHIR delle risorse:

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

Di seguito sono riportati alcuni esempi di come è possibile utilizzare l'autorizzazione granulare per specificare ulteriormente i tipi di risorse relativi all'accesso ai dati. FHIR

- Quando `fhirUser` è un'autorizzazione `Practitioner` dettagliata determina l'insieme di pazienti a cui l'utente può accedere. L'accesso `fhirUser` è consentito solo ai pazienti per i quali il Paziente si rivolge a un medico `fhirUser` generico.

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

- Quando `fhirUser` è un `Patient` operatore `RelatedPerson` e il paziente a cui si fa riferimento nella richiesta è diverso dall'autorizzazione dettagliata a determinare l'`fhirUser`accesso per il

paziente richiesto. `fhirUser` L'accesso è consentito quando esiste una relazione specificata nella risorsa richiesta. `Patient`

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

Recupero di un Discovery Document SMART su un HealthLake data store FHIR abilitato

Affinché un'applicazione client effettui una FHIR REST richiesta di successo, deve raccogliere i requisiti di autorizzazione definiti nell'archivio HealthLake dati. Non è richiesta alcuna autorizzazione (bearer token) per il successo di questa richiesta.

A tale scopo, effettua una `GET` richiesta e aggiungila `/.well-known/smart-configuration` all'endpoint del data store

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

Ciò restituisce il Discovery Document del HealthLake data store come blob. JSON In esso troverai il `authorization_endpoint` e il `token_endpoint` insieme alle specifiche e alle funzionalità definite nell'archivio HealthLake dati.

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

URLsnecessario per avviare correttamente un'applicazione client

- Endpoint di autorizzazione: URL necessario per autorizzare un'applicazione o un utente client.
- Endpoint token: l'endpoint del server di autorizzazione utilizzato dall'applicazione client per comunicare con essa.

Effettuare una FHIR REST API richiesta su un HealthLake data store SMART abilitato

È possibile effettuare FHIR REST API richieste SMART su un HealthLake data store non FHIR abilitato. L'esempio seguente mostra una richiesta proveniente da un'applicazione client contenente un JWT nell'intestazione di autorizzazione e come Lambda deve decodificare la risposta. Dopo che la richiesta dell'applicazione client è stata autorizzata e autenticata, deve ricevere un token bearer dal server di autorizzazione. Utilizza il token bearer nell'intestazione di autorizzazione quando invii una FHIR REST API richiesta SMART su FHIR un data store abilitato. HealthLake

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

Poiché è stato trovato un token bearer nell'intestazione di autorizzazione e non è stata rilevata alcuna AWS IAM identità, HealthLake richiama la funzione Lambda specificata al momento della creazione SMART del data store FHIR on HealthLake enabled. Quando il token viene decodificato con successo dalla funzione Lambda, ecco un esempio di risposta inviato a. HealthLake

```
{
  "authPayload": {
    "iss": "https://authorization-server-endpoint/oauth2/token", # The issuer
 identifier of the authorization server
    "aud": "https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/
r4/", # Required, data store endpoint
    "iat": 1677115637,  # Identifies the time at which the token was issued
    "nbf": 1677115637,  # Required, the earliest time the JWT would be valid
    "exp": 1997877061,  # Required, the time at which the JWT is no longer valid
    "isAuthorized": "true",  # Required, boolean indicating the request has been
 authorized
    "uid": "100101",  # Unique identifier returned by the auth server
    "scope": "system/*.*" # Required, the scope of the request
  },
  "iamRoleARN": "iam-role-arn" #Required, IAM role to complete the request
}
```

Configurazione delle risorse necessarie per implementare un SMART data store non FHIR conforme

Questo argomento descrive le risorse necessarie per il provisioning dell' AWS account esterno HealthLake, la creazione di un SMART HealthLake data store FHIR abilitato e il modo in cui un'applicazione SMART on FHIR client interagirebbe con un server di autorizzazione e un HealthLake data store.

I passaggi di questo flusso di lavoro definiscono i passaggi di base per la gestione delle FHIR richieste e le risorse necessarie per il loro esito positivo. SMART

In un processo SMART su FHIR richiesta, tre applicazioni interagiscono:

- L'utente finale: in genere, un paziente o un medico che utilizza un'FHIRapplicazione SMART on di terze parti per accedere ai dati in un HealthLake archivio dati.
- L'FHIRapplicazione SMART on (denominata applicazione client): un'applicazione che desidera accedere ai dati presenti nell' HealthLake archivio dati.
- Il server di autorizzazione: un server conforme a OpenID Connect in grado di autenticare gli utenti ed emettere token di accesso.
- L'archivio HealthLake dati: un SMART data store non FHIR abilitato HealthLake che utilizza una funzione Lambda per rispondere alle FHIR REST richieste che forniscono un token portatore.

Affinché queste applicazioni funzionino insieme, è necessario creare le seguenti risorse.

Si consiglia di creare il HealthLake data store SMART on FHIR enabled dopo aver configurato il server di autorizzazione, definito gli ambiti necessari su di esso e creato una AWS Lambda funzione per gestire l'introspezione dei token.

1. Configurazione di un endpoint del server di autorizzazione: server di autorizzazione

Per utilizzare il FHIR framework SMART on è necessario configurare un server di autorizzazione di terze parti in grado di convalidare FHIR REST le richieste effettuate su un data store. Per ulteriori informazioni sulla configurazione di un endpoint del server di autorizzazione con cui funzionare HealthLake, consulta. Requisiti di autenticazione per SMART on FHIR

2. Definisci gli ambiti per controllare chi può accedere a quali dati nel tuo archivio HealthLake dati sul tuo server di autorizzazione: server di autorizzazione

Il FHIR framework SMART on utilizza OAuth gli ambiti per determinare a quali FHIR risorse ha accesso una richiesta autenticata e in che misura. La definizione degli ambiti è un modo per progettare con privilegi minimi. Per ulteriori informazioni sugli ambiti definiti dal FHIR framework SMART on e supportati da, vedi,. HealthLake Supportato SMART sui FHIR OAuth cannocchiali da HealthLake

3. Imposta una AWS Lambda funzione in grado di eseguire l'introspezione dei token: il tuo account AWS

Una FHIR REST richiesta inviata dall'applicazione client SMART su un data store non FHIR abilitato conterrà un JSON Web Token (). JWT Per ulteriori informazioni sulla configurazione di una funzione Lambda in grado di decodificarla e convalidarla, vedi Decodifica a. JWT

4. Crea un HealthLake data store non SMART FHIR abilitato: il tuo account AWS

Per creare un FHIR HealthLake data store SMART on line devi fornire un`IdentityProviderConfiguration`. Per ulteriori informazioni sui `IdentityProviderConfiguration` parametri richiesti in una reateFHIRDatastore richiesta C, consulta Creazione di un SMART HealthLake data store non FHIR abilitato.

In che modo un'applicazione client avvia e richiede dati da un SMART data store FHIR onenable HealthLake

Questa sezione spiega come un'applicazione client viene avviata nel FHIR contesto SMART on ed è in grado di effettuare una FHIR REST richiesta corretta su un data store. HealthLake

1. L'applicazione client effettua una **`GET`** richiesta a Known Uniform Resource Identifier

Un'applicazione client SMART abilitata deve effettuare una `GET` richiesta per trovare gli endpoint di autorizzazione del HealthLake data store. Questa operazione viene eseguita tramite una richiesta Known Uniform Resource Identifier (URI). Per ulteriori informazioni, consulta Fetching a SMART on FHIR enabled HealthLake datastore's Discovery Document.

2. Richiesta di accesso e ambiti

L'applicazione client utilizza l'endpoint di autorizzazione del server di autorizzazione, in modo che l'utente possa accedere. Questo processo autentica l'utente. Gli ambiti vengono utilizzati per definire a quali FHIR risorse del HealthLake data store può accedere un'applicazione client. Per ulteriori informazioni sulla definizione degli ambiti, consulta. Supportato SMART sui FHIR OAuth cannocchiali da HealthLake

3. Token di accesso

Ora che l'utente è stato autenticato, un'applicazione client riceve un token di JWT accesso dal server di autorizzazione. Questo token viene fornito quando l'applicazione client invia una FHIR REST richiesta a HealthLake. Per ulteriori informazioni su come JWT viene decodificato utilizzando una funzione Lambda, vedere. Esecuzione della convalida dei token

4. Effettuare una FHIR REST richiesta SMART su un data store FHIR abilitato HealthLake

Ora, l'applicazione client può inviare una FHIR REST richiesta a un endpoint del HealthLake data store utilizzando il token di accesso fornito dal server di autorizzazione. Per vedere un esempio di FHIR REST richiesta, consultaEffettuare una FHIR REST API richiesta su un HealthLake data store SMART abilitato.

5. Convalida del token di JWT accesso

Per convalidare il token di accesso inviato nella FHIR REST richiesta, usa una funzione Lambda. Per scoprire come creare una funzione Lambda in grado di eseguire l'introspezione dei token, vedere. Creazione di una funzione AWS Lambda

Utilizzo della generazione automatizzata di risorse basata sull'elaborazione del linguaggio naturale (NLP) del tipo di FHIR DocumentReference risorsa in AWS HealthLake

> **Note**
>
> Dopo il 20 febbraio 2023, gli archivi HealthLake dati non utilizzano l'elaborazione integrata del linguaggio naturale (NLP) per impostazione predefinita. Se sei interessato ad attivare questa funzionalità sul tuo data store, consulta il Come posso attivare la funzionalità integrata HealthLake di elaborazione del linguaggio naturale? capitolo Risoluzione dei problemi.
> Se hai attivato l'integrazione di Amazon Comprehend MedicalNLP, quando crei o aggiorni `DocumentReference` le risorse, verranno addebitati degli addebiti sul tuo account. AWS Per ulteriori dettagli, consulta i prezzi.AWS HealthLake
> Amazon Comprehend Medical non è disponibile in Asia Pacifico (Mumbai). HealthLake gli archivi di dati creati nella regione Asia Pacifico (Mumbai) non supportano l'elaborazione integrata del linguaggio naturale (). NLP

HealthLake fornisce automaticamente l'elaborazione integrata del linguaggio naturale (NLP) utilizzando Amazon Comprehend Medical per l'elaborazione di dati non strutturati per i dati archiviati `DocumentReference` nel tipo di risorsa. Per fare ciò, HealthLake chiama Amazon Comprehend `DetectEntities-V2` Medical `InferICD10-CM` `InferRxNorm` API e operations. I risultati vengono aggiunti automaticamente alla `DocumentReference` risorsa come estensione. Quando le operazioni di Amazon Comprehend API Medical rilevano caratteristiche che `SIGN` sono`SYMPTOM`, `DIAGNOSIS` e, viene generato automaticamente `Linkage` un tipo di risorsa. Le nuove risorse relative alle condizioni e all'osservazione vengono create da entità identificate con i tratti di `SIGN``SYMPTOM`, or`DIAGNOSIS`, e sono collegate al documento di origine tramite questa risorsa di collegamento.

Per le risorse generate dall'integrazioneNLP, è possibile effettuare `GET` richieste, ma la ricerca di queste nuove risorse non è supportata.

Per ulteriori informazioni sulla ricerca di queste estensioni utilizzando l' HealthLakeintegrazione con Athena, consulta. Interroga il tuo archivio HealthLake dati utilizzando SQL

Indice

  - Integrazione con le operazioni FHIR REST API
  - Esempi di come le operazioni di Amazon Comprehend API Medical sono integrate in HealthLake
- Parametri di ricerca

In che modo Amazon Comprehend Medical è integrato con HealthLake

HealthLake deduce i dati trovati nel tipo di `DocumentReference` risorsa utilizzando Amazon Comprehend Medical. Amazon Comprehend API Medical `DetectEntities-V2` opera `InferICD10-CM` `InferRxNorm` e rileva le condizioni mediche come caratteristiche. Ogni operazione fornisce informazioni diverse.

> ⚠ Supporto linguistico
>
> Le operazioni di Amazon Comprehend API Medical rilevano le entità mediche solo nei testi in lingua inglese.

- DetectEntities-V2: ispeziona il testo clinico per una varietà di entità mediche e restituisce informazioni specifiche su di esse, come la categoria dell'entità, l'ubicazione e il punteggio di fiducia.
- Inferisci ICD1 0-CM: rileva le condizioni mediche nella cartella clinica di un paziente come entità e collega tali entità agli identificatori concettuali normalizzati nella knowledge base ICD -10-CM del CDC National Center for Health Statistics, previa autorizzazione dell'Organizzazione Mondiale della Sanità (). WHO
- InferRxNorm: rileva i farmaci come entità elencate nella cartella di un paziente e li collega agli identificatori concettuali normalizzati presenti nel database della National Library of Medicine. RxNorm

Le caratteristiche supportate per ogni API operazione sono`SIGN`, e. `SYMPTOM` `DIAGNOSIS` Se vengono rilevate caratteristiche, vengono aggiunte come estensioni FHIR conformi a diverse posizioni del data store. HealthLake

Posizioni in cui vengono aggiunte le estensioni.

- `DocumentReference`: I risultati delle operazioni di Amazon Comprehend API Medical vengono aggiunti come `extension` un unico documento presente all'interno DocumentReference del tipo di risorsa. I risultati dell'estensione sono divisi in due gruppi. Puoi trovarli nei risultati in base ai loro`URL`.
  - `http://healthlake.amazonaws.com/system-generated-resources/`
    - Si tratta di tipi di risorse che sono stati creati o aggiunti da HealthLake.
  - `http://healthlake.amazonaws.com/aws-cm/`
    - Dove l'output non elaborato delle operazioni di Amazon Comprehend API Medical viene aggiunto al HealthLake tuo data store.
- `Linkage`: Questo tipo di risorsa viene aggiunto o creato come risultato dell'integrazioneNLP. Una `GET` richiesta su uno specifico `Linkage` restituisce un elenco di risorse collegate. Per identificare se `Linkage` è stato aggiunto un da HealthLake, cerca la coppia `"tag": [{"display": "SYSTEM_GENERATED"}]` chiave-valore aggiunta. Per ulteriori informazioni sulle FHIR specifiche del collegamento, consulta Tipo di risorsa: collegamento nell'indice della documentazione. FHIR
- FHIRtipi di risorse generati come risultato delle operazioni di Amazon Comprehend API Medical.
  - `Observation`: include i risultati delle DetectEntities operazioni di Amazon Comprehend API Medical -V2 e ICD1 Infer 0-CM quando le caratteristiche sono o. `SIGN` `SYMPTOM`
  - `Condition`: include i risultati delle DetectEntities operazioni Amazon Comprehend API Medical -V2 e ICD1 Infer 0-CM quando le caratteristiche sono presenti. `DIAGNOSIS`
  - `MedicationStatement`: Sono stati aggiunti i risultati dell' InferRxNorm operazione Amazon Comprehend API Medical.

Integrazione con le operazioni FHIR REST API

Per impostazione predefinita, le caratteristiche rilevate dalle API operazioni di Amazon Comprehend Medical non vengono restituite quando si effettua una richiesta. `GET`

Per visualizzare i risultati delle NLP operazioni integrate per questi tipi di risorse, devi specificare un valore noto. `ID`

- `Linkage`
- `Observation`
- `Condition`

- `MedicationStatement`

I risultati delle NLP operazioni integrate al di fuori del tipo di DocumentReference risorsa sono disponibili solo utilizzando una `GET` richiesta in cui `ID` si sa che il dato specificato contiene risultati delle operazioni di Amazon Comprehend API Medical.

Esempi di come le operazioni di Amazon Comprehend API Medical sono integrate in HealthLake

Esempio 1: cartella del paziente inserita in un archivio dati HealthLake

Ecco un esempio di nota clinica basata sull'incontro di un paziente con un medico.

> **⚠ Dati sintetici**
>
> Il testo in questo esempio è contenuto sintetico e non contiene informazioni sanitarie personali (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)
```

Come promemoria, queste informazioni sono codificate in formato base64 nella risorsa. DocumentReference Quando questo documento viene inserito HealthLake e le operazioni di Amazon Comprehend API Medical sono complete, per vedere i risultati, puoi iniziare con `GET` la richiesta sul `DocumentReference` tipo di risorsa.

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

Quando le operazioni di Amazon Comprehend API Medical hanno esito positivo, cerca queste coppie chiave-valore all'interno `extension` del link seguente `"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."
```

```
}
```

Le seguenti schede mostrano come la cartella clinica ingerita viene riportata nel tuo archivio HealthLake dati in base al tipo di risorsa.

DocumentReference

Per visualizzare i risultati per un singolo tipo di `DocumentReference` risorsa, effettua una `GET` richiesta in cui viene fornita `id` una risorsa specifica.

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

In caso di successo, si ottiene un codice di `200` HTTP risposta e la seguente JSON risposta (che è stata troncata per maggiore chiarezza).

Ecco la parte. `http://healthlake.amazonaws.com/system-generated-resources/` Puoi vedere che ne `Linkage/`*`e366d29f-2c22-4c19-866e-09603937935a`* è stata aggiunta una nuova. Puoi anche vedere dove sono HealthLake stati aggiunti risultati basati sull'inferenza a tipi di `Condition` risorse specifici`Observation`.

Per vedere come sono stati modificati questi tipi di risorse, scegli le schede correlate.

```
{
  "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

Per visualizzare i risultati per un singolo tipo di `Linkage` risorsa, effettua una `GET` richiesta in cui `ID` viene fornita una risorsa specifica.

