aws

Entwicklerhandbuch

AWS HealthLake

AWS HealthLake: Entwicklerhandbuch

Table of Contents

Was ist AWS HealthLake?

AWS HealthLake ist ein HIPAA geeigneter Service für die Erfassung, Speicherung und Analyse klinischer Daten unter Verwendung der Healthcare Interoperability FHIR (R4) -Spezifikation.

> **Note**
>
> Nach dem 20. Februar 2023 verwenden HealthLake Datenspeicher standardmäßig keine integrierte Verarbeitung natürlicher Sprache (NLP). Wenn Sie daran interessiert sind, diese Funktion in Ihrem Datenspeicher zu aktivieren, finden Sie Wie aktiviere ich die HealthLake integrierte Funktion zur Verarbeitung natürlicher Sprache? weitere Informationen im Kapitel Fehlerbehebung.

Gesundheitsdaten sind häufig unvollständig und inkonsistent. Außerdem sind sie häufig unstrukturiert und enthalten Informationen in klinischen Notizen, Laborberichten, Versicherungsansprüchen, medizinischen Bildern, aufgezeichneten Gesprächen und Zeitreihendaten (z. B. Herz ECG - oder EEG Gehirnspuren).

Gesundheitsdienstleister können sie nutzen, HealthLake um Daten in der Cloud zu speichern, zu transformieren, abzufragen und zu analysieren. AWS Mithilfe der HealthLake integrierten Funktionen zur Verarbeitung natürlicher Sprache (NLP) in der Medizin können Sie unstrukturierten klinischen Text aus verschiedenen Quellen analysieren. HealthLake transformiert unstrukturierte Daten mithilfe von Modellen zur Verarbeitung natürlicher Sprache und bietet leistungsstarke Abfrage- und Suchfunktionen. Sie können HealthLake es verwenden, um Patienteninformationen sicher, konform und überprüfbar zu organisieren, zu indexieren und zu strukturieren.

HealthLake ist auch in Amazon Athena und AWS Lake Formation integriert. Sie können diese Integration verwenden, um Ihren Datenspeicher mithilfe von SQL abzufragen.

Vorteile von AWS HealthLake

Mit AWS HealthLake können Sie:

- Schnelle und einfache Erfassung von Gesundheitsdaten — Sie können lokale Fast Healthcare Interoperability Resources (FHIR) -Dateien, einschließlich klinischer Notizen, Laborberichte, Versicherungsansprüche und mehr, in einen Amazon Simple Storage Service (Amazon

S3) -Bucket in großen Mengen importieren. Sie können die Daten dann in nachgelagerten Anwendungen oder Workflows verwenden.

- FHIRRESTAPIOperationen verwenden — HealthLake unterstützt die Verwendung der FHIR REST API Operationen zur Ausführung von CRUD (Create/Read/Update/Delete) -Vorgängen in Ihrem Datenspeicher. FHIRDie Suche wird ebenfalls unterstützt.
- Speichern Sie Ihre Daten in der AWS Cloud auf sichere, HIPAA geeignete und überprüfbare Weise. Sie können Daten in diesem FHIR Format speichern, sodass sie einfach abgefragt werden können. HealthLake erstellt eine vollständige, chronologische Ansicht der Krankengeschichte jedes Patienten und strukturiert sie im R4-Standardformat. FHIR
- Athena-Integration — HealthLake Durch die Integration mit Athena können Sie leistungsstarke Abfragen SQL erstellen, mit denen Sie komplexe Filterkriterien erstellen und speichern können. Anschließend können Sie diese Daten in nachgelagerten Anwendungen wie SageMaker KI verwenden, um ein Modell für maschinelles Lernen zu trainieren, oder Amazon QuickSight , um Dashboards und Datenvisualisierungen zu erstellen.
- Transformieren Sie unstrukturierte Daten mithilfe spezialisierter Modelle für maschinelles Lernen (ML) — HealthLake bietet integrierte medizinische Verarbeitung natürlicher Sprache (NLP) mithilfe von Amazon Comprehend Medical. Medizinische Rohdaten werden mithilfe spezialisierter ML-Modelle transformiert. Diese Modelle wurden darauf trainiert, unstrukturierte Gesundheitsdaten zu verstehen und aussagekräftige Informationen aus ihnen zu extrahieren. Mit Integrated Medical NLP können Sie automatisch Daten zu Entitäten (z. B. medizinischen Eingriffen und Medikamenten), Entitätsbeziehungen (z. B. ein Medikament und dessen Dosierung) und zu Entitätsmerkmalen (z. B. positives oder negatives Testergebnis oder Zeitpunkt des Eingriffs) aus Ihrem medizinischen Text extrahieren. HealthLake erstellt dann neue Ressourcen auf der Grundlage der Merkmale „Zeichen“, „Symptom“ und „Zustand“. Diese werden als neue Zustands-, Beobachtungs- und MedicationStatement Ressourcentypen hinzugefügt.