```
GET https://https://healthlake.your-region.amazonaws.com/
datastore/your-datastore-id/r4/eeb8005725ae22b35b4edbdc68cf2dfd/r4/
Linkage/e366d29f-2c22-4c19-866e-09603937935a
```

In caso di successo, si ottiene un codice di `200` HTTP risposta e la seguente risposta troncataJSON.

La risposta contiene l'elemento. `item` In essa, la coppia chiave-valore `"type": "source"` indica la `DocumentReference` voce specifica utilizzata per modificare la coppia chiave-valore `Condition` ed è `Observations` elencata sotto la coppia `"type": "alternate"` chiave-valore.

Sono inoltre visibili l'*`meta`*elemento e la corrispondente coppia chiave-valore, che indica che queste risorse sono state *`"tag": [{"display": "SYSTEM_GENERATED"}]`* create da. 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

Per visualizzare i risultati per un singolo tipo di `Observation` risorsa, effettua una `GET` richiesta in cui viene `ID` fornita una risorsa specifica.

```
GET https://https://healthlake.your-region.amazonaws.com/
datastore/your-datastore-id/r4/eeb8005725ae22b35b4edbdc68cf2dfd/r4/
Observation/e366d29f-2c22-4c19-866e-09603937935a
```

I risultati delle operazioni di Amazon Comprehend API Medical sono stati modificati in base ai seguenti elementicode:`meta`,, `modifierExtension` e.

`code`

Un elemento di tipo`CodeableConcept`. Per ulteriori informazioni, CodeableConceptconsulta l'Indice della FHIR documentazione.

HealthLake aggiunge le seguenti tre coppie chiave-valore.

- `"system": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/"`: dove si URL riferisce a una specifica operazione di Amazon Comprehend API Medical. In questo caso, Infer 0CM. ICD1

- `"code": "A52.06"`: `A52.06` Dov'è il codice ICD -10-CM che identifica il concetto trovato nella knowledge base dei Centers for Disease Control.
- `"display": "Other syphilitic heart involvement"`: `"Other syphilitic heart involvement"` Dov'è la descrizione estesa del codice ICD -10-CM nell'ontologia.

La seguente risposta troncata JSON contiene solo l'elemento. `code`

```
"code": {
  "coding":
  [
    {
   "system": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/",
   "code": "A52.06",
   "display": "Other syphilitic heart involvement"
    }
  ],
  "text": "Other syphilitic heart involvement"
 }
```

Per comprendere la fiducia del modello nella correttezza del codice ICD -10-CM assegnato, utilizzate l'elemento. `modifierExtension`

`meta`

L'`meta`elemento contiene metadati che indicano se l'`code`elemento contiene dettagli che sono stati aggiunti dalle operazioni di Amazon Comprehend Medical. API

La seguente JSON risposta troncata contiene solo l'elemento. `meta`

```
"meta": {
  "lastUpdated": "2022-10-21T19:38:30.879Z",
  "tag": [{
   "display": "SYSTEM_GENERATED"
  }]
 }
```

`modifierExtension`

L'`modifierExtension`elemento contiene ulteriori dettagli sul livello di confidenza dei codici assegnati presenti nell'elemento. `code` Dispone inoltre di coppie chiave-valore che forniscono un

collegamento all'originale DocumentReference utilizzato per generare i risultati e al tipo di risorsa Linkage correlato.

Per ogni `coding` elemento aggiunto, vedrai un `entity-score` e un `entity-Concept-Score` aggiunto a. modifierExtension Per ogni valore nella coppia chiave-valore, viene visualizzato un punteggio. Infatti`entity-score`, questo punteggio è il livello di fiducia che Amazon Comprehend Medical ha nell'accuratezza del rilevamento. Infatti`entity-Concept-Score`, questo punteggio è il livello di fiducia di Amazon Comprehend Medical nel fatto che l'entità sia accuratamente collegata a ICD un concetto di -10 CM.

La seguente risposta troncata JSON contiene solo l'elemento. `modifierExtension`

```
"modifierExtension": [{
   "url": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/aws-cm-icd10-entity-
score",
   "valueDecimal": 0.45005733
  },
  {
   "url": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/aws-cm-icd10-entity-
Concept-Score",
   "valueDecimal": 0.1111792
  },
  {
   "url": "http://healthlake.amazonaws.com/system-generated-linkage",
   "valueReference": {
    "reference": "Linkage/e366d29f-2c22-4c19-866e-09603937935a"
   }
  },
  {
   "url": "http://healthlake.amazonaws.com/source-document-reference",
   "valueReference": {
    "reference": "DocumentReference/0e938f03-da7f-4178-acd8-eea9586c46ed"
   }
  }
 ]
```

Risposta completa JSON

```
{
  "subject": {
   "reference": "Patient/0679b7b7-937d-488a-b48d-6315b8e7003b"
  },
```

```
  "resourceType": "Observation",
  "status": "unknown",
  "code": {
   "coding": [{
    "system": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/",
    "code": "A52.06",
    "display": "Other syphilitic heart involvement"
   }],
   "text": "Other syphilitic heart involvement"
  },
  "meta": {
   "lastUpdated": "2022-10-21T19:38:30.879Z",
   "tag": [{
    "display": "SYSTEM_GENERATED"
   }]
  },
  "modifierExtension": [{
    "url": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/aws-cm-icd10-entity-
score",
    "valueDecimal": 0.45005733
   },
   {
    "url": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/aws-cm-icd10-entity-
Concept-Score",
    "valueDecimal": 0.1111792
   },
   {
    "url": "http://healthlake.amazonaws.com/system-generated-linkage",
    "valueReference": {
     "reference": "Linkage/e366d29f-2c22-4c19-866e-09603937935a"
    }
   },
   {
    "url": "http://healthlake.amazonaws.com/source-document-reference",
    "valueReference": {
     "reference": "DocumentReference/0e938f03-da7f-4178-acd8-eea9586c46ed"
    }
   }
  ],
  "id": "7e88c7c5-21a5-4dd7-8fc2-a02474fba583"
 }
```

Condition

Per visualizzare i risultati per un singolo tipo di `Condition` risorsa, effettua una `GET` richiesta in cui viene fornita una risorsa specifica. `ID`

```
GET https://https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/eeb8005725ae22b35b4edbdc68cf2dfd/r4/Condition/b06d343d-ddb8-4f36-82cb-853fcd434dfd
```

I risultati delle operazioni di Amazon Comprehend API Medical sono stati modificati in base ai seguenti elementicode:`meta`,, `modifierExtension` e.

`code`

Un elemento di tipo`CodeableConcept`. Per ulteriori informazioni, CodeableConceptconsulta l'Indice della FHIR documentazione.

HealthLake aggiunge le seguenti tre coppie chiave-valore.

- `"system": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/"`: dove si URL riferisce a una specifica operazione di Amazon Comprehend API Medical. In questo caso, Infer 0CM. ICD1
- `"code": "I70.0"`: `A52.06` Dov'è il codice ICD -10-CM che identifica il concetto trovato nella knowledge base dei Centers for Disease Control.
- `"display": "Atherosclerosis of aorta"`: `"Other syphilitic heart involvement"` Dov'è la descrizione estesa del codice ICD -10-CM nell'ontologia.

La seguente risposta troncata JSON contiene solo l'elemento. `code`

```
"code": {
  "coding":
  [
    {
   "system": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/",
   "code": "I70.0",
   "display": "Atherosclerosis of aorta"
    }
  ],
  "text": "Atherosclerosis of aorta"
 }
```

Per comprendere la fiducia del modello nella correttezza del codice ICD -10-CM assegnato, utilizzate l'elemento. `modifierExtension`

`meta`

L'`meta`elemento contiene metadati che indicano se l'`code`elemento contiene dettagli che sono stati aggiunti dalle operazioni di Amazon Comprehend Medical. API

La seguente JSON risposta troncata contiene solo l'elemento. `meta`

```
"meta": {
  "lastUpdated": "2022-10-21T19:38:30.877Z",
  "tag": [{
   "display": "SYSTEM_GENERATED"
  }]
 }
```

`modifierExtension`

L'`modifierExtension`elemento contiene ulteriori dettagli sul livello di confidenza dei codici assegnati presenti nell'elemento. `code` Dispone inoltre di coppie chiave-valore che forniscono un collegamento all'originale DocumentReference utilizzato per generare i risultati e al tipo di risorsa Linkage correlato.

Per ogni `coding` elemento aggiunto, vedrai un `entity-score` e un `entity-Concept-Score` aggiunto a. modifierExtension Per ogni valore nella coppia chiave-valore, viene visualizzato un punteggio. Infatti`entity-score`, questo punteggio è il livello di fiducia che Amazon Comprehend Medical ha nell'accuratezza del rilevamento. Infatti`entity-Concept-Score`, questo punteggio è il livello di fiducia di Amazon Comprehend Medical nel fatto che l'entità sia accuratamente collegata a ICD un concetto di -10 CM.

La seguente risposta troncata JSON contiene solo l'elemento. `modifierExtension`

```
"modifierExtension": [{
   "url": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/aws-cm-icd10-entity-
score",
   "valueDecimal": 0.94417894
  },
  {
   "url": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/aws-cm-icd10-entity-
Concept-Score",
```

```
   "valueDecimal": 0.8458298
  },
  {
   "url": "http://healthlake.amazonaws.com/system-generated-linkage",
   "valueReference": {
    "reference": "Linkage/e366d29f-2c22-4c19-866e-09603937935a"
   }
  },
  {
   "url": "http://healthlake.amazonaws.com/source-document-reference",
   "valueReference": {
    "reference": "DocumentReference/0e938f03-da7f-4178-acd8-eea9586c46ed"
   }
  }
 ]
```

Risposta completa JSON

```
{
  "subject": {
   "reference": "Patient/0679b7b7-937d-488a-b48d-6315b8e7003b"
  },
  "resourceType": "Condition",
  "code": {
   "coding": [{
    "system": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/",
    "code": "I70.0",
    "display": "Atherosclerosis of aorta"
   }],
   "text": "Atherosclerosis of aorta"
  },
  "meta": {
   "lastUpdated": "2022-10-21T19:38:30.877Z",
   "tag": [{
    "display": "SYSTEM_GENERATED"
   }]
  },
  "modifierExtension": [{
    "url": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/aws-cm-icd10-entity-
score",
    "valueDecimal": 0.94417894
   },
   {
```

```
    "url": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/aws-cm-icd10-entity-
Concept-Score",
    "valueDecimal": 0.8458298
   },
   {
    "url": "http://healthlake.amazonaws.com/system-generated-linkage",
    "valueReference": {
     "reference": "Linkage/e366d29f-2c22-4c19-866e-09603937935a"
    }
   },
   {
    "url": "http://healthlake.amazonaws.com/source-document-reference",
    "valueReference": {
     "reference": "DocumentReference/0e938f03-da7f-4178-acd8-eea9586c46ed"
    }
   }
  ],
  "id": "b06d343d-ddb8-4f36-82cb-853fcd434dfd"
 }
```


Esempio 2: A **`DocumentReference`** che contiene un tipo di MedicationStatement risorsa

Ecco un esempio di nota clinica basata sull'incontro di un paziente con un medico.

> ⚠ Dati sintetici
>
> Il testo in questo esempio è contenuto sintetico e non contiene informazioni sanitarie personali (PHI).

```
Tom is not prescribed Advil
```

Le schede seguenti mostrano come la cartella clinica ingerita viene riportata nell'archivio HealthLake dati in base al tipo di risorsa.

DocumentReference

Per visualizzare i risultati per un singolo tipo di `DocumentReference` risorsa, effettua una `GET` richiesta in cui viene fornita `ID` una risorsa specifica.

```
GET https://https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/eeb8005725ae22b35b4edbdc68cf2dfd/r4/DocumentReference/c549125d-a218-421f-b8bf-23614c5e796c
```

In caso di successo, si ottiene un codice di `200` HTTP risposta e la seguente risposta troncataJSON.

La coppia chiave-valore indica che i `"url": "http://healthlake.amazonaws.com/system-generated-resources/"` tipi di risorse al suo interno `extension` sono stati aggiunti dalle operazioni di Amazon API Comprehend Medical. Puoi vedere il nuovo tipo di `Linkage` risorsa e più risorse. `MedicationStatement`

```
"extension": [{
   "extension": [{
     "url": "http://healthlake.amazonaws.com/linkage",
     "valueReference": {
      "reference": "Linkage/394bb244-177b-4409-8657-26b20ed56dd7"
     }
    },
    {
     "url": "http://healthlake.amazonaws.com/nlp-entity",
     "valueReference": {
      "reference": "MedicationStatement/cbf6af10-b0b9-451c-bdde-99611e3498a8"
     }
    },
    {
     "url": "http://healthlake.amazonaws.com/nlp-entity",
     "valueReference": {
      "reference": "MedicationStatement/9a89b0d3-6681-45ca-9926-27951edce5c7"
     }
    },
    {
     "url": "http://healthlake.amazonaws.com/nlp-entity",
     "valueReference": {
      "reference": "MedicationStatement/4a01f6c8-5f3a-4122-80ab-405312f96aa2"
     }
    },
    {
     "url": "http://healthlake.amazonaws.com/nlp-entity",
     "valueReference": {
      "reference": "MedicationStatement/fbfb77d8-70cf-4579-b4c0-d6fe3c01656b"
     }
```

```
    },
    {
     "url": "http://healthlake.amazonaws.com/nlp-entity",
     "valueReference": {
      "reference": "MedicationStatement/1340c9ce-9c48-4bf9-9b2f-d0ab027f5e0b"
     }
    }
   ],
   "url": "http://healthlake.amazonaws.com/system-generated-resources/"
  }
```

Linkage

Per visualizzare i risultati per un singolo tipo di `Linkage` risorsa, effettua una `GET` richiesta in cui viene fornita una risorsa specifica. `ID`

```
GET https://https://healthlake.your-region.amazonaws.com/
datastore/your-datastore-id/r4/eeb8005725ae22b35b4edbdc68cf2dfd/r4/
Linkage/394bb244-177b-4409-8657-26b20ed56dd7
```

In caso di successo, si ottiene un codice di `200` HTTP risposta e la seguente JSON risposta.

La risposta contiene l'`item`elemento. In essa, la coppia chiave-valore `"type": "source"` indica la `DocumentReference` voce specifica utilizzata per modificare i tipi di `MedicationStatement` risorse.

È inoltre possibile visualizzare l'`meta`elemento e la corrispondente coppia chiave-valore`"tag": [{"display": "SYSTEM_GENERATED"}]`, a indicare che queste risorse sono state create da. 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

Per visualizzare i risultati per un singolo tipo di `MedicationStatement` risorsa, effettua una `GET` richiesta in cui viene `ID` fornita una risorsa specifica.

```
GET https://https://healthlake.your-region.amazonaws.com/
datastore/your-datastore-id/r4/eeb8005725ae22b35b4edbdc68cf2dfd/r4/
MedicationStatement/9a89b0d3-6681-45ca-9926-27951edce5c7
```

Il tipo di MedicationStatement risorsa è il luogo in cui si trovano i risultati dell'operazione Amazon Comprehend InferRxNorm API Medical. I risultati vengono modificati in base ai seguenti elementi:`medicationCodeableConcept`,`meta`, e`modifierExtension`.

medicationCodeableConcept

Un elemento di tipo`CodeableConcept`. Per ulteriori informazioni, CodeableConceptconsulta l'Indice della FHIR documentazione.

HealthLake aggiunge le seguenti tre coppie chiave-valore.

- `"system": ""http://healthlake.amazonaws.com/aws-cm/infer-rxnorm/`: dove si URL riferisce a una specifica operazione di Amazon Comprehend API Medical. In questo caso, InferRxNorm.
- `"code": "731533"`: `731533` Dov'è un RxNorm concept ID, noto anche come RxCUI.
- `"display": "ibuprofen 200 MG Oral Capsule [Advil]"`: `ibuprofen 200 MG Oral Capsule [Advil]` Dov'è la descrizione del RxNorm concetto.

La seguente JSON risposta troncata contiene solo l'elemento. `MedicationStatement`

```
"medicationCodeableConcept": {
  "coding": [
   {
    "system": "http://healthlake.amazonaws.com/aws-cm/infer-rxnorm/",
    "code": "731533",
    "display": "ibuprofen 200 MG Oral Capsule [Advil]"
   }
  ]
 }
```

meta

L'`meta`elemento contiene metadati che indicano se l'`code`elemento contiene dettagli che sono stati aggiunti dalle operazioni di Amazon Comprehend Medical. API

La seguente JSON risposta troncata contiene solo l'elemento. `meta`

```
"meta": {
  "lastUpdated": "2022-10-24T20:05:02.800Z",
  "tag": [
   {
    "display": "SYSTEM_GENERATED"
   }
  ]
 }
```

`modifierExtension`

L'`modifierExtension`elemento contiene coppie chiave-valore che forniscono un collegamento all'originale DocumentReference utilizzato per generare i risultati e al relativo tipo di risorsa Linkage.

```
"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"
    }
   }
 ]
```

Parametri di ricerca

La tabella seguente elenca gli attributi ricercabili per la medicina NLP integrata.

Parametri di ricerca

Parametri di ricerca	Trova le corrispondenze per
detectEntities-entità-categoria	Categoria di entità all'interno della DetectEntities sottoestensione all'interno dell'estensione CM AWS
detectEntities-entità-testo	Testo dell'entità all'interno della DetectEntities sottoestensione all'interno dell'estensione CM AWS
detectEntities-tipo di entità	Tipo di entità all'interno della DetectEntities sottoeste nsione all'interno dell'estensione CM AWS
detectEntities-entity-score	Entity Score all'interno della DetectEntities sottoeste nsione all'interno dell'estensione CM AWS
infer-icd10 cm-entity-text	Testo dell'entità all'interno della sottoestensione Infer ICD1 0CM all'interno dell'estensione CM AWS
infer-icd10 cm-entity-score	Entity Score all'interno della sottoestensione Infer ICD1 0CM all'interno dell'estensione CM AWS
infer-icd10 cm-entity-concept-code	Entity Concept Code all'interno della sottoestensione Infer ICD1 0CM all'interno dell'estensione CM AWS
infer-icd10 cm-entity-concept-description	Descrizione del concetto di entità all'interno della sottoestensione Infer ICD1 0CM all'interno dell'este nsione CM AWS
infer-icd10 cm-entity-concept-score	Entity Concept Score all'interno della sottoestensione Infer ICD1 0CM all'interno dell'estensione CM AWS
infer-rxnorm-entity-score	Entity Score all'interno della InferRxNorm sottoeste nsione all'interno dell'estensione CM AWS
infer-rxnorm-entity-text	Testo dell'entità all'interno della InferRxNorm sottoestensione all'interno dell'AWSestensione CM
infer-rxnorm-entity-concept-codice	Entity Concept Code all'interno della InferRxNorm sottoestensione all'interno dell'AWSestensione CM

Parametri di ricerca	Trova le corrispondenze per
infer-rxnorm-entity-concept-descrizione	Descrizione del concetto di entità all'interno della InferRxNorm sottoestensione all'interno dell'AWSe stensione CM
infer-rxnorm-entity-concept-punteggio	Entity Concept Score all'interno della InferRxNorm sottoestensione all'interno dell'AWSestensione CM

Per soddisfare i criteri in cui `EntityText` e `EntityCategory` fanno parte della stessa entità, HealthLake fornisce una ricerca speciale. La tabella seguente descrive i parametri di ricerca speciali supportati da HealthLake.

Parametri di ricerca

Parametri di ricerca	Riscontri restituiti
detectEntities-entity-text-category	Se è presente almeno un'entità nella DetectEntities sottoestensione che corrisponde sia a che entityText . entityCategory
detectEntities-entity-type-score	Se è presente almeno un'entità nella DetectEntities sottoestensione che corrisponde sia a che. entityType entityScore
detectEntities-entity-text-score	Se è presente almeno un'entità nella DetectEntities sottoestensione che corrisponde sia a che. entityText entityScore
detectEntities-entity-text-type	Se è presente almeno un'entità nella DetectEntities sottoestensione che corrisponde sia a che. entityText entityType
detectEntities-entity-category-score	Se è presente almeno un'entità che corrisponde sia a che entityCategory . entityScore
codice infer-icd10 cm-entity-text-concept	Se c'è almeno un'entità nella sottoestensione Infer ICD1 0CM che corrisponde a entityText e ce n'è

Parametri di ricerca	Riscontri restituiti
	almeno una conceptCode per quell'entità che corrisponde al codice.
infer-icd10 -punteggio cm-entity-text-concept	Se c'è almeno un'entità nella sottoestensione Infer ICD1 0CM che corrisponde a entityText e ce n'è almeno una conceptScore per quell'entità che corrisponde al punteggio.
infer-icd10 -concept-score cm-entity-concept-description	Se c'è almeno un concetto all'interno dell'entità nella sottoestensione Infer ICD1 0CM che corrisponde alla descrizione del concetto e al. conceptScore
infer-rxnorm-entity-text-codice concettuale	Se c'è almeno un'entità nella InferRxNorm sottoestensione che corrisponde a entityText e ce n'è almeno una conceptCode per quell'entità che corrisponde al codice.
infer-rxnorm-entity-text-partitura concettuale	Se c'è almeno un'entità nella InferRxNorm sottoestensione che corrisponde a entityText e ce n'è almeno una conceptScore per quell'entità che corrisponde al punteggio.
infer-rxnorm-entity-concept-description-concept-score	Se c'è almeno un concetto all'interno dell'entità nella InferRxNorm sottoestensione che corrisponde alla descrizione del concetto e al. conceptScore

Sicurezza in AWS HealthLake

La sicurezza del cloud AWS è la massima priorità. In qualità di AWS cliente, puoi beneficiare di un data center e di un'architettura di rete progettati per soddisfare i requisiti delle organizzazioni più sensibili alla sicurezza.

La sicurezza è una responsabilità condivisa tra AWS te e te. Il modello di responsabilità condivisa descrive questo aspetto come sicurezza del cloud e sicurezza nel cloud:

- La sicurezza del cloud AWS è responsabile della protezione dell'infrastruttura che gestisce AWS i servizi nel AWS cloud. AWS ti fornisce anche servizi che puoi utilizzare in modo sicuro. I revisori esterni testano e verificano regolarmente l'efficacia della nostra sicurezza nell'ambito dei AWS Programmi di AWS conformità dei Programmi di conformità dei di . Per ulteriori informazioni sui programmi di conformità applicabili HealthLake, consulta AWSServizi nell'ambito del programma di conformità AWS .
- Sicurezza nel cloud: la tua responsabilità è determinata dal AWS servizio che utilizzi. Sei anche responsabile di altri fattori, tra cui la riservatezza dei dati, i requisiti della tua azienda e le leggi e normative vigenti.

Questa documentazione ti aiuta a capire come applicare il modello di responsabilità condivisa durante l'utilizzo HealthLake. I seguenti argomenti mostrano come eseguire la configurazione HealthLake per soddisfare gli obiettivi di sicurezza e conformità. Imparerai anche a utilizzare altri AWS servizi che ti aiutano a monitorare e proteggere HealthLake le tue risorse.

Argomenti

Protezione dei dati in AWS HealthLake

Il modello di responsabilità AWS condivisa modello di di si applica alla protezione dei dati in AWS HealthLake. Come descritto in questo modello, AWS è responsabile della protezione dell'infrastruttura globale che gestisce tutti i Cloud AWS. L'utente è responsabile del controllo dei contenuti ospitati su questa infrastruttura. L'utente è inoltre responsabile della configurazione della protezione e delle attività di gestione per i Servizi AWS utilizzati. Per ulteriori informazioni sulla privacy dei dati, consulta la sezione Privacy dei dati FAQ. Per informazioni sulla protezione dei dati in Europa, consulta il Modello di responsabilitàAWS condivisa e GDPR il post sul blog sulla AWS sicurezza.

Ai fini della protezione dei dati, ti consigliamo di proteggere Account AWS le credenziali e di configurare i singoli utenti con AWS IAM Identity Center o AWS Identity and Access Management (IAM). In tal modo, a ogni utente verranno assegnate solo le autorizzazioni necessarie per svolgere i suoi compiti. Ti suggeriamo, inoltre, di proteggere i dati nei seguenti modi:

- Utilizza l'autenticazione a più fattori (MFA) con ogni account.
- UsaSSL/TLSper comunicare con AWS le risorse. Richiediamo TLS 1.2 e consigliamo TLS 1.3.
- Configurazione API e registrazione delle attività degli utenti con AWS CloudTrail. Per informazioni sull'uso dei CloudTrail percorsi per registrare AWS le attività, consulta Lavorare con i CloudTrail percorsi nella Guida per l'AWS CloudTrail utente.
- Utilizza soluzioni di AWS crittografia, insieme a tutti i controlli di sicurezza predefiniti all'interno Servizi AWS.
- Utilizza i servizi di sicurezza gestiti avanzati, come Amazon Macie, che aiutano a individuare e proteggere i dati sensibili archiviati in Amazon S3.
- Se hai bisogno di FIPS 140-3 moduli crittografici convalidati per accedere AWS tramite un'interfaccia a riga di comando o unAPI, usa un endpoint. FIPS Per ulteriori informazioni sugli FIPS endpoint disponibili, vedere Federal Information Processing Standard () 140-3. FIPS

Ti consigliamo vivamente di non inserire mai informazioni riservate o sensibili, ad esempio gli indirizzi e-mail dei clienti, nei tag o nei campi di testo in formato libero, ad esempio nel campo Nome. Ciò include quando lavori HealthLake o Servizi AWS utilizzi in altro modo la console, API AWS CLI, o. AWS SDKs I dati inseriti nei tag o nei campi di testo in formato libero utilizzati per i nomi possono essere utilizzati per i la fatturazione o i log di diagnostica. Se fornisci un URL a un server esterno, ti consigliamo vivamente di non includere le informazioni sulle credenziali URL per convalidare la tua richiesta a quel server.

Crittografia presso for REST AWS HealthLake

HealthLake fornisce la crittografia di default per proteggere i dati sensibili dei clienti archiviati utilizzando una AWS chiave Key Management Service (AWSKMS) di proprietà del servizio. Sono supportate anche KMS le chiavi gestite dal cliente e sono necessarie sia per l'importazione che per l'esportazione di file da un data store. Per ulteriori informazioni sulla KMS chiave gestita dal cliente, consulta Amazon Key Management Service. I clienti possono scegliere una KMS chiave AWS di proprietà o una KMS chiave gestita dal cliente durante la creazione di un data store. La configurazione di crittografia non può essere modificata dopo la creazione di un data store. Se un data store utilizza una KMS chiave AWS di proprietà, la chiave specifica utilizzata per la crittografia verrà contrassegnata come inattiva AWS_OWNED_KMS_KEY e non verrà visualizzata.

AWSchiave di proprietà KMS

HealthLake utilizza queste chiavi per impostazione predefinita per crittografare automaticamente le informazioni potenzialmente sensibili come i dati personali identificabili o i dati Private Health Information (PHI) archiviati. AWSKMSle chiavi di proprietà non sono archiviate nel tuo account. Fanno parte di una raccolta di KMS chiavi che AWS possiede e gestisce per l'utilizzo in più AWS account. AWSi servizi possono utilizzare KMS chiavi AWS di proprietà per proteggere i dati. Non puoi visualizzare, gestire, utilizzare KMS chiavi AWS di proprietà o controllarne l'utilizzo. Tuttavia, non è necessario eseguire alcuna operazione o modificare alcun programma per proteggere le chiavi che crittografano i dati.

Non ti viene addebitato un canone mensile o un canone di utilizzo se utilizzi KMS chiavi AWS di proprietà e queste non vengono conteggiate nelle AWS KMS quote del tuo account. Per ulteriori informazioni, consulta Chiavi AWS possedute.

KMSChiavi gestite dal cliente

HealthLake supporta l'uso di una KMS chiave simmetrica gestita dal cliente che l'utente può creare, possedere e gestire per aggiungere un secondo livello di crittografia rispetto alla crittografia di AWS proprietà esistente. Avendo il pieno controllo di questo livello di crittografia, è possibile eseguire operazioni quali:

- Stabilire e mantenere politiche, IAM politiche e sovvenzioni chiave
- Ruotare i materiali crittografici delle chiavi
- Abilitare e disabilitare le policy delle chiavi
- Aggiungere tag

- Creare alias delle chiavi
- Pianificare l'eliminazione delle chiavi

Puoi anche utilizzarlo CloudTrail per tenere traccia delle richieste HealthLake inviate a per tuo AWS KMS conto. AWS KMS Si applicano costi aggiuntivi. Per ulteriori informazioni, consulta la sezione Customer Owned Keys.

Creazione di una chiave gestita dal cliente

È possibile creare una chiave simmetrica gestita dal cliente utilizzando la console di AWS gestione o il. AWS KMS APIs

Segui i passaggi per la creazione di una chiave simmetrica gestita dal cliente nella AWS Key Management Service Developer Guide.

Le policy della chiave controllano l'accesso alla chiave gestita dal cliente. Ogni chiave gestita dal cliente deve avere esattamente una policy della chiave, che contiene istruzioni che determinano chi può usare la chiave e come la possono usare. Quando crei la chiave gestita dal cliente, puoi specificare una policy della chiave. Per ulteriori informazioni, consulta Gestire l'accesso alle chiavi gestite dal cliente nella AWS Key Management Service Developer Guide.

Per utilizzare la chiave gestita dal cliente con HealthLake le tue risorse, kms: CreateGrant le operazioni devono essere consentite nella politica chiave. Ciò aggiunge una concessione a una chiave gestita dal cliente che controlla l'accesso a una KMS chiave specificata, che consente a un utente l'accesso alle operazioni kms:grant richieste. HealthLake Per ulteriori informazioni, vedere Utilizzo delle sovvenzioni.

Per utilizzare la KMS chiave gestita dal cliente con HealthLake le tue risorse, nella politica chiave devono essere consentite le seguenti API operazioni:

- kms: CreateGrant aggiunge le concessioni a una KMS chiave specifica gestita dal cliente che consente l'accesso alle operazioni di concessione.
- kms: DescribeKey fornisce i dettagli chiave gestiti dal cliente necessari per convalidare la chiave. Questo è necessario per tutte le operazioni.
- kms: GenerateDataKey fornisce l'accesso alle risorse di crittografia a riposo per tutte le operazioni di scrittura.
- KMS:Decrypt fornisce l'accesso alle operazioni di lettura o ricerca per risorse crittografate.

Di seguito è riportato un esempio di dichiarazione politica che consente a un utente di creare e interagire con un archivio dati in AWS HealthLake cui è crittografato da tale chiave:

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

IAMAutorizzazioni richieste per l'utilizzo di una chiave gestita KMS dal cliente

Quando si crea un archivio dati con AWS KMS crittografia abilitata utilizzando una KMS chiave gestita dal cliente, sono necessarie le autorizzazioni sia per la policy chiave che per la IAM politica per l'utente o il ruolo che crea il HealthLake data store.

È possibile utilizzare la chiave kms: ViaService condition per limitare l'uso della KMS chiave solo alle richieste che provengono da. HealthLake

Per ulteriori informazioni sulle politiche chiave, vedere Enabling IAM policies nella AWS Key Management Service Developer Guide.

L'IAMutente, il IAM ruolo o l'AWSaccount che crea i tuoi repository deve disporre delle autorizzazioni kms:CreateGrant, kms:GenerateDataKey, e più kms: DescribeKey le autorizzazioni necessarie. HealthLake

Come utilizza le sovvenzioni in HealthLake AWS KMS

HealthLake richiede una concessione per utilizzare la KMS chiave gestita dal cliente. Quando crei un Data Store crittografato con una KMS chiave gestita dal cliente, HealthLake crea una concessione per tuo conto inviando una CreateGrantrichiesta a AWSKMS. Le sovvenzioni AWS KMS vengono utilizzate per HealthLake consentire l'accesso a una KMS chiave in un account cliente.

Le sovvenzioni HealthLake create per tuo conto non devono essere revocate o ritirate. Se revochi o ritiri la concessione che HealthLake autorizza l'uso delle AWS KMS chiavi del tuo account, HealthLake non puoi accedere a questi dati, crittografare le nuove FHIR risorse trasferite nell'archivio dati o decrittografarle quando vengono estratte. Quando si revoca o si ritira una sovvenzione, la modifica avviene immediatamente. HealthLake Per revocare i diritti di accesso, è necessario eliminare l'archivio dati anziché revocare la concessione. Quando un data store viene eliminato, annulla le HealthLake concessioni per tuo conto.

Monitoraggio delle chiavi di crittografia per HealthLake

Puoi utilizzarlo CloudTrail per tenere traccia delle richieste HealthLake inviate a per tuo AWS KMS conto quando utilizzi una KMS chiave gestita dal cliente. Le voci di registro nel CloudTrail registro mostrano healthlake.amazonaws.com nel userAgent campo per distinguere chiaramente le richieste effettuate da. HealthLake

Gli esempi seguenti sono CloudTrail eventi per CreateGrant GenerateDataKey, Decrypt e per monitorare AWS KMS le operazioni richieste per accedere DescribeKey HealthLake ai dati crittografati dalla chiave gestita dal cliente.

Di seguito viene illustrato come utilizzare CreateGrant per consentire l'accesso HealthLake a una KMS chiave fornita dal cliente, che consente di utilizzare tale KMS chiave HealthLake per crittografare tutti i dati del cliente inattivi.

Gli utenti non sono tenuti a creare le proprie sovvenzioni. HealthLake crea una sovvenzione per tuo conto inviando una CreateGrant richiesta a AWSKMS. Le sovvenzioni AWS KMS vengono utilizzate per HealthLake consentire l'accesso a una AWS KMS chiave in un account 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"
}
```

Gli esempi seguenti mostrano come GenerateDataKey garantire che l'utente disponga delle autorizzazioni necessarie per crittografare i dati prima di archiviarli.

```
         {
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "EXAMPLEUSER",
        "arn": "arn:aws:sts::111122223333:assumed-role/Sampleuser01",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLEKEYID",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "EXAMPLEROLE",
                "arn": "arn:aws:iam::111122223333:role/Sampleuser01",
                "accountId": "111122223333",
                "userName": "Sampleuser01"
            },
            "webIdFederationData": {},
            "attributes": {
```

```
                "creationDate": "2021-06-30T21:17:06Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "healthlake.amazonaws.com"
    },
    "eventTime": "2021-06-30T21:17:37Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "healthlake.amazonaws.com",
    "userAgent": "healthlake.amazonaws.com",
    "requestParameters": {
        "keySpec": "AES_256",
        "keyId": "arn:aws:kms:us-east-1:111122223333:key/EXAMPLE_KEY_ARN"
    },
    "responseElements": null,
    "requestID": "EXAMPLE_ID_01",
    "eventID": "EXAMPLE_ID_02",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-east-1:111122223333:key/EXAMPLE_KEY_ARN"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

L'esempio seguente mostra come HealthLake richiama l'operazione Decrypt per utilizzare la chiave di dati crittografati archiviata per accedere ai dati crittografati.