HealthLake Anwendungsfälle

Sie können es HealthLake für die folgenden Anwendungen im Gesundheitswesen verwenden:

- Bevölkerungsgesundheitsmanagement — HealthLake unterstützt Organisationen im Gesundheitswesen bei der Analyse von Trends, Ergebnissen und Kosten im Bereich der Bevölkerungsgesundheit. Dies hilft Organisationen dabei, die für eine Patientenpopulation am besten geeignete Intervention zu finden und bessere Optionen für das Pflegemanagement zu wählen.

- Verbesserung der Versorgungsqualität — HealthLake unterstützt Krankenhäuser, Krankenkassen und Organisationen der Biowissenschaften dabei, Versorgungslücken zu schließen, die Qualität der Versorgung zu verbessern und Kosten zu senken, indem ein vollständiger Überblick über die Krankengeschichte eines Patienten erstellt wird.
- Optimierung der Krankenhauseffizienz — HealthLake bietet Krankenhäusern wichtige Tools für Analysen und maschinelles Lernen, um die Effizienz zu verbessern und Krankenhausabfälle zu reduzieren.

Zugreifen HealthLake

Sie können HealthLake über das AWS Management Console, AWS Command Line Interface (AWS CLI) oder das zugreifen AWS SDKs.

1. AWS Management Console — Stellt eine Weboberfläche bereit, über die Sie darauf zugreifen können HealthLake.
2. AWS Command Line Interface (AWS CLI) — Stellt Befehle für eine Vielzahl von AWS Diensten bereit, darunter Windows HealthLake, MacOS und Linux, und wird unter diesen unterstützt. Weitere Informationen zur Installation von finden Sie unter AWS Command Line Interface. AWS CLI
3. AWS SDKs— AWS stellt SDKs (Software Development Kits) bereit, die aus Bibliotheken und Beispielcode für verschiedene Programmiersprachen und Plattformen bestehen (Java, Python, Ruby,. NET, iOS, Android usw.). SDKsSie bieten eine bequeme Möglichkeit, programmatischen Zugriff auf HealthLake und AWS zu erstellen. Weitere Informationen finden Sie unter AWSSDKfür Python.

HIPAAEignung und Datensicherheit

Dies ist ein HIPAA berechtigter Dienst. Weitere Informationen AWS zum US-amerikanischen Health Insurance Portability and Accountability Act von 1996 (HIPAA) und zur Nutzung von AWS Diensten zur Verarbeitung, Speicherung und Übertragung geschützter Gesundheitsinformationen (PHI) finden Sie unter HIPAAÜberblick.

Verbindungen, die personenbezogene Daten (PII) HealthLake enthalten, müssen verschlüsselt sein. Standardmäßig werden alle Verbindungen HTTPS über HealthLake verwendetTLS. HealthLake speichert verschlüsselte Kundeninhalte und arbeitet nach dem Prinzip der AWS gemeinsamen Verantwortung.

Preisgestaltung

Informationen zur HealthLake Preisgestaltung finden Sie auf der Seite mit den AWS HealthLake Preisen. Um die damit verbundenen potenziellen Kosten besser einschätzen zu können HealthLake, können Sie den HealthLake Preisrechner verwenden.

Wie AWS HealthLake funktioniert

AWS HealthLake erstellt einen Datenspeicher, in dem Patientenakten unter Verwendung der Healthcare Interoperability FHIR (R4) -Spezifikation gespeichert werden. Mit HealthLake können Sie die folgenden Aufgaben ausführen.

> **Note**
>
> Nach dem 20. Februar 2023 verwenden HealthLake Datenspeicher standardmäßig keine integrierte Verarbeitung natürlicher Sprache (NLP). Wenn Sie daran interessiert sind, diese Funktion in Ihrem Datenspeicher zu aktivieren, finden Sie Wie aktiviere ich die HealthLake integrierte Funktion zur Verarbeitung natürlicher Sprache? weitere Informationen im Kapitel Fehlerbehebung.

- Datenspeicher erstellen, überwachen und löschen.
- Wird verwendet`StartFHIRImportJob`, um Gesundheitsdaten in großen Mengen aus einem Amazon Simple Storage Service (Amazon S3) -Bucket in einen Datenspeicher zu importieren.
- Verwenden Sie die Vorgänge Erstellen, Lesen, Aktualisieren und Löschen (CRUD), um die in Ihrem Datenspeicher gespeicherten Daten zu verwalten.
- Verwenden Sie es SQL in Amazon Athena, um Ihren Datenspeicher abzufragen.
- Verwenden Sie bei den FHIR REST API Vorgängen einen HTTP Client, um Ihren Datenspeicher zu durchsuchen.
- Ermöglichen Sie Amazon Comprehend MedicalAPI, mithilfe natürlicher Sprachverarbeitung nach medizinischen Erkenntnissen in Ihren Daten zu suchen ()NLP.