```
            {
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "EXAMPLEUSER",
```

```
        "arn": "arn:aws:sts::111122223333:assumed-role/Sampleuser01",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLEKEYID",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "EXAMPLEROLE",
                "arn": "arn:aws:iam::111122223333:role/Sampleuser01",
                "accountId": "111122223333",
                "userName": "Sampleuser01"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2021-06-30T21:17:06Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "healthlake.amazonaws.com"
    },
    "eventTime": "2021-06-30T21:21:59Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "healthlake.amazonaws.com",
    "userAgent": "healthlake.amazonaws.com",
    "requestParameters": {
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "keyId": "arn:aws:kms:us-east-1:111122223333:key/EXAMPLE_KEY_ARN"
    },
    "responseElements": null,
    "requestID": "EXAMPLE_ID_01",
    "eventID": "EXAMPLE_ID_02",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-east-1:111122223333:key/EXAMPLE_KEY_ARN"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
```

```
}
```

L'esempio seguente mostra come HealthLake utilizza l' DescribeKey operazione per verificare se la AWS KMS chiave di proprietà AWS KMS del cliente è in uno stato utilizzabile e per aiutare l'utente a risolvere i problemi se non funziona.

```
         {
    "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"
}
```

Ulteriori informazioni

Le seguenti risorse forniscono ulteriori informazioni sulla crittografia dei dati a riposo.

Per ulteriori informazioni sui concetti di base del servizio di gestione delle AWS chiavi, consulta la AWS KMS documentazione.

Per ulteriori informazioni sulle migliori pratiche di sicurezza, consulta la AWS KMS documentazione.

Crittografia in transito per AWS HealthLake

AWS HealthLake utilizza TLS 1.2 per crittografare i dati in transito attraverso l'endpoint pubblico e tramite i servizi di backend.

Gestione delle identità e degli accessi per AWS HealthLake

AWS Identity and Access Management (IAM) è un dispositivo Servizio AWS che aiuta un amministratore a controllare in modo sicuro l'accesso alle AWS risorse. IAMgli amministratori controllano chi può essere autenticato (effettuato l'accesso) e autorizzato (disporre delle autorizzazioni) a utilizzare le risorse. HealthLake IAMè un dispositivo Servizio AWS che puoi utilizzare senza costi aggiuntivi.

Argomenti

- Destinatari

- Autenticazione con identità
- Gestione dell'accesso con policy
- Come AWS HealthLake funziona con IAM
- Esempi di policy basate sull'identità per AWS HealthLake
- AWS politiche gestite per AWS HealthLake
- Risoluzione dei problemi AWS HealthLake di identità e accesso

Destinatari

Il modo in cui usi AWS Identity and Access Management (IAM) varia a seconda del lavoro che svolgi. HealthLake

Utente del servizio: se utilizzi il HealthLake servizio per svolgere il tuo lavoro, l'amministratore ti fornisce le credenziali e le autorizzazioni necessarie. Man mano che utilizzi più HealthLake funzionalità per svolgere il tuo lavoro, potresti aver bisogno di autorizzazioni aggiuntive. La comprensione della gestione dell'accesso ti consente di richiedere le autorizzazioni corrette all'amministratore. Se non riesci ad accedere a una funzionalità di HealthLake, consulta Risoluzione dei problemi AWS HealthLake di identità e accesso.

Amministratore del servizio: se sei responsabile delle HealthLake risorse della tua azienda, probabilmente hai pieno accesso a HealthLake. È tuo compito determinare a quali HealthLake funzionalità e risorse devono accedere gli utenti del servizio. Devi inviare le richieste all'amministratore IAM per cambiare le autorizzazioni degli utenti del servizio. Esamina le informazioni contenute in questa pagina per comprendere le nozioni di base di IAM. Per ulteriori informazioni su come la tua azienda può utilizzare IAM HealthLake, consultaCome AWS HealthLake funziona con IAM.

IAMamministratore: se sei un IAM amministratore, potresti voler conoscere i dettagli su come scrivere politiche a cui gestire l'accesso HealthLake. Per visualizzare esempi di policy HealthLake basate sull'identità che puoi utilizzare inIAM, consulta. Esempi di policy basate sull'identità per AWS HealthLake

Autenticazione con identità

L'autenticazione è il modo in cui accedi AWS utilizzando le tue credenziali di identità. È necessario autenticarsi (accedere a AWS) come Utente root dell'account AWS, come IAM utente o assumendo un ruolo. IAM

È possibile accedere AWS come identità federata utilizzando le credenziali fornite tramite una fonte di identità. AWS IAM Identity Center Gli utenti (IAMIdentity Center), l'autenticazione Single Sign-On della tua azienda e le tue credenziali di Google o Facebook sono esempi di identità federate. Se accedi come identità federata, l'amministratore ha configurato in precedenza la federazione delle identità utilizzando i ruoli IAM. Quando accedi AWS utilizzando la federazione, assumi indirettamente un ruolo.

A seconda del tipo di utente, puoi accedere al AWS Management Console o al portale di AWS accesso. Per ulteriori informazioni sull'accesso a AWS, vedi Come accedere al tuo Account AWS nella Guida per l'Accedi ad AWS utente.

Se accedi a AWS livello di codice, AWS fornisce un kit di sviluppo software (SDK) e un'interfaccia a riga di comando () per firmare crittograficamente le tue richieste utilizzando le tue credenziali. CLI Se non utilizzi AWS strumenti, devi firmare tu stesso le richieste. Per ulteriori informazioni sull'utilizzo del metodo consigliato per firmare autonomamente le richieste, consulta AWS Signature Version 4 per API le richieste nella Guida per l'IAMutente.

A prescindere dal metodo di autenticazione utilizzato, potrebbe essere necessario specificare ulteriori informazioni sulla sicurezza. Ad esempio, ti AWS consiglia di utilizzare l'autenticazione a più fattori (MFA) per aumentare la sicurezza del tuo account. Per ulteriori informazioni, consulta Autenticazione a più fattori nella Guida per l'AWS IAM Identity Center utente e Autenticazione a AWS più fattori IAM nella Guida per l'IAMutente.

Account AWS utente root

Quando si crea un account Account AWS, si inizia con un'identità di accesso che ha accesso completo a tutte Servizi AWS le risorse dell'account. Questa identità è denominata utente Account AWS root ed è accessibile effettuando l'accesso con l'indirizzo e-mail e la password utilizzati per creare l'account. Si consiglia vivamente di non utilizzare l'utente root per le attività quotidiane. Conserva le credenziali dell'utente root e utilizzale per eseguire le operazioni che solo l'utente root può eseguire. Per l'elenco completo delle attività che richiedono l'accesso come utente root, consulta Attività che richiedono le credenziali dell'utente root nella Guida per l'IAMutente.

Identità federata

Come procedura consigliata, richiedi agli utenti umani, compresi gli utenti che richiedono l'accesso come amministratore, di utilizzare la federazione con un provider di identità per accedere Servizi AWS utilizzando credenziali temporanee.

Un'identità federata è un utente dell'elenco utenti aziendale, di un provider di identità Web AWS Directory Service, della directory Identity Center o di qualsiasi utente che accede utilizzando le Servizi AWS credenziali fornite tramite un'origine di identità. Quando le identità federate accedono Account AWS, assumono ruoli e i ruoli forniscono credenziali temporanee.

Per la gestione centralizzata degli accessi, consigliamo di utilizzare AWS IAM Identity Center. Puoi creare utenti e gruppi in IAM Identity Center oppure puoi connetterti e sincronizzarti con un set di utenti e gruppi nella tua fonte di identità per utilizzarli su tutte le tue applicazioni. Account AWS Per informazioni su IAM Identity Center, vedi Cos'è IAM Identity Center? nella Guida AWS IAM Identity Center per l'utente.

IAM users and groups

Un IAMutente è un'identità interna all'utente Account AWS che dispone di autorizzazioni specifiche per una singola persona o applicazione. Laddove possibile, consigliamo di fare affidamento su credenziali temporanee anziché creare IAM utenti con credenziali a lungo termine come password e chiavi di accesso. Tuttavia, se hai casi d'uso specifici che richiedono credenziali a lungo termine con IAM gli utenti, ti consigliamo di ruotare le chiavi di accesso. Per ulteriori informazioni, consulta Ruotare regolarmente le chiavi di accesso per i casi d'uso che richiedono credenziali a lungo termine nella Guida per l'utente. IAM

Un gruppo IAM è un'identità che specifica una raccolta di utenti IAM. Non è possibile eseguire l'accesso come gruppo. È possibile utilizzare gruppi per specificare le autorizzazioni per più utenti alla volta. I gruppi semplificano la gestione delle autorizzazioni per set di utenti di grandi dimensioni. Ad esempio, è possibile assegnare un nome a un gruppo IAMAdminse concedere a tale gruppo le autorizzazioni per amministrare le risorse. IAM

Gli utenti sono diversi dai ruoli. Un utente è associato in modo univoco a una persona o un'applicazione, mentre un ruolo è destinato a essere assunto da chiunque ne abbia bisogno. Gli utenti dispongono di credenziali a lungo termine permanenti, mentre i ruoli forniscono credenziali temporanee. Per ulteriori informazioni, consulta Casi d'uso per IAM gli utenti nella Guida per l'IAMutente.

Ruoli IAM

Un IAMruolo è un'identità interna all'utente Account AWS che dispone di autorizzazioni specifiche. È simile a un utente IAM ma non è associato a una persona specifica. Per assumere temporaneamente un IAM ruolo in AWS Management Console, puoi passare da un utente a un IAM ruolo (console). È possibile assumere un ruolo chiamando un' AWS APIoperazione AWS CLI or o utilizzando

un'operazione personalizzataURL. Per ulteriori informazioni sui metodi di utilizzo dei ruoli, vedere Metodi per assumere un ruolo nella Guida per l'IAMutente.

I ruoli IAM con credenziali temporanee sono utili nelle seguenti situazioni:

- Accesso utente federato: per assegnare le autorizzazioni a una identità federata, è possibile creare un ruolo e definire le autorizzazioni per il ruolo. Quando un'identità federata viene autenticata, l'identità viene associata al ruolo e ottiene le autorizzazioni da esso definite. Per informazioni sui ruoli per la federazione, consulta Creare un ruolo per un provider di identità di terze parti (federazione) nella Guida per l'IAMutente. Se utilizzi IAM Identity Center, configuri un set di autorizzazioni. Per controllare a cosa possono accedere le identità dopo l'autenticazione, IAM Identity Center correla il set di autorizzazioni a un ruolo in. IAM Per informazioni sui set di autorizzazioni, consulta Set di autorizzazioni nella Guida per l'utente di AWS IAM Identity Center .
- Autorizzazioni IAM utente temporanee: un IAM utente o un ruolo può assumere il IAM ruolo di assumere temporaneamente autorizzazioni diverse per un'attività specifica.
- Accesso multi-account: è possibile utilizzare un ruolo IAM per permettere a un utente (principale attendibile) di un account diverso di accedere alle risorse nel tuo account. I ruoli sono lo strumento principale per concedere l'accesso multi-account. Tuttavia, con alcuni Servizi AWS, è possibile allegare una policy direttamente a una risorsa (anziché utilizzare un ruolo come proxy). Per conoscere la differenza tra ruoli e politiche basate sulle risorse per l'accesso tra account diversi, consulta la sezione Accesso alle risorse su più account IAM nella Guida per l'utente. IAM
- Accesso tra servizi: alcuni Servizi AWS utilizzano funzionalità in altri. Servizi AWS Ad esempio, quando effettui una chiamata in un servizio, è normale che quel servizio esegua applicazioni in Amazon EC2 o archivi oggetti in Amazon S3. Un servizio può eseguire questa operazione utilizzando le autorizzazioni dell'entità chiamante, utilizzando un ruolo di servizio o utilizzando un ruolo collegato al servizio.
  - Sessioni di accesso diretto (FAS): quando utilizzi un IAM utente o un ruolo per eseguire azioni AWS, sei considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'azione che quindi avvia un'altra azione in un servizio diverso. FASutilizza le autorizzazioni del principale che chiama an Servizio AWS, in combinazione con la richiesta Servizio AWS per effettuare richieste ai servizi downstream. FASle richieste vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le azioni. Per i dettagli FAS delle politiche relative alle richieste, consulta Forward access sessions.
  - Ruolo di servizio: un ruolo di servizio è un IAMruolo che un servizio assume per eseguire azioni per conto dell'utente. Un amministratore IAM può creare, modificare ed eliminare un ruolo di

servizio da IAM. Per ulteriori informazioni, consulta Creare un ruolo per delegare le autorizzazioni a un utente Servizio AWS nella Guida per l'IAMutente.

- Ruolo collegato al servizio: un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un. Servizio AWS Il servizio può assumere il ruolo per eseguire un'azione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare, ma non modificare le autorizzazioni dei ruoli collegati ai servizi.

- Applicazioni in esecuzione su Amazon EC2: puoi utilizzare un IAM ruolo per gestire le credenziali temporanee per le applicazioni in esecuzione su un'EC2istanza e che effettuano AWS CLI o richiedono AWS API. Ciò è preferibile all'archiviazione delle chiavi di accesso nell'istanza EC2. Per assegnare un AWS ruolo a un'EC2istanza e renderlo disponibile per tutte le sue applicazioni, crei un profilo di istanza collegato all'istanza. Un profilo dell'istanza contiene il ruolo e consente ai programmi in esecuzione sull'istanza EC2 di ottenere le credenziali temporanee. Per ulteriori informazioni, consulta Utilizzare un IAM ruolo per concedere le autorizzazioni alle applicazioni in esecuzione su EC2 istanze Amazon nella Guida per l'IAMutente.

Gestione dell'accesso con policy

Puoi controllare l'accesso AWS creando policy e associandole a AWS identità o risorse. Una policy è un oggetto AWS che, se associato a un'identità o a una risorsa, ne definisce le autorizzazioni. AWS valuta queste politiche quando un principale (utente, utente root o sessione di ruolo) effettua una richiesta. Le autorizzazioni nelle policy determinano l'approvazione o il rifiuto della richiesta. La maggior parte delle politiche viene archiviata AWS come JSON documenti. Per ulteriori informazioni sulla struttura e il contenuto dei documenti relativi alle JSON politiche, vedere Panoramica delle JSON politiche nella Guida per l'IAMutente.

Gli amministratori possono utilizzare AWS JSON le politiche per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire azioni su quali risorse e in quali condizioni.

Per impostazione predefinita, utenti e ruoli non dispongono di autorizzazioni. Per concedere agli utenti l'autorizzazione a eseguire azioni sulle risorse di cui hanno bisogno, un IAM amministratore può creare IAM politiche. L'amministratore può quindi aggiungere le IAM politiche ai ruoli e gli utenti possono assumerli.

Le policy IAM definiscono le autorizzazioni relative a un'operazione, indipendentemente dal metodo utilizzato per eseguirla. Ad esempio, supponiamo di disporre di una policy che consente l'operazione

`iam:GetRole`. Un utente con tale criterio può ottenere informazioni sul ruolo da AWS Management Console, da o da AWS API. AWS CLI

Policy basate su identità

I criteri basati sull'identità sono documenti relativi alle politiche di JSON autorizzazione che è possibile allegare a un'identità, ad esempio un IAM utente, un gruppo di utenti o un ruolo. Tali policy definiscono le azioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una politica basata sull'identità, consulta Definire le IAM autorizzazioni personalizzate con le politiche gestite dal cliente nella Guida per l'utente. IAM

Le policy basate su identità possono essere ulteriormente classificate come policy inline o policy gestite. Le policy inline sono integrate direttamente in un singolo utente, gruppo o ruolo. Le politiche gestite sono politiche autonome che puoi allegare a più utenti, gruppi e ruoli all'interno del tuo. Account AWS Le politiche gestite includono politiche AWS gestite e politiche gestite dai clienti. Per informazioni su come scegliere tra una politica gestita o una politica in linea, consulta Scegliere tra politiche gestite e politiche in linea nella Guida per l'IAMutente.

Policy basate su risorse

Le politiche basate sulle risorse sono documenti di JSON policy allegati a una risorsa. Esempi di politiche basate sulle risorse sono le policy di trust dei IAM ruoli e le policy dei bucket di Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le azioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario specificare un principale in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Le policy basate sulle risorse sono policy inline che si trovano in tale servizio. Non è possibile utilizzare le policy AWS gestite contenute IAM in una policy basata sulle risorse.

Elenchi di controllo degli accessi () ACLs

Le liste di controllo degli accessi (ACLs) controllano quali principali (membri dell'account, utenti o ruoli) dispongono delle autorizzazioni per accedere a una risorsa. ACLssono simili alle politiche basate sulle risorse, sebbene non utilizzino il formato del documento di policy. JSON

Amazon S3 e Amazon VPC sono esempi di servizi che supportano. AWS WAF ACLs Per ulteriori informazioniACLs, consulta la panoramica di Access control list (ACL) nella Amazon Simple Storage Service Developer Guide.

Altri tipi di policy

AWS supporta tipi di policy aggiuntivi e meno comuni. Questi tipi di policy possono impostare il numero massimo di autorizzazioni concesse dai tipi di policy più comuni.

- Limiti delle autorizzazioni: un limite di autorizzazioni è una funzionalità avanzata in cui si impostano le autorizzazioni massime che una politica basata sull'identità può concedere a un'entità (utente o ruolo). IAM IAM È possibile impostare un limite delle autorizzazioni per un'entità. Le autorizzazioni risultanti sono l'intersezione delle policy basate su identità dell'entità e i relativi limiti delle autorizzazioni. Le policy basate su risorse che specificano l'utente o il ruolo nel campo `Principal`sono condizionate dal limite delle autorizzazioni. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni sui limiti delle autorizzazioni, consulta Limiti delle autorizzazioni per le entità nella Guida per l'utente. IAM IAM
- Politiche di controllo del servizio (SCPs): SCPs sono JSON politiche che specificano le autorizzazioni massime per un'organizzazione o un'unità organizzativa (OU) in. AWS Organizations AWS Organizations è un servizio per il raggruppamento e la gestione centralizzata di più Account AWS di proprietà dell'azienda. Se abiliti tutte le funzionalità di un'organizzazione, puoi applicare le politiche di controllo del servizio (SCPs) a uno o tutti i tuoi account. SCPLimita le autorizzazioni per le entità negli account dei membri, inclusa ciascuna Utente root dell'account AWS. Per ulteriori informazioni su Organizations andSCPs, consulta le politiche di controllo dei servizi nella Guida AWS Organizations per l'utente.
- Criteri di controllo delle risorse (RCPs): RCPs sono JSON criteri che puoi utilizzare per impostare le autorizzazioni massime disponibili per le risorse nei tuoi account senza aggiornare le IAM politiche allegate a ciascuna risorsa di tua proprietà. RCPLimita le autorizzazioni per le risorse negli account dei membri e può influire sulle autorizzazioni effettive per le identità, incluse le Utente root dell'account AWS, indipendentemente dal fatto che appartengano o meno all'organizzazione. Per ulteriori informazioni su Organizations eRCPs, incluso un elenco di Servizi AWS tale supportoRCPs, vedere Resource control policies (RCPs) nella Guida per l'AWS Organizations utente.
- Policy di sessione: le policy di sessione sono policy avanzate che vengono trasmesse come parametro quando si crea in modo programmatico una sessione temporanea per un ruolo o un utente federato. Le autorizzazioni della sessione risultante sono l'intersezione delle policy basate su identità del ruolo o dell'utente e le policy di sessione. Le autorizzazioni possono anche provenire da una policy basata su risorse. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni, consulta Policy di sessione nella Guida per l'utente di IAM.

Più tipi di policy

Quando più tipi di policy si applicano a una richiesta, le autorizzazioni risultanti sono più complicate da comprendere. Per sapere come si AWS determina se consentire una richiesta quando sono coinvolti più tipi di policy, consulta Logica di valutazione delle politiche nella Guida per l'IAMutente.

Come AWS HealthLake funziona con IAM

Prima di utilizzare IAM per gestire l'accesso a HealthLake, scopri con quali IAM funzionalità è disponibile l'uso HealthLake.

IAMfunzionalità che puoi usare con AWS HealthLake

Caratteristica IAM	HealthLake supporto
Policy basate su identità	Sì
Policy basate su risorse	No
Azioni di policy	Sì
Risorse relative alle policy	Sì
Chiavi di condizione delle policy	Sì
ACLs	No
ABAC(tag nelle politiche)	Sì
Credenziali temporanee	Sì
Autorizzazioni del principale	Sì
Ruoli di servizio	Sì
Ruoli collegati al servizio	No

Per avere una panoramica generale del funzionamento HealthLake e degli altri AWS servizi con la maggior parte delle IAM funzionalità, consulta AWS i servizi che funzionano con la maggior parte delle funzionalità IAM nella Guida per l'IAMutente.

Politiche basate sull'identità per AWS HealthLake

Supporta le policy basate su identità: sì

Le politiche basate sull'identità sono documenti relativi alle politiche di JSON autorizzazione che è possibile allegare a un'identità, ad esempio un IAM utente, un gruppo di utenti o un ruolo. Tali policy definiscono le azioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una politica basata sull'identità, consulta Definire le IAM autorizzazioni personalizzate con le politiche gestite dal cliente nella Guida per l'utente. IAM

Con le policy IAM IAM basate su identità, puoi specificare operazioni e risorse consentite o rifiutate, nonché le condizioni in base alle quali le operazioni sono consentite o rifiutate. Non è possibile specificare l'entità principale in una policy basata sull'identità perché si applica all'utente o al ruolo a cui è associato. Per maggiori informazioni su tutti gli elementi che puoi utilizzare in una JSON policy, consulta il riferimento agli elementi della IAM JSON policy nella Guida per l'utente. IAM

Esempi di policy basate sull'identità per AWS HealthLake

Per visualizzare esempi di politiche basate sull' HealthLake identità, vedere. Esempi di policy basate sull'identità per AWS HealthLake

Politiche basate sulle risorse all'interno AWS HealthLake

Supporta le policy basate su risorse: no

Le politiche basate sulle risorse sono documenti di JSON policy allegati a una risorsa. Esempi di politiche basate sulle risorse sono le policy di trust dei IAM ruoli e le policy dei bucket di Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le azioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario specificare un principale in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Per consentire l'accesso a più account, è possibile specificare un intero account o entità IAM in un altro account come entità principale in una policy basata su risorse. L'aggiunta di un principale multi-account a una policy basata sulle risorse rappresenta solo una parte della relazione di trust. Quando il principale e la risorsa sono diversi Account AWS, un IAM amministratore dell'account fidato deve inoltre concedere all'entità principale (utente o ruolo) l'autorizzazione ad accedere alla risorsa.

L'autorizzazione viene concessa collegando all'entità una policy basata sull'identità. Tuttavia, se una policy basata su risorse concede l'accesso a un principale nello stesso account, non sono richieste ulteriori policy basate su identità. Per ulteriori informazioni, consulta la sezione Cross Account Resource Access IAM nella Guida IAM per l'utente.

Azioni politiche per AWS HealthLake

Supporta le operazioni di policy: si

Gli amministratori possono utilizzare AWS JSON le policy per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'`Action`elemento di una JSON policy descrive le azioni che è possibile utilizzare per consentire o negare l'accesso a una policy. Le azioni politiche in genere hanno lo stesso nome dell' AWS APIoperazione associata. Esistono alcune eccezioni, come le azioni basate solo sulle autorizzazioni che non hanno un'operazione corrispondente. API Esistono anche alcune operazioni che richiedono più operazioni in una policy. Queste operazioni aggiuntive sono denominate operazioni dipendenti.

Includi le operazioni in una policy per concedere le autorizzazioni a eseguire l'operazione associata.

Per visualizzare un elenco di HealthLake azioni, vedere Azioni definite da AWS HealthLake nel Service Authorization Reference.

Le azioni politiche in HealthLake uso utilizzano il seguente prefisso prima dell'azione:

```
healthlake
```

Per specificare più azioni in una singola istruzione, separa ogni azione con una virgola.

```
"Action": [
      "healthlake:action1",
      "healthlake:action2"
         ]
```

Per visualizzare esempi di politiche HealthLake basate sull'identità, vedere. Esempi di policy basate sull'identità per AWS HealthLake

Risorse politiche per AWS HealthLake

Supporta le risorse di policy: sì

Gli amministratori possono utilizzare AWS JSON le policy per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento `Resource` JSON policy specifica l'oggetto o gli oggetti a cui si applica l'azione. Le istruzioni devono includere un elemento `Resource`o un elemento `NotResource`. Come best practice, specifica una risorsa utilizzando il relativo Amazon Resource Name (ARN). Puoi eseguire questa operazione per azioni che supportano un tipo di risorsa specifico, note come autorizzazioni a livello di risorsa.

Per le azioni che non supportano le autorizzazioni a livello di risorsa, ad esempio le operazioni di elenco, utilizza un carattere jolly (*) per indicare che l'istruzione si applica a tutte le risorse.

```
"Resource": "*"
```

Per visualizzare un elenco dei tipi di HealthLake risorse e relativiARNs, consulta Resources defined by AWS HealthLake nel Service Authorization Reference. Per informazioni sulle azioni con cui è possibile specificare il valore ARN di ciascuna risorsa, vedere Azioni definite da AWS HealthLake.

Per visualizzare esempi di politiche HealthLake basate sull'identità, vedere. Esempi di policy basate sull'identità per AWS HealthLake

Chiavi relative alle condizioni delle politiche per AWS HealthLake

Supporta le chiavi di condizione delle policy specifiche del servizio: sì

Gli amministratori possono utilizzare AWS JSON le politiche per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire azioni su quali risorse, e in quali condizioni.

L'elemento `Condition`(o blocco `Condition`) consente di specificare le condizioni in cui un'istruzione è in vigore. L'elemento `Condition`è facoltativo. Puoi compilare espressioni condizionali che utilizzano operatori di condizione, ad esempio uguale a o minore di, per soddisfare la condizione nella policy con i valori nella richiesta.

Se specifichi più elementi `Condition`in un'istruzione o più chiavi in un singolo elemento `Condition`, questi vengono valutati da AWS utilizzando un'operazione `AND`logica. Se si specificano

più valori per una singola chiave di condizione, AWS valuta la condizione utilizzando un'operazione logica0R. Tutte le condizioni devono essere soddisfatte prima che le autorizzazioni dell'istruzione vengano concesse.

Puoi anche utilizzare variabili segnaposto quando specifichi le condizioni. Ad esempio, puoi concedere a un utente IAM l'autorizzazione per accedere a una risorsa solo se è stata taggata con il nome utente IAM. Per ulteriori informazioni, consulta Elementi IAM della politica: variabili e tag nella Guida per l'IAMutente.

AWS supporta chiavi di condizione globali e chiavi di condizione specifiche del servizio. Per visualizzare tutte le chiavi di condizione AWS globali, consulta le chiavi di contesto delle condizioni AWS globali nella Guida per l'IAMutente.

Per visualizzare un elenco di chiavi di HealthLake condizione, consulta Condition keys for AWS HealthLake nel Service Authorization Reference. Per informazioni sulle azioni e le risorse con cui è possibile utilizzare una chiave di condizione, consulta Azioni definite da AWS HealthLake.

Per visualizzare esempi di politiche HealthLake basate sull'identità, vedere. Esempi di policy basate sull'identità per AWS HealthLake

Liste di controllo degli accessi () in ACLs AWS HealthLake

SupportiACLs: no

Le liste di controllo degli accessi (ACLs) controllano quali principali (membri dell'account, utenti o ruoli) dispongono delle autorizzazioni per accedere a una risorsa. ACLssono simili alle politiche basate sulle risorse, sebbene non utilizzino il formato del documento di policy. JSON

Controllo degli accessi basato sugli attributi () con ABAC AWS HealthLake

Supporti ABAC (tag nelle politiche): Sì

Il controllo degli accessi basato sugli attributi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base agli attributi. In AWS, questi attributi sono chiamati tag. È possibile allegare tag a IAM entità (utenti o ruoli) e a molte AWS risorse. L'etichettatura di entità e risorse è il primo passo diABAC. Quindi si progettano ABAC politiche per consentire le operazioni quando il tag del principale corrisponde al tag sulla risorsa a cui sta tentando di accedere.

ABACè utile in ambienti in rapida crescita e aiuta in situazioni in cui la gestione delle politiche diventa complicata.

Per controllare l'accesso basato su tag, fornisci informazioni sui tag nell'elemento condizione di una policy utilizzando le chiavi di condizione `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`o `aws:TagKeys`.

Se un servizio supporta tutte e tre le chiavi di condizione per ogni tipo di risorsa, il valore per il servizio è Yes (Sì). Se un servizio supporta tutte e tre le chiavi di condizione solo per alcuni tipi di risorsa, allora il valore sarà Parziale.

Per ulteriori informazioni in meritoABAC, vedere Definizione delle autorizzazioni con ABAC autorizzazione nella Guida per l'IAMutente. Per visualizzare un tutorial con i passaggi per la configurazioneABAC, consulta Use Attribute-based access control (ABAC) nella Guida per l'utente. IAM

Utilizzo di credenziali temporanee con AWS HealthLake

Supporta le credenziali temporanee: sì

Alcuni Servizi AWS non funzionano quando si accede utilizzando credenziali temporanee. Per ulteriori informazioni, incluse quelle che Servizi AWS funzionano con credenziali temporanee, consulta la sezione Servizi AWS relativa alla funzionalità IAM nella Guida per l'IAMutente.

Si utilizzano credenziali temporanee se si accede AWS Management Console utilizzando qualsiasi metodo tranne il nome utente e la password. Ad esempio, quando accedete AWS utilizzando il link Single Sign-on (SSO) della vostra azienda, tale processo crea automaticamente credenziali temporanee. Le credenziali temporanee vengono create in automatico anche quando accedi alla console come utente e poi cambi ruolo. Per ulteriori informazioni sul cambio di ruolo, consulta Passare da un utente a un IAM ruolo (console) nella Guida per l'IAMutente.

È possibile creare manualmente credenziali temporanee utilizzando AWS CLI o AWS API. È quindi possibile utilizzare tali credenziali temporanee per accedere. AWS AWS consiglia di generare dinamicamente credenziali temporanee anziché utilizzare chiavi di accesso a lungo termine. Per ulteriori informazioni, vedere Credenziali di sicurezza temporanee in. IAM

Autorizzazioni principali per più servizi per AWS HealthLake

Supporta sessioni di accesso diretto (FAS): Sì

Quando utilizzi un IAM utente o un ruolo per eseguire azioni AWS, sei considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'azione che quindi avvia un'altra azione

in un servizio diverso. FASutilizza le autorizzazioni del principale che chiama an Servizio AWS, in combinazione con la richiesta Servizio AWS per effettuare richieste ai servizi downstream. FASle richieste vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le azioni. Per i dettagli FAS delle politiche relative alle richieste, consulta Forward access sessions.

Ruoli di servizio per AWS HealthLake

Supporta i ruoli di servizio: sì

Un ruolo di servizio è un IAMruolo che un servizio assume per eseguire azioni per conto dell'utente. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio da IAM. Per ulteriori informazioni, consulta Creare un ruolo per delegare le autorizzazioni a un utente Servizio AWS nella Guida per l'IAMutente.

Per informazioni sui ruoli di servizio e sulla politica in linea richiesta per l'accesso completo a AWS HealthLake, vedere. Configurazione delle autorizzazioni per iniziare a utilizzare AWS HealthLake

> **Warning**
>
> La modifica delle autorizzazioni per un ruolo di servizio potrebbe comprometterne la funzionalità. HealthLake Modifica i ruoli di servizio solo quando viene HealthLake fornita una guida in tal senso.

Ruoli collegati ai servizi per AWS HealthLake

Supporta i ruoli collegati ai servizi: no

Un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un. Servizio AWS Il servizio può assumere il ruolo per eseguire un'azione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare, ma non modificare le autorizzazioni dei ruoli collegati ai servizi.

Per informazioni dettagliate sulla creazione o la gestione di ruoli collegati ai servizi, consulta AWS Servizi compatibili con. IAM Trova un servizio nella tabella che include un `Yes` nella colonna Service-linked role (Ruolo collegato ai servizi). Scegli il collegamento Sì per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

Esempi di policy basate sull'identità per AWS HealthLake

Per impostazione predefinita, gli utenti e i ruoli non dispongono dell'autorizzazione per creare o modificare risorse HealthLake. Inoltre, non possono eseguire attività utilizzando AWS Management Console, AWS Command Line Interface (AWS CLI) o. AWS API Per concedere agli utenti il permesso di eseguire azioni sulle risorse di cui hanno bisogno, un IAM amministratore può creare IAM policy. L'amministratore può quindi aggiungere le IAM politiche ai ruoli e gli utenti possono assumerli.

Per informazioni su come creare una politica IAM basata sull'identità utilizzando questi documenti di esempio, consulta Create JSON IAM policy (console) nella Guida per l'IAMutente.

Per informazioni dettagliate sulle azioni e sui tipi di risorse definiti da HealthLake, incluso il formato di ARNs per ogni tipo di risorsa, vedere Azioni, risorse e chiavi di condizione AWS HealthLake nel Service Authorization Reference.

Argomenti

- Best practice per le policy
- Utilizzo della console di AWS HealthLake
- Accesso a un AWS HealthLake data store in Amazon Athena
- Consentire agli utenti di visualizzare le loro autorizzazioni

Best practice per le policy

Le politiche basate sull'identità determinano se qualcuno può creare, accedere o eliminare HealthLake risorse nel tuo account. Queste azioni possono comportare costi aggiuntivi per l' Account AWS. Quando crei o modifichi policy basate su identità, segui queste linee guida e raccomandazioni:

- Inizia con le policy AWS gestite e passa alle autorizzazioni con privilegi minimi: per iniziare a concedere autorizzazioni a utenti e carichi di lavoro, utilizza le politiche gestite che concedono le autorizzazioni per molti casi d'uso comuni.AWS Sono disponibili nel tuo. Account AWS Ti consigliamo di ridurre ulteriormente le autorizzazioni definendo politiche gestite dai AWS clienti specifiche per i tuoi casi d'uso. Per ulteriori informazioni, consulta le politiche AWS gestite o le politiche AWS gestite per le funzioni lavorative nella Guida per l'IAMutente.
- Applica le autorizzazioni con privilegi minimi: quando imposti le autorizzazioni con le IAM politiche, concedi solo le autorizzazioni necessarie per eseguire un'attività. Puoi farlo definendo le azioni che possono essere intraprese su risorse specifiche in condizioni specifiche, note anche

come autorizzazioni con privilegi minimi. Per ulteriori informazioni sull'utilizzo per applicare le autorizzazioni, consulta Politiche e autorizzazioni nella Guida IAM per l'utente. IAM IAM

- Utilizza le condizioni nelle IAM politiche per limitare ulteriormente l'accesso: puoi aggiungere una condizione alle tue politiche per limitare l'accesso ad azioni e risorse. Ad esempio, puoi scrivere una condizione di policy per specificare che tutte le richieste devono essere inviate utilizzandoSSL. È inoltre possibile utilizzare condizioni per concedere l'accesso alle azioni di servizio se vengono utilizzate tramite uno specifico Servizio AWS, ad esempio AWS CloudFormation. Per ulteriori informazioni, consulta Elementi IAM JSON della politica: Condizione nella Guida IAM per l'utente.
- Usa IAM Access Analyzer per convalidare IAM le tue policy e garantire autorizzazioni sicure e funzionali: IAM Access Analyzer convalida le policy nuove ed esistenti in modo che aderiscano al linguaggio delle IAM policy () e alle best practice. JSON IAM IAM Access Analyzer fornisce più di 100 controlli delle policy e consigli pratici per aiutarti a creare policy sicure e funzionali. Per ulteriori informazioni, consulta Convalida delle politiche con IAM Access Analyzer nella Guida per l'utente. IAM
- Richiedi l'autenticazione a più fattori (MFA): se hai uno scenario che richiede l'utilizzo di IAM utenti o di un utente root Account AWS, attiva questa opzione MFA per una maggiore sicurezza. Per richiedere MFA quando vengono richiamate API le operazioni, aggiungi MFA delle condizioni alle tue politiche. Per ulteriori informazioni, consulta Secure API access with MFA nella Guida IAM per l'utente.

Per ulteriori informazioni sulle best practice inIAM, consulta la sezione Procedure consigliate in materia di sicurezza IAM nella Guida IAM per l'utente.

Utilizzo della console di AWS HealthLake

Per accedere alla AWS HealthLake console, è necessario disporre di un set minimo di autorizzazioni. Queste autorizzazioni devono consentirti di elencare e visualizzare i dettagli sulle HealthLake risorse del tuo. Account AWS Se crei una policy basata sull'identità più restrittiva rispetto alle autorizzazioni minime richieste, la console non funzionerà nel modo previsto per le entità (utenti o ruoli) associate a tale policy.

Non è necessario concedere autorizzazioni minime per la console agli utenti che effettuano chiamate solo verso il AWS CLI o il. AWS API Consenti invece l'accesso solo alle azioni che corrispondono all'APIoperazione che stanno cercando di eseguire.

Per un accesso completo a HealthLake, allega le seguenti politiche a un IAM utente o ruolo: `AmazonHealthLakeFullAccess` e`AWSLakeFormationDataAdmin`. È inoltre necessario allegare

la politica HealthLake in linea che è un ruolo di servizio. Un ruolo di servizio è un IAMruolo che un servizio assume per eseguire azioni per conto dell'utente. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio da IAM. Per ulteriori informazioni, consulta Creare un ruolo per delegare le autorizzazioni a un utente Servizio AWS nella Guida per l'IAMutente. Per informazioni sulla politica in linea che crea il ruolo di servizio richiesto, vedere. Configurazione delle autorizzazioni per iniziare a utilizzare AWS HealthLake È inoltre necessario utilizzare la AWS Lake Formation console o CLI assegnare all' HealthLake amministratore il ruolo di amministratore di AWS Lake Formation Data Lake. Per ulteriori informazioni, consulta Configurazione delle autorizzazioni per iniziare a utilizzare AWS HealthLake.

Accesso a un AWS HealthLake data store in Amazon Athena

Se desideri fornire a utenti e ruoli l'accesso agli archivi HealthLake dati in Amazon Athena, collega le seguenti IAM politiche al ruolo o all'utente: `AmazonAthenaFullAccess` e`AmazonS3FullAccess`. `Select` `Describe` le autorizzazioni sono richieste anche per le tabelle gestite da AWS Lake Formation. AWS Lake Formation le autorizzazioni relative alle tabelle vengono concesse da un AWS Lake Formation amministratore nella AWS Lake Formation console o tramite. CLI Per ulteriori informazioni, consulta Configurazione delle autorizzazioni per iniziare a utilizzare AWS HealthLake

Consentire agli utenti di visualizzare le loro autorizzazioni

Questo esempio mostra in che modo è possibile creare una policy che consente agli utenti IAM di visualizzare le policy inline e gestite che sono collegate alla relativa identità utente. Questa politica include le autorizzazioni per completare questa azione sulla console o utilizzando o a livello di codice. AWS CLI 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 politiche gestite per AWS HealthLake

Una politica AWS gestita è una politica autonoma creata e amministrata da AWS. AWS le politiche gestite sono progettate per fornire autorizzazioni per molti casi d'uso comuni, in modo da poter iniziare ad assegnare autorizzazioni a utenti, gruppi e ruoli.

Tieni presente che le policy AWS gestite potrebbero non concedere le autorizzazioni con il privilegio minimo per i tuoi casi d'uso specifici, poiché sono disponibili per tutti i clienti. AWS Ti consigliamo pertanto di ridurre ulteriormente le autorizzazioni definendo policy gestite dal cliente specifiche per i tuoi casi d'uso.

Non è possibile modificare le autorizzazioni definite nelle politiche gestite. AWS Se AWS aggiorna le autorizzazioni definite in una politica AWS gestita, l'aggiornamento ha effetto su tutte le identità principali (utenti, gruppi e ruoli) a cui è associata la politica. AWS è più probabile che aggiorni una policy AWS gestita quando ne Servizio AWS viene lanciata una nuova o quando diventano disponibili nuove API operazioni per i servizi esistenti.

Per ulteriori informazioni, consulta Policy gestite da AWSnellaGuida per l'utente IAM.

AWS politica gestita: AmazonHealthLakeFullAccess

La `AmazonHealthLakeFullAccess` politica fornisce l'accesso completo a HealthLake. Con questa policy associata al proprio utente o ruolo, gli utenti possono utilizzarla HealthLake per accedere, interrogare, importare ed esportare dati in HealthLake. Per eseguire molte azioni comuni in HealthLake, è necessario aggiungere politiche aggiuntive all'utente o al ruolo. Per ulteriori informazioni, consulta Configurazione delle autorizzazioni per iniziare a utilizzare AWS HealthLake e HealthLake operazioni e autorizzazioni.

È possibile collegare la policy `AmazonHealthLakeFullAccess` alle identità IAM.

Questa politica concede *`administrative and contributor`* autorizzazioni che consentono a utenti e ruoli di eseguire query, cercare, importare ed esportare e consente inoltre di eseguire azioni HealthLake per conto degli utenti e dei ruoli che dispongono di tali autorizzazioni. HealthLake

Dettagli dell'autorizzazione

Questa politica include la seguente dichiarazione.