Datenspeicher erstellen und überwachen

Mit können Sie Datenspeicher erstellen und überwachen HealthLake, in denen Daten aus Fast Healthcare Interoperability Resources (FHIR) gespeichert werden können.

Um einen neuen Datenspeicher zu erstellen, können Sie C reateFHIRDatastore oder die HealthLake Konsole verwenden. Um den Status eines Datenspeichers zu sehen, verwenden Sie escribeFHIRDatastoreD. Verwenden Sie L, um den Status mehrerer aktiver

Datenspeicher anzuzeigenistFHIRDatastores. Verwenden Sie D, um einen Datenspeicher zu löscheneleteFHIRDatastore.

FHIRRESTAPIOperationen

Sie können die FHIR REST API Operationen verwenden, um Erstellungs-, Lese-, Aktualisierungs- und Löschvorgänge (CRUD) in Ihrem HealthLake Datenspeicher auszuführen. Weitere Informationen darüber, wie die FHIR REST API Operationen HealthLake unterstützt werden, finden Sie unterFHIRRESTAPIInteraktionen mit einem HealthLake Datenspeicher verwenden.

Automatisierte Generierung von Ressourcen aus FHIR DocumentReference Ressourcenerweiterungen

> **Note**
>
> Wenn Sie einen HealthLake Datenspeicher erstellen und Daten hinzufügen, die diese enthaltenDocumentReference, fallen Gebühren auf Ihrem AWS Konto an. Weitere Informationen finden Sie unter AWS HealthLake Preise.

HealthLake stellt keine Dokumente NLP zur Verfügung, die im DocumentReference Ressourcentyp gefunden wurden. HealthLake Verwendet zur Analyse des Textes die folgenden Operationen von Amazon Comprehend MedicalAPI.

- DetectEntitiesV2: Prüft den klinischen Text auf eine Vielzahl von medizinischen Entitäten und gibt spezifische Informationen zu ihnen zurück, z. B. die Kategorie der Entität, den Standort und den Konfidenzwert.
- InferICD10CM: Prüft den klinischen Text, um Erkrankungen als Entitäten zu erkennen, die in einer Patientenakte aufgeführt sind, und verknüpft diese Entitäten mit normalisierten Konzeptidentifikatoren in der ICD -10-CM-Wissensdatenbank der Centers for Disease Control.
- InferRxNorm: Prüft den klinischen Text, um zu erkennen, ob es sich bei Medikamenten um Entitäten handelt, die in einer Patientenakte aufgeführt sind, und verknüpft sie mit den normalisierten Konzeptidentifikatoren in der Datenbank der RxNorm National Library of Medicine.

HealthLake analysiert automatisch die im DocumentReference Ressourcentyp gefundenen Daten, wenn sie Ihrem Datenspeicher hinzugefügt werden. Die ursprünglichen DocumentReference

Ressourcendateien bleiben unverändert. Die extrahierten medizinischen Informationen werden automatisch als FHIR -konforme Erweiterungen angehängt. Weitere Informationen zur NLP Funktionsweise in HealthLake finden Sie unter. Verwenden der automatisierten Ressourcengenerierung auf der Grundlage der Verarbeitung natürlicher Sprache (NLP) des FHIR DocumentReference Ressourcentyps in AWS HealthLake

Suchen Sie mit SQL basierten Abfragen

> **Note**
>
> Bei Datenspeichern, die vor dem 14. November 2022 erstellt wurden, ist Ihre Suche auf die folgenden FHIR REST API Vorgänge beschränkt. Informationen zur Verwendung von SQL basierten Abfragen für Daten in Ihrem HealthLake Datenspeicher finden Sie unterAWS HealthLake Datenspeicher SQL in Amazon Athena abfragen.

Amazon Athena ist ein serverloser SQL Abfrageservice. HealthLake Datenspeicher werden als Apache Iceberg-Tabellen in Athena aufgenommen. Diese Tabellen sind so konzipiert, dass sie große analytische Datensätze unterstützen. In Athena wird jeder FHIR Ressourcentyp als Tabelle dargestellt. Mit Athena können Sie nur `READ` Anfragen für Ihren Datenspeicher stellen. Weitere Informationen zur SQL basierten Suche finden Sie unterFragen Sie Ihren HealthLake Datenspeicher ab mit SQL.