```
{
 "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 politica gestita: AmazonHealthLakeReadOnlyAccess

`AmazonHealthLakeReadOnlyAccess`la politica concede l'accesso e le autorizzazioni in sola lettura HealthLake e alle risorse correlate in altri servizi. AWS Applica questo criterio agli utenti a cui desideri concedere la possibilità di interrogare e visualizzare gli archivi HealthLake dati, ma non la possibilità di crearli o modificarli.

È possibile collegare la policy `AmazonHealthLakeReadOnlyAccess` alle identità IAM.

Questa politica concede *`read-only`* autorizzazioni che consentono a utenti e ruoli di eseguire query. HealthLake

Dettagli dell'autorizzazione

Questa politica include la seguente dichiarazione.

```
{
    "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 operazioni e autorizzazioni

Nella tabella seguente sono elencate le operazioni tipiche di HealthLake e le autorizzazioni necessarie per eseguirle.

HealthLake operazioni	Autorizzazioni richieste
Creare un archivio dati in HealthLake	`AmazonHealthLakeFullAccess` `AmazonLakeFormationDataAdmin` , policy in linea e autorizzazioni di AWS Lake Formation amministratore gestite da AWS Lake Formation
Eliminare un archivio dati in HealthLake	`AmazonHealthLakeFullAccess` `AmazonLakeFormationDataAdmin` , policy in linea e autorizzazioni di AWS Lake Formation amministratore gestite da AWS Lake Formation
Elenca, cerca o interroga un archivio dati in HealthLake	`AmazonHealthLakeReadOnlyAccess`
Effettua una query su un data store utilizzando Amazon Athena	`AmazonAthenaFullAccess` AWS Lake Formation `Select` e `Describe` autorizzazioni per le tabelle gestite da `AmazonS3FullAccess` AWS Lake Formation
Importa dati da HealthLake	Per informazioni, consulta Impostazione delle autorizzazioni per i lavori di importazione.

HealthLake operazioni	Autorizzazioni richieste
Esporta dati da HealthLake	Per informazioni, consulta Esportazione di file dal tuo archivio dati ()AWS SDKs.

HealthLake aggiornamenti alle politiche AWS gestite

Visualizza i dettagli sugli aggiornamenti delle politiche AWS gestite HealthLake dal momento in cui questo servizio ha iniziato a tenere traccia di queste modifiche. Per ricevere avvisi automatici sulle modifiche a questa pagina, iscriviti al RSS feed nella pagina della cronologia dei HealthLake documenti.

Modifica	Descrizione	Data
AmazonHealthLakeFullAccess	`AmazonHealthLakeFullAccess` criterio richiesto per consentire l'accesso completo a. HealthLake	14 novembre 2022
AmazonHealthLakeReadOnlyAccess	`AmazonHealthLakeReadOnlyAccess` criterio richiesto per l'accesso in sola lettura a. HealthLake	14 novembre 2022
HealthLake ha iniziato a tenere traccia delle modifiche	HealthLake ha iniziato a tenere traccia delle modifiche per le sue politiche AWS gestite.	14 novembre 2022

Risoluzione dei problemi AWS HealthLake di identità e accesso

Utilizza le seguenti informazioni per aiutarti a diagnosticare e risolvere i problemi più comuni che potresti riscontrare quando lavori con HealthLake eIAM.

Argomenti

- Non sono autorizzato a eseguire alcuna azione in AWS HealthLake

- Non sono autorizzato a eseguire iam: PassRole
- Voglio consentire a persone esterne al mio AWS account di accedere alle mie AWS HealthLake risorse

Non sono autorizzato a eseguire alcuna azione in AWS HealthLake

Se ti AWS Management Console dice che non sei autorizzato a eseguire un'azione, devi contattare l'amministratore per ricevere assistenza. L'amministratore è la persona da cui si sono ricevuti il nome utente e la password.

L'errore di esempio seguente si verifica quando l'`mateojackson`IAMutente tenta di utilizzare la console per visualizzare i dettagli su una *`my-example-widget`* risorsa fittizia ma non dispone delle autorizzazioni fittizie`healthlake:`*`GetWidget`*.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 healthlake:GetWidget on resource: my-example-widget
```

In questo caso, Mateo richiede al suo amministratore di aggiornare le policy per poter accedere alla risorsa *`my-example-widget`* utilizzando l'operazione `healthlake:`*`GetWidget`*.

Non sono autorizzato a eseguire iam: PassRole

Se ricevi un errore che indica che non sei autorizzato a eseguire l'operazione `iam:PassRole`, le tue policy devono essere aggiornate per poter passare un ruolo a HealthLake.

Alcuni Servizi AWS consentono di passare un ruolo esistente a quel servizio invece di creare un nuovo ruolo di servizio o un ruolo collegato al servizio. Per eseguire questa operazione, è necessario disporre delle autorizzazioni per trasmettere il ruolo al servizio.

L'errore di esempio seguente si verifica quando un utente IAM denominato `marymajor` cerca di utilizzare la console per eseguire un'operazione in HealthLake. Tuttavia, l'azione richiede che il servizio disponga delle autorizzazioni concesse da un ruolo di servizio. Mary non dispone delle autorizzazioni per passare il ruolo al servizio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In questo caso, le policy di Mary devono essere aggiornate per poter eseguire l'operazione `iam:PassRole`.

Se hai bisogno di aiuto, contatta il tuo AWS amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

Voglio consentire a persone esterne al mio AWS account di accedere alle mie AWS HealthLake risorse

È possibile creare un ruolo con il quale utenti in altri account o persone esterne all'organizzazione possono accedere alle tue risorse. È possibile specificare chi è attendibile per l'assunzione del ruolo. Per i servizi che supportano politiche basate sulle risorse o liste di controllo degli accessi (ACLs), puoi utilizzare tali politiche per concedere alle persone l'accesso alle tue risorse.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- Per sapere se HealthLake supporta queste funzionalità, consulta. Come AWS HealthLake funziona con IAM
- Per informazioni su Account AWS come fornire l'accesso alle risorse di tua proprietà, consulta Fornire l'accesso a un IAM utente di un altro Account AWS utente di tua proprietà nella Guida per l'IAMutente.
- Per scoprire come fornire l'accesso alle tue risorse a terze parti Account AWS, consulta Fornire l'accesso a persone Account AWS di proprietà di terzi nella Guida per l'IAMutente.
- Per informazioni su come fornire l'accesso tramite la federazione delle identità, consulta Fornire l'accesso agli utenti autenticati esternamente (federazione delle identità) nella Guida per l'IAMutente.
- Per conoscere la differenza tra l'utilizzo di ruoli e politiche basate sulle risorse per l'accesso tra account diversi, consulta la sezione Accesso alle risorse tra account nella Guida per l'utente. IAM IAM

Chiamate AWS HealthLake API di registrazione con AWS CloudTrail

AWS HealthLake è integrato con AWS CloudTrail, un servizio che fornisce una registrazione delle azioni intraprese da un utente, un ruolo o un AWS servizio in HealthLake. CloudTrail acquisisce tutte le API chiamate HealthLake come eventi. Le chiamate acquisite includono chiamate dalla HealthLake console e chiamate in codice alle HealthLake API operazioni. Se crei un trail, puoi abilitare la distribuzione continua di CloudTrail eventi a un bucket Amazon S3, inclusi gli eventi per. HealthLake Se non configuri un percorso, puoi comunque visualizzare gli eventi più recenti nella

CloudTrail console nella cronologia degli eventi. Utilizzando le informazioni raccolte da CloudTrail, puoi determinare a quale richiesta è stata inviata HealthLake, l'indirizzo IP da cui è stata effettuata, chi ha effettuato la richiesta, quando è stata effettuata e dettagli aggiuntivi.

Per ulteriori informazioni CloudTrail, consulta la Guida AWS CloudTrail per l'utente.

AWS HealthLake Informazioni in CloudTrail

CloudTrail è abilitato sul tuo AWS account al momento della creazione dell'account. Quando si verifica un'attività in HealthLake, tale attività viene registrata in un CloudTrail evento insieme ad altri eventi AWS di servizio nella cronologia degli eventi. È possibile visualizzare, cercare e scaricare gli eventi recenti nell'account AWS . Per ulteriori informazioni, vedere Visualizzazione degli eventi con la cronologia degli CloudTrail eventi.

Per una registrazione continua degli eventi nel tuo AWS account, inclusi gli eventi di HealthLake, crea un percorso. Un trail consente di CloudTrail inviare file di log a un bucket Amazon S3. Per impostazione predefinita, quando si crea un trail nella console, il trail sarà valido in tutte le Regioni AWS. Il trail registra gli eventi di tutte le regioni della AWS partizione e consegna i file di log al bucket Amazon S3 specificato. Inoltre, puoi configurare altri AWS servizi per analizzare ulteriormente e agire in base ai dati sugli eventi raccolti nei log. CloudTrail Per ulteriori informazioni, consulta gli argomenti seguenti:

- Panoramica della creazione di un trail
- CloudTrail Servizi e integrazioni supportati
- Configurazione di Amazon SNS Notifications per CloudTrail
- Ricezione di file di CloudTrail registro da più regioni e ricezione di file di CloudTrail registro da più account

Tutte HealthLake le azioni vengono registrate da CloudTrail e sono documentate nella Guida di HealthLake APIriferimento e nella presente Guida per gli sviluppatori per le azioni eseguite utilizzando. FHIR REST API Ad esempio, le chiamate alle seguenti azioni generano voci nei file di CloudTrail registro:

- `DescribeFHIRImportJob`
- `DescribeFHIRExportJob`
- `StartFHIRImportJob`
- `ListFHIRImportJobs`

- `StartFHIRExportJob`
- `ListFHIRExportJobs`
- `CreateFHIRDatastore`
- `ListFHIRDatastores`
- `DeleteFHIRDatastore`
- `DescribeFHIRDatastore`
- `UpdateResource`
- `CreateResource`
- `DeleteResource`
- `ReadResource`
- `GetCapabilities`
- `SearchWithGet`
- `SearchWithPost`

Ogni evento o voce di log contiene informazioni sull'utente che ha generato la richiesta. Le informazioni di identità consentono di determinare quanto segue:

- Se la richiesta è stata effettuata con credenziali utente root o AWS Identity and Access Management (IAM).
- Se la richiesta è stata effettuata con le credenziali di sicurezza temporanee per un ruolo o un utente federato.
- Se la richiesta è stata effettuata da un altro AWS servizio.

Per ulteriori informazioni, consulta l'CloudTrail userIdentityelemento.

Comprensione delle AWS HealthLake voci dei file di registro

Un trail è una configurazione che consente la distribuzione di eventi come file di log in un bucket Amazon S3 specificato dall'utente. CloudTrail i file di registro contengono una o più voci di registro. Un evento rappresenta una singola richiesta proveniente da qualsiasi fonte e include informazioni sull'azione richiesta, la data e l'ora dell'azione, i parametri della richiesta e così via. CloudTrail i file di registro non sono una traccia stack ordinata delle API chiamate pubbliche, quindi non vengono visualizzati in un ordine specifico.

L'esempio seguente mostra una voce di CloudTrail registro che illustra l'`CreateFHIRDatastore`azione.


```
{
    "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"
}
```

Convalida della conformità per AWS HealthLake

I revisori esterni valutano la sicurezza e la conformità nell' AWS HealthLake ambito di più programmi di AWS conformità. Per HealthLake questo includeHIPAA.

Per sapere se un Servizio AWS programma rientra nell'ambito di specifici programmi di conformità, consulta Servizi AWS la sezione Ambito per programma di conformitàServizi AWS di conformità e scegli il programma di conformità che ti interessa. Per informazioni generali, consulta Programmi di AWS conformità Programmi di .

È possibile scaricare report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta Scaricamento dei report in AWS Artifact .

La vostra responsabilità di conformità durante l'utilizzo Servizi AWS è determinata dalla sensibilità dei dati, dagli obiettivi di conformità dell'azienda e dalle leggi e dai regolamenti applicabili. AWS fornisce le seguenti risorse per contribuire alla conformità:

- Governance e conformità per la sicurezza: queste guide all'implementazione di soluzioni illustrano considerazioni relative all'architettura e i passaggi per implementare le funzionalità di sicurezza e conformità.
- Architettura per la HIPAA sicurezza e la conformità su Amazon Web Services: questo white paper descrive in che modo le aziende possono utilizzare AWS per creare applicazioni idonee. HIPAA

> **Note**
>
> Non tutte sono idonee. Servizi AWS HIPAA Per ulteriori informazioni, consulta la Guida ai servizi HIPAA idonei.

- AWS Risorse per la per la conformità: questa raccolta di cartelle di lavoro e guide potrebbe riguardare il settore e la località in cui operi.
- AWS Guide alla conformità dei clienti: comprendi il modello di responsabilità condivisa attraverso la lente della conformità. Le guide riassumono le migliori pratiche per la protezione Servizi AWS e mappano le linee guida per i controlli di sicurezza su più framework (tra cui il National Institute of Standards and Technology (NIST), il Payment Card Industry Security Standards Council (PCI) e l'International Organization for Standardization ()). ISO
- Evaluating Resources with Rules nella Guida per gli AWS Config sviluppatori: il AWS Config servizio valuta la conformità delle configurazioni delle risorse alle pratiche interne, alle linee guida del settore e alle normative.
- AWS Security Hub— Ciò Servizio AWS fornisce una visione completa dello stato di sicurezza interno. AWS La Centrale di sicurezza utilizza i controlli di sicurezza per valutare le risorse AWS e verificare la conformità agli standard e alle best practice del settore della sicurezza. Per un elenco dei servizi e dei controlli supportati, consulta la pagina Documentazione di riferimento sui controlli della Centrale di sicurezza.
- Amazon GuardDuty: Servizio AWS rileva potenziali minacce ai tuoi carichi di lavoro Account AWS, ai contenitori e ai dati monitorando l'ambiente alla ricerca di attività sospette e dannose. GuardDuty può aiutarti a soddisfare vari requisiti di conformità, ad esempio PCI DSS soddisfacendo i requisiti di rilevamento delle intrusioni imposti da determinati framework di conformità.
- AWS Audit Manager— Ciò Servizio AWS consente di verificare continuamente AWS l'utilizzo per semplificare la gestione del rischio e la conformità alle normative e agli standard di settore.

Resilienza in AWS HealthLake

L'infrastruttura AWS globale è costruita attorno a AWS regioni e zone di disponibilità. AWS Le regioni forniscono più zone di disponibilità fisicamente separate e isolate, collegate con reti a bassa latenza, ad alto throughput e altamente ridondanti. Con le zone di disponibilità, puoi progettare e gestire applicazioni e database che eseguono automaticamente il failover tra zone di disponibilità senza interruzioni. Le zone di disponibilità sono più disponibili, tolleranti ai guasti e scalabili rispetto alle infrastrutture a data center singolo o multiplo tradizionali.

Per ulteriori informazioni su AWS regioni e zone di disponibilità, consulta Global Infrastructure.AWS

Oltre all'infrastruttura AWS globale, HealthLake offre diverse funzionalità per supportare le esigenze di resilienza e backup dei dati.

Sicurezza dell'infrastruttura in AWS HealthLake

In quanto servizio gestito, AWS HealthLake è protetto dalle procedure di sicurezza della rete AWS globale descritte nel white paper Amazon Web Services: Overview of Security Processes.

Utilizzi le API chiamate AWS pubblicate per accedere HealthLake attraverso la rete. I client devono supportare Transport Layer Security (TLS) 1.0 o versione successiva. Consigliamo TLS 1.2 o versioni successive. I client devono inoltre supportare suite di crittografia con Perfect Forward Secrecy (PFS) come Ephemeral Diffie-Hellman () o Elliptic Curve Ephemeral Diffie-Hellman (). DHE ECDHE La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.

Inoltre, le richieste devono essere firmate utilizzando un ID chiave di accesso e una chiave di accesso segreta che è associata a un'entità IAM. O puoi utilizzare AWS Security Token Service (AWS STS) per generare credenziali di sicurezza temporanee per sottoscrivere le richieste.

Best practice per la sicurezza in AWS HealthLake

AWS HealthLake fornisce una serie di funzionalità di sicurezza da considerare durante lo sviluppo e l'implementazione delle proprie politiche di sicurezza. Le seguenti best practice sono linee guida generali e non rappresentano una soluzione di sicurezza completa. Poiché queste best practice potrebbero non essere appropriate o sufficienti per l'ambiente, gestiscile come considerazioni utili anziché prescrizioni.

- Implementa l'accesso con privilegi minimi.
- Quando possibile, usa Customer-Managed-Keys (CMKs) per crittografare i dati. Per ulteriori informazioniCMKs, consulta Amazon Key Management Service.
- Usa Cerca conPOST, non Cerca con GET quando esegui query per PHI o PII nel tuo archivio dati.
- Limita l'accesso a funzioni di controllo sensibili e importanti.
- Quando create risorse tramite l'aggiornamento o l'importazione in bloccoAPIs, non utilizzate PHI né PII includete i nomi degli archivi dati e dei lavori in alcun campo visibile o nell'FHIRID logico ()LID.
- Quando inviate richieste di creazione, lettura, aggiornamento, eliminazione o ricerca, non utilizzatele PHI nell'HTTPintestazione.

- Abilita AWS CloudTrail per controllare AWS HealthLake l'utilizzo e per garantire che non vi siano attività impreviste.
- Consulta le best practice per utilizzare i bucket Amazon S3 in modo sicuro. Per ulteriori informazioni, consulta le best practice di sicurezza nella guida per l'utente di Amazon S3.

AWS HealthLake endpoint e quote

Le seguenti sezioni contengono informazioni sulle AWS HealthLake quote e sugli endpoint. Per le quote regolabili, puoi richiedere un aumento della quota utilizzando la console Service Quotas. Per ulteriori informazioni, consulta Richiesta di un aumento di quota nella Guida per l'utente per Service Quotas.

Endpoint di servizio

La tabella mostra gli endpoint di HealthLake servizio disponibili in una determinata regione.

Nome della regione	Regione	Endpoint	Protocollo	
US East (Ohio)	us-east-2	healthlake.us-east-2.amazonaws.com healthlake-fips.us-east-2.amazonaws.com	HTTPS HTTPS	
US East (N. Virginia)	us-east-1	healthlake.us-east-1.amazonaws.com healthlake-fips.us-east-1.amazonaws.com	HTTPS HTTPS	
US West (Oregon)	us-west-2	healthlake.us-west-2.amazonaws.com healthlake-fips.us-west-2.amazonaws.com	HTTPS HTTPS	
Asia Pacifico (Mumbai)	ap-south-1	healthlake.ap-south-1.amazonaws.com	HTTPS	
Asia Pacific (Sydney)	ap-southeast-2	healthlake.ap-southeast-2.amazonaws.com	HTTPS	
Europa (Londra)	eu-west-2	healthlake.eu-west-2.amazonaws.com	HTTPS	

Quote di servizio per HealthLake

Di seguito sono riportate le quote predefinite per. HealthLake

Nome	Predefinita	Adattabile	Descrizione
Numero di caratteri in una nota medica	Ogni regione supportata: 10.000	No	Il numero massimo di caratteri in una singola nota medica all'interno del tipo di DocumentReference risorsa (POST/PUTrichieste).
Numero di tartFHIRImport lavori S Job simultanei	Ogni regione supportata: 1	No	Il numero massimo di lavori S tartFHIRImport Job simultanei.
Numero di lavori concurrentStart FHIRExportJob	Ogni regione supportata: 1	No	Il numero massimo di lavori S tartFHIRExport Job simultanei.
Numero di archivi dati per account	Ogni regione supportata: 10	Sì	Il numero massimo predefinito di archivi dati attivi per account.
Numero di file in un S tartFHIRImport Job	Ogni regione supportata: 10.000	No	Il numero massimo di file in un S tartFHIRImport Job.
Numero di risorse per pacchetto	Ogni regione supportata: 160	No	Il numero massimo di risorse consentite in una richiesta Bundle.
Frequenza delle richieste di pacchetti per account	Ogni regione supportata: 20	Sì	Il numero massimo di richieste POST Bundle che puoi effettuare al secondo per account.

Nome	Predefinita	Adatta e	Descrizione
Frequenza di richieste Bundle per archivio dati	Ogni regione supportata: 10	Sì	Il numero massimo di richieste POST Bundle che è possibile effettuare al secondo per data store. I data store creati prima del 21/08/2023 saranno limitati a 1 richiesta al secondo.
Frequenza di richieste C ancelFHIR Export Job utilizzate DELETE per account	Ogni regione supportata: 1	No	Il numero massimo di richieste di ancelFHIR Export lavoro C Job DELETE che è possibile effettuare al minuto per account.
Frequenza di reateFHIRDatastore richieste C per account	Ogni regione supportata: 1	No	Il numero massimo di reateFHIRDatastore richieste C che puoi effettuare al minuto per account.
Tasso di DELETE richieste per account	Ogni regione supportata: 2.000	Sì	Il numero massimo di DELETE richieste che puoi effettuare al secondo per account.

Nome	Predefinita	Adatta e	Descrizione
Frequenza di DELETE richieste per archivio dati	Ogni regione supportata: 1.000	Sì	Il numero massimo di DELETE richieste che è possibile effettuare al secondo per archivio dati. Gli archivi dati creati prima del 21/08/2023 saranno limitati a 100 richieste al secondo.
Frequenza di eleteFHIRDatastore richieste D per account	Ogni regione supportata: 1	No	Il numero massimo di eleteFHIRDatastore richieste D che puoi effettuare al minuto per account.
Frequenza di escribeFHIRDatastore richieste D per account	Ogni regione supportata: 10	No	Il numero massimo di escribeFHIRDatasto re richieste D che puoi effettuare al secondo per account.
Percentuale di richieste di escribeFH IRExport lavoro D Job per account	Ogni regione supportata: 10	No	Il numero massimo di richieste D escribeFH IRExport Job che puoi effettuare al secondo per account.
Frequenza di richieste D escribeFH IRExport Job utilizzate GET per account	Ogni regione supportata: 10	No	Il numero massimo di richieste di escribeFH IRExport lavoro D Job GET utilizzabili al secondo per account.

Nome	Predefinita	Adatta e	Descrizione
Percentuale di richieste di escribeFH IRImport lavoro D Job per account	Ogni regione supportata: 10	No	Il numero massimo di richieste D escribeFH IRImport Job che puoi effettuare al secondo per account.
Percentuale di richieste Discovery per account	Ogni regione supportata: 10	No	Il numero massimo di richieste Discovery che puoi effettuare al minuto per account.
Tasso di GET richieste per account	Ogni regione supportata: 6.000	Sì	Il numero massimo di GET richieste che puoi effettuare al secondo per account.
Frequenza di GET richieste per archivio dati	Ogni regione supportata: 3.000	Sì	Il numero massimo di GET richieste che è possibile effettuare al secondo per archivio dati. Gli archivi dati creati prima del 21/08/2023 saranno limitati a 100 richieste al secondo.
Tasso di GetCapabilities richieste per account	Ogni regione supportata: 10	No	Il numero massimo di GetCapabilities richieste che puoi effettuare al secondo per account.
Frequenza di L istFHIRDatastores richieste per account	Ogni regione supportata: 10	No	Il numero massimo di istFHIRDatastores richieste L che puoi effettuare al secondo per account.

Nome	Predefinita	Adatta e	Descrizione
Frequenza di richieste L istFHIRExport Jobs per account	Ogni regione supportata: 10	No	Il numero massimo di richieste L istFHIRExport Jobs che puoi effettuare al secondo per account.
Frequenza di richieste L istFHIRImport Jobs per account	Ogni regione supportata: 10	No	Il numero massimo di richieste L istFHIRImport Jobs che puoi effettuare al secondo per account.
Tasso di ListTagsforResource richieste per account	Ogni regione supportata: 10	No	Il numero massimo di ListTagsforResourc e richieste che puoi effettuare al secondo per account.
Tasso di POST richieste per account	Ogni regione supportata: 2.000	Sì	Il numero massimo di POST richieste che puoi effettuare al secondo per account.
Frequenza di POST richieste per archivio dati	Ogni regione supportata: 1.000	Sì	Il numero massimo di POST richieste che è possibile effettuare al secondo per archivio dati. Gli archivi dati creati prima del 21/08/2023 saranno limitati a 100 richieste al secondo.
Tasso di PUT richieste per account	Ogni regione supportata: 2.000	Sì	Il numero massimo di PUT richieste che puoi effettuare al secondo per account.

Nome	Predefinita	Adatta e	Descrizione
Frequenza di PUT richieste per archivio dati	Ogni regione supportata: 1.000	Sì	Il numero massimo di PUT richieste che è possibile effettuare al secondo per archivio dati. Gli archivi dati creati prima del 21/08/2023 saranno limitati a 100 richieste al secondo.
Percentuale di richieste S tartFHIRE xport Job per account	Ogni regione supportata: 1	No	Il numero massimo di richieste S tartFHIRE xport Job che puoi effettuare al minuto per account.
Frequenza di richieste S tartFHIRExport Job utilizzate POST per account	Ogni regione supportata: 1	No	Il numero massimo di richieste S tartFHIRE xport Job POST che puoi effettuare al minuto per account.
Percentuale di richieste S tartFHIRI mport Job per account	Ogni regione supportata: 1	No	Il numero massimo di richieste S tartFHIRImport Job che puoi effettuare al minuto per account.
Tasso di TagResource richieste per account	Ogni regione supportata: 10	No	Il numero massimo di TagResource richieste che puoi effettuare al secondo.
Tasso di UntagResource richieste per account	Ogni regione supportata: 10	No	Il numero massimo di UntagResource richieste che puoi effettuare al secondo per account.

Nome	Predefinita	Adatta e	Descrizione
Frequenza delle richieste di ricerca utilizzate GET per account	Ogni Regione supportata: 200	Sì	Il numero massimo di richieste di ricerca GET utilizzabili al secondo per account.
Frequenza delle richieste di ricerca utilizzate GET per archivio dati	Ogni regione supportata: 100	Sì	Il numero massimo di richieste di ricerca GET utilizzabili al secondo per archivio dati.
Frequenza delle richieste di ricerca utilizzate POST per account	Ogni Regione supportata: 200	Sì	Il numero massimo di richieste di ricerca POST utilizzabili che è possibile effettuare al secondo.
Frequenza delle richieste di ricerca utilizzate POST per archivio dati	Ogni regione supportata: 100	Sì	Il numero massimo di richieste di ricerca POST utilizzabili al secondo per archivio dati.
Dimensione del singolo file importato	Ogni regione supportata: 5 GB	No	La dimensione massima (in GB) di un singolo file incluso in un S tartFHIRI mport Job.
Dimensione totale del processo di importazione	Ogni regione supportata: 500 Gigabyte	No	La dimensione massima (in GB) di tutti i file inclusi nel processo di importazi one.

Risoluzione dei problemi

La seguente documentazione può aiutarti a risolvere i problemi che potresti riscontrare con l'utilizzo. AWS HealthLake

Argomenti

Perché non posso creare un archivio HealthLake dati?

Il 14 novembre 2022, HealthLake ha aggiornato le IAM autorizzazioni richieste necessarie per creare un nuovo archivio dati. Se non hai aggiornato le politiche allegate all'utente o al ruolo che accede, viene HealthLake visualizzato il seguente errore.

```
AccessDeniedException: Insufficient Lake Formation permission(s): Required Database on
 Catalog
```

Per visualizzare i requisiti IAM delle policy aggiornati per la creazione di un archivio dati, consulta la politica AWS gestita: AmazonHealthLakeFullAccess. Per step-by-step indicazioni su come aggiungere queste politiche al tuo IAM utente o ruolo, vediConfigurazione delle autorizzazioni per iniziare a utilizzare AWS HealthLake.

Per creare un data store, devi anche utilizzare una chiave simmetrica di proprietà del cliente o di proprietà di Amazon. KMS Assicurati di disporre delle autorizzazioni corrette nella tua politica. IAM Per ulteriori informazioni AWS KMS, consulta AWS Key Management Servicela Guida per gli AWS Key Management Service sviluppatori.

È stato superato il numero di archivi dati consentiti per account

HealthLake ha una quota di 10 archivi dati per account. Per informazioni su come richiedere un aumento della quota, visita il AWS Support Center.

Come posso creare l'autorizzazione per FHIR RESTfulAPIs?

Gli utenti devono utilizzare un processo di firma Signature Version 4 per aggiungere l'autenticazione alle HealthLake API richieste inviate tramite un HTTP client. Per ulteriori informazioni, consulta Procedura di firma di Signature Version 4.

Per creare l'autorizzazione sigv4 usando AWS SDK for Python, create uno script simile all'esempio seguente.

```
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())
```

Ulteriori informazioni sull'utilizzo dell'autorizzazione sigv4 utilizzando per AWS SDK Python sono disponibili nell'argomento delle credenziali Boto3.

I miei dati non sono in formato FHIR R4, posso ancora usarli? HealthLake

Solo i dati in formato FHIR R4 possono essere importati in un HealthLake data store. Per un elenco di partner che offrono prodotti per aiutare gli utenti a trasformare i propri dati, consulta AWS HealthLake Partner.

Perché ricevo AccessDenied errori quando utilizzo l'FHIRRESTfulAPIsarchivio dati crittografato con una KMS chiave gestita dal cliente?

Le autorizzazioni sia per la chiave gestita dal cliente che per le IAM policy sono necessarie per consentire a un utente o un ruolo di accedere a un archivio dati. Un utente deve disporre delle IAM autorizzazioni necessarie per utilizzare una chiave gestita dal cliente. Se un utente ha revocato o ritirato una concessione che HealthLake consentiva l'uso della KMS chiave gestita dal cliente, HealthLake restituirà un errore. AccessDenied

HealthLake deve disporre dell'autorizzazione per accedere ai dati dei clienti, crittografare FHIR le nuove risorse importate in un data store e decrittografare le FHIR risorse quando vengono richieste.

Per ulteriori informazioni, consulta Risoluzione dei problemi di accesso tramite chiave.

Perché la mia importazione non è riuscita?

Un processo di importazione riuscito genererà una cartella con i inputFileName file.ndjson di output, tuttavia i singoli record potrebbero non essere importati. Quando ciò accade, verrà generata una seconda FAILURE cartella con un manifesto di record che non sono stati importati. La posizione di output del lavoro per accedere al file manifesto è JobProperties. JobOutputDataConfig.Configurazione S3.S3Uri.

Questo file manifest contiene dettagli sull'output del lavoro, come la posizione di tutte le risposte riuscite (. successOutput successOutputS3Uri), la posizione di tutte le risposte non riuscite (. failureOutput failureOutputS3Uri) e metriche di lavoro aggiuntive. Il contenuto del file manifest è analizzabile a livello di codice. Il seguente file manifesto di esempio elenca i bucket Amazon S3 di input e output, oltre a informazioni sul numero di risorse analizzate e su quante sono state importate correttamente.

```
            {
            "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
            }
```

Per analizzare il motivo per cui un processo di importazione non è riuscito, utilizzare il D escribeFHIRImport Job API per analizzare il JobProperties. Si consiglia quanto segue:

- Se lo stato è FAILED ed è presente un messaggio, gli errori sono correlati a parametri del lavoro, come la dimensione dei dati di input o il numero di file di input, che superano le HealthLake quote.
- Se lo stato del processo di importazione è COMPLETED _ WITH _ERRORS, controllate il file manifest, Manifest.json, per informazioni sui file che non sono stati importati correttamente.
- Se lo stato del processo di importazione è impostato FAILED e non è presente alcun messaggio, vai alla posizione di output del lavoro per accedere al file manifest, Manifest.json.

Per ogni file di input, esiste un file di output di errore con il nome del file di input per ogni risorsa che non riesce a importare. Le risposte contengono il numero di riga (lineId) corrispondente alla posizione dei dati di input, FHIR all'oggetto di risposta (UpdateResourceResponse) e al codice di stato (statusCode) della risposta.

Un file di output di esempio sarebbe simile al seguente:

```
{"lineId":3, UpdateResourceResponse:{"jsonBlob":
{"resourceType":"OperationOutcome","issue":
[{"severity":"error","code":"processing","diagnostics":"1 validation error detected:
 Value 'Patient123' at 'resourceType' failed to satisfy constraint: Member must satisfy
 regular expression pattern: [A-Za-z]{1,256}"}]}, "statusCode":400}
{"lineId":5, UpdateResourceResponse:{"jsonBlob":
{"resourceType":"OperationOutcome","issue":
[{"severity":"error","code":"processing","diagnostics":"This property must be an
 simple value, not a com.google.gson.JsonArray","location":["/EffectEvidenceSynthesis/
name"]},{"severity":"error","code":"processing","diagnostics":"Unrecognised
 property '@telecom'","location":["/EffectEvidenceSynthesis"]},
{"severity":"error","code":"processing","diagnostics":"Unrecognised
 property '@gender'","location":["/EffectEvidenceSynthesis"]},
{"severity":"error","code":"processing","diagnostics":"Unrecognised
 property '@birthDate'","location":["/EffectEvidenceSynthesis"]},
{"severity":"error","code":"processing","diagnostics":"Unrecognised
```

```
 property '@address'","location":["/EffectEvidenceSynthesis"]},
{"severity":"error","code":"processing","diagnostics":"Unrecognised
 property '@maritalStatus'","location":["/EffectEvidenceSynthesis"]},
{"severity":"error","code":"processing","diagnostics":"Unrecognised
 property '@multipleBirthBoolean'","location":["/EffectEvidenceSynthesis"]},
{"severity":"error","code":"processing","diagnostics":"Unrecognised
 property '@communication'","location":["/EffectEvidenceSynthesis"]},
{"severity":"warning","code":"processing","diagnostics":"Name should be usable as an
 identifier for the module by machine processing applications such as code generation
 [name.matches('[A-Z]([A-Za-z0-9_]){0,254}')]","location":["EffectEvidenceSynthesis"]},
{"severity":"error","code":"processing","diagnostics":"Profile http://hl7.org/fhir/
StructureDefinition/EffectEvidenceSynthesis, Element 'EffectEvidenceSynthesis.status':
 minimum required = 1, but only found 0","location":["EffectEvidenceSynthesis"]},
{"severity":"error","code":"processing","diagnostics":"Profile
 http://hl7.org/fhir/StructureDefinition/EffectEvidenceSynthesis,
 Element 'EffectEvidenceSynthesis.population': minimum required
 = 1, but only found 0","location":["EffectEvidenceSynthesis"]},
{"severity":"error","code":"processing","diagnostics":"Profile
 http://hl7.org/fhir/StructureDefinition/EffectEvidenceSynthesis,
 Element 'EffectEvidenceSynthesis.exposure': minimum required =
 1, but only found 0","location":["EffectEvidenceSynthesis"]},
{"severity":"error","code":"processing","diagnostics":"Profile http://
hl7.org/fhir/StructureDefinition/EffectEvidenceSynthesis, Element
 'EffectEvidenceSynthesis.exposureAlternative': minimum required
 = 1, but only found 0","location":["EffectEvidenceSynthesis"]},
{"severity":"error","code":"processing","diagnostics":"Profile http://hl7.org/fhir/
StructureDefinition/EffectEvidenceSynthesis, Element 'EffectEvidenceSynthesis.outcome':
 minimum required = 1, but only found 0","location":["EffectEvidenceSynthesis"]},
{"severity":"information","code":"processing","diagnostics":"Unknown
 extension http://synthetichealth.github.io/synthea/disability-adjusted-
life-years","location":["EffectEvidenceSynthesis.extension[3]"]},
{"severity":"information","code":"processing","diagnostics":"Unknown extension
 http://synthetichealth.github.io/synthea/quality-adjusted-life-years","location":
["EffectEvidenceSynthesis.extension[4]"]}]}, "statusCode":400}
{"lineId":7, UpdateResourceResponse:{"jsonBlob":
{"resourceType":"OperationOutcome","issue":
[{"severity":"error","code":"processing","diagnostics":"2 validation errors detected:
 Value at 'resourceId' failed to satisfy constraint: Member must satisfy regular
 expression pattern: [A-Za-z0-9-.]{1,64}; Value at 'resourceId' failed to satisfy
 constraint: Member must have length greater than or equal to 1"}]}, "statusCode":400}
{"lineId":9, UpdateResourceResponse:{"jsonBlob":
{"resourceType":"OperationOutcome","issue":
[{"severity":"error","code":"processing","diagnostics":"Missing required id field in
 resource json"}]}, "statusCode":400}
```

```
{"lineId":15, UpdateResourceResponse:{"jsonBlob":
{"resourceType":"OperationOutcome","issue":
[{"severity":"error","code":"processing","diagnostics":"Invalid JSON found in input
 file"}]}, "statusCode":400}
```

L'esempio mostra che si sono verificati errori sulle righe 3, 4, 7, 9, 15 dalle corrispondenti righe di input del file di input. Per ognuna di queste righe, le spiegazioni sono le seguenti:

- Nella riga 3, la risposta spiega che il contenuto resourceType fornito nella riga 3 del file di input non è valido.
- Alla riga 5, la risposta spiega che c'è un errore di FHIR convalida nella riga 5 del file di input.
- Alla riga 7, la risposta spiega che c'è un problema di convalida se viene resourceId fornito come input.
- Alla riga 9, la risposta spiega che il file di input deve contenere un ID di risorsa valido.
- Alla riga 15, la risposta del file di input è che il file non è in un JSON formato valido.

Come posso trovare DocumentReference risorse che non possono essere elaborate?

Se una DocumentReference risorsa non era valida, HealthLake fornirà un'estensione che indica un errore di convalida anziché l'NLPoutput medico integrato. Per trovare DocumentReference le risorse che hanno portato a un errore di convalida durante l'NLPelaborazione, i clienti possono utilizzare la funzione HealthLake di ricerca con la chiave di ricerca cm-decoration-statuse il valore VALIDATION di ricerca _. ERROR Questa ricerca elencherà tutte le DocumentReference risorse che hanno portato a errori di convalida, insieme a un messaggio di errore che descrive la natura dell'errore. La struttura del campo di estensione in quelle DocumentReference risorse con errori di convalida sarà simile all'esempio seguente.

```
"extension": [
          {
              "extension": [
                  {
                      "url": "http://healthlake.amazonaws.com/aws-cm/status/",
                      "valueString": "VALIDATION_ERROR"
                  },
```

```
                {
                    "url": "http://healthlake.amazonaws.com/aws-cm/message/",
                    "valueString": "Resource led to too many nested objects after NLP
operation processed the document. 10937 nested objects exceeds the limit of 10000."
                }
            ],
            "url": "http://healthlake.amazonaws.com/aws-cm/"
        }
    ]
```

Un VALIDATION_ ERROR può verificarsi anche se NLP la decorazione crea più di 10.000 oggetti annidati. Quando ciò accade, il documento deve essere suddiviso in documenti più piccoli prima dell'elaborazione.

Migrazione di un data store esistente per utilizzare Amazon Athena

gli archivi di dati creati prima del 14 novembre 2022 sono funzionali, ma non possono essere interrogati in Athena utilizzando. SQL Per interrogare un data store preesistente con Athena, devi prima migrarlo in un nuovo data store.

Per migrare i dati in un nuovo data store

1. Creare un nuovo data store.
2. Esporta i dati dal bucket preesistente in un bucket Amazon S3.
3. Importa i dati nel nuovo data store dal bucket Amazon S3.

L'esportazione di dati in un bucket Amazon S3 comporta un costo aggiuntivo. Il costo aggiuntivo dipende dalla dimensione dei dati esportati.

Connessione dei risultati della ricerca in Athena ad altri servizi AWS

È possibile che si verifichino problemi durante la condivisione dei risultati di ricerca di Athena con altri AWS servizi.

Il problema può verificarsi quando si utilizza `json_extract[1]` come parte di una query di SQL ricerca.

Per risolvere questo problema, è necessario eseguire l'aggiornamento a`CATVAR`.

Potresti riscontrare questo problema quando provi a creare risultati di salvataggio, una tabella (statica) o una vista (dinamica).

La console Athena non funziona dopo l'importazione dei dati in un nuovo archivio dati

Dopo aver importato i dati in un nuovo archivio dati, i dati potrebbero non essere immediatamente disponibili per l'uso. Questo serve per consentire ai dati di essere inseriti nelle tabelle degli iceberg. Riprova in un secondo momento.

Perché ricevo un errore di autorizzazione di Lake Formation: lakeformation: PutDataLakeSettings quando aggiungo un nuovo amministratore di data lake?

Se IAM l'utente o il ruolo contiene la policy `AWSLakeFormationDataAdmin` AWS gestita, non è possibile aggiungere nuovi amministratori di data lake. Verrà visualizzato un errore contenente quanto segue:

```
User arn:aws:sts::111122223333:assumed-role/lakeformation-admin-user is not authorized
 to perform: lakeformation:PutDataLakeSettings on resource: arn:aws:lakeformation:us-
east-2:111122223333:catalog:111122223333 with an explicit deny in an identity-based
 policy
```

La policy AWS gestita `AdministratorAccess` è necessaria per aggiungere un IAM utente o un ruolo come amministratore del data lake AWS Lake Formation. Se IAM l'utente o il ruolo contiene anche `AWSLakeFormationDataAdmin` l'azione, l'azione avrà esito negativo. La politica `AWSLakeFormationDataAdmin` AWS gestita contiene un rifiuto esplicito per l'APIoperazione AWS Lake Formation,. `PutDataLakeSetting`

Anche gli amministratori con accesso completo all'AWSutilizzo della policy `AdministratorAccess` AWS gestita possono essere limitati dalla policy. `AWSLakeFormationDataAdmin`

Come posso attivare la funzionalità integrata HealthLake di elaborazione del linguaggio naturale?

A partire dal 20 febbraio 2023, il comportamento predefinito degli archivi HealthLake dati è cambiato.

Archivi dati attuali: tutti gli archivi di HealthLake dati attuali smetteranno di utilizzare l'elaborazione del linguaggio naturale (NLP) su risorse codificate in base64`DocumentReference`. Ciò significa che `DocumentReference` le nuove risorse non verranno analizzate utilizzando NLP e non verranno generate nuove risorse in base al testo del tipo di risorsa. `DocumentReference` Per `DocumentReference` le risorse esistenti, i dati e le risorse generati tramite NLP rimangono, ma non verranno aggiornati dopo il 20 febbraio 2023.

Nuovi archivi HealthLake dati: gli archivi dati creati dopo il 20 febbraio 2023 non eseguiranno l'elaborazione del linguaggio naturale (NLP) su risorse codificate in base64`DocumentReference`.

Per attivare questa funzionalità è necessario creare un case utilizzando. AWS Support Center Console Per creare il tuo caso, accedi al tuo Account AWS, quindi scegli Crea custodia. Per ulteriori informazioni sulla creazione di un caso e sulla gestione dei casi, consulta Creazione di casi di supporto e gestione dei casi nella Guida per l'Supporto utente.

Lo stato del mio archivio dati non cambia rispetto a Creazione

Se provi a creare un nuovo HealthLake data store e lo stato del tuo data store non cambia da Creazione, devi aggiornare Athena per utilizzare. AWS Glue Data Catalog

Per ulteriori informazioni, consulta l'aggiornamento al AWS Glue Data Catalog step-by-step nella Guida per l'utente di Amazon Athena.

Dopo aver aggiornato correttamente AWS Glue Data Catalog, ora puoi creare un data store.

Per rimuovere il vecchio data store, inizia creando un case usando. AWS Support Center Console Per creare il tuo caso, accedi al tuo Account AWS, quindi scegli Crea caso. Per ulteriori informazioni, consulta Creazione di casi di supporto e gestione dei casi nella Guida Supporto per l'utente.

Lo stato di creazione del mio SDK data store restituisce un'eccezione o uno stato sconosciuto

Aggiorna il tuo SDK alla versione più recente se il data store o la descrizione delle API chiamate al data store restituiscono un'eccezione o lo stato sconosciuto del data store.

La mia FHIR POST API operazione con un documento da 10 MB genera un errore 413Request Entity Too Large. HealthLake

AWS HealthLake ha un API limite sincrono di creazione e aggiornamento di 5 MB per evitare un aumento delle latenze e dei timeout.

È possibile importare documenti di grandi dimensioni, fino a 164 MB, utilizzando il file binario utilizzando l'importazione in blocco. ResourceType API

Cronologia dei documenti per la guida per AWS HealthLake gli sviluppatori

La tabella seguente descrive le modifiche alla documentazione per le AWS HealthLake versioni.

- APIversione: più recente
- Ultimo aggiornamento della documentazione: 25/10/2024

Modifica	Descrizione	Data
HealthLake ora supporti e interazioni FHIR `history`y`vread`	HealthLake ora supporta l'FHIR`history`interazione per il recupero della cronologia di una particolare risorsa e l'`vread`interazione per l'esecuzione di una lettura specifica della versione di una risorsa.	25 ottobre 2024
HealthLake ora supporta nuovi parametri FHIR di ricerca, estensioni e tipi di risorse.	HealthLake ora supporta nuovi parametri FHIR di ricerca, estensioni e tipi di risorse.	9 dicembre 2023
HealthLake ora supporta il FHIR framework SMART on	HealthLake ora supporta la creazione SMART su archivi HealthLake dati FHIR abilitati.	31 maggio 2023
HealthLake ora supporta la convalida del profilo	HealthLake ora supporta la convalida FHIR del profilo.	31 maggio 2023
HealthLake ora supporta `export`	HealthLake ora supporta l'esportazione di file utilizzando l'FHIRRESTAPIoperazione`export`.	31 maggio 2023

Regione Asia Pacifico (Mumbai)	AWS HealthLake è ora disponibile nella regione Asia Pacifico (Mumbai).	4 aprile 2023
L'elaborazione integrata del linguaggio naturale è disattivata	HealthLake ha disattivato l'elaborazione integrata del linguaggio naturale (NLP) su tutti gli archivi di dati a partire dal 20 febbraio 2023.	20 febbraio 2023
HealthLake si integra con Amazon Athena	Ora puoi usare Athena per interrogare gli archivi dati creati dopo il 14 novembre 2022.	14 novembre 2022
La dimensione totale dei lavori di importazione è aumentata	La dimensione totale massima di tutti i file in una richiesta StartFHIRImport Job è ora di 500 GB.	3 ottobre 2022
Supporto in bundle	HealthLake ora supporta il tipo di risorsa Bundle per l'acquisizione di più risorse.	5 agosto 2022
Quote aggiornate per le operazioni in CRUD HealthLake	HealthLake ora supporta limiti più elevati per CRUD le richieste.	14 luglio 2022
Includi il supporto	HealthLake ora supporta le interrogazioni `_include` nei data store.	14 luglio 2022
AWS HealthLake è ora disponibile a livello generale	HealthLake è ora disponibile al pubblico.	30 luglio 2020

AWS Glossario

Per la AWS terminologia più recente, consultate il AWS glossario nella sezione Reference.Glossario AWS

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.