Suchen Sie mit FHIR REST API Operationen

Sie können die in Ihrem Datenspeicher gespeicherten Gesundheitsdaten durchsuchen, indem Sie entweder einen Ressourcentyp mit unterstützten Suchparametern angeben oder indem Sie eine auf dem Server gefundene Ressourcen-ID verwenden, ohne den Ressourcentyp anzugeben. Weitere Informationen zum Suchen mithilfe der FHIR REST API Operationen finden Sie unterFHIRRESTAPIInteraktionen mit einem HealthLake Datenspeicher verwenden.

Aktionen für Datenimporte

Wird verwendet AWS HealthLake , um Ihre Dateien in großen Mengen aus einem Amazon S3 S3-Bucket zu importieren. Verwenden Sie entweder die Konsole oder S tartFHIRImport Job, um einen Importjob zu starten. Nach dem Import Ihrer Dateien können Sie D escribeFHIRImport

Job verwenden, um den Status des Jobs zu überwachen. Nach Abschluss des Importauftrags können die Daten dann zu Athena hinzugefügt, transformiert oder analysiert und in nachgelagerten Anwendungen verwendet werden.

Aktionen für Datenexporte

Wird verwendet HealthLake , um Ihre Dateien in großen Mengen in einen Amazon S3 S3-Bucket zu exportieren. Verwenden Sie entweder die Konsole oder S tartFHIRExport Job, um einen Exportjob zu starten. Nachdem Sie Ihre Dateien exportiert haben, können Sie D escribeFHIRExport Job verwenden, um den Status des Jobs zu überwachen und seine Eigenschaften einzusehen. Nach Abschluss des Exportauftrags können Sie die Daten mithilfe von Amazon visualisieren QuickSight oder mit anderen AWS Diensten darauf zugreifen.

AWS HealthLake unterstützte FHIR Profilvalidierungen

HealthLake unterstützt die FHIRR4-Basisspezifikation. In der R4-Spezifikation sind FHIR Profile enthalten. Profile werden für einen FHIR Ressourcentyp verwendet, um mithilfe von Einschränkungen und/oder Erweiterungen für den Basisressourcentyp eine spezifischere Definition eines Ressourcentyps zu definieren. Ein FHIR Profil kann beispielsweise Pflichtfelder wie Erweiterungen und Wertesätze identifizieren. Eine Ressource kann mehrere Profile unterstützen. Alle HealthLake Datenspeicher unterstützen die Verwendung von FHIR Profilen.

Das Hinzufügen eines FHIR Profils ist nicht erforderlich, wenn Daten zu einem HealthLake Datenspeicher hinzugefügt werden. Wenn beim Hinzufügen oder Aktualisieren einer Ressource kein FHIR Profil angegeben wird, wird die Ressource nur anhand des FHIR R4-Basisschemas validiert.

FHIRProfile, denen eine Ressource entspricht, werden in die Ressource aufgenommen, bevor sie aufgenommen wird. HealthLake HealthLake validiert die angegebenen FHIR Profile, wenn sie Ihrem HealthLake Datenspeicher hinzugefügt werden.

FHIRProfile sind in einem Implementierungsleitfaden angegeben. HealthLake validiert die in den folgenden Implementierungsleitfäden definierten FHIR Profile.

Unterstützte FHIR Profile von HealthLake

Name	Versi	Implementierungsleitfaden	Funktion
US Core	3.1.1	http://hl7.org/fhir/us/core/STU3.1.1/	Standard
US-Kern	4.0.0	https://hl7.org/fhir/us/core/STU4/index.html	Unterstützt
CARINBlauer Knopf	1.1.0	http://hl7.org/fhir/us/carin-bb/STU1.1/	Standard
CARINBlauer Knopf	1.0.0	https://hl7.org/fhir/us/carin-bb/STU1/	Unterstützt
Da Vinci Payer Data Exchange	1.0.0	https://hl7.org/fhir/us/davinci-pdex/	Standard
Austausch von Gesundheitsakten	0.2.0	https://hl7.org/fhir/us/davinci-hrex/2020Sep/	Standard

Name	Versi	Implementierungsleitfaden	Funktion
von Da Vinci () HRex			
DaVinci PDEXNetz planen	1.1.0	https://hl7.org/fhir/us/davinci-pdex-plan-net/STU1.1/	Standard
DaVinci PDEXNetz planen	1.0.0	https://hl7.org/fhir/us/davinci-pdex-plan-net/STU1/	Unterstützt
DaVinci Data Exchange zwischen Kostenträgern (PDex) Arzneimittelformel in den USA	1.1.0	https://hl7.org/fhir/us/davinci-drug-formulary/STU1.1/	Standard
DaVinci Data Exchange zwischen Kostenträgern (PDex) Arzneimittelformel in den USA	1.0.1	https://hl7.org/fhir/us/davinci-drug-formulary/STU1.0.1/	Unterstützt
Digitale Mission Ayushman Bharat der Nationalen Gesundheitsbehörde () ABDM	2.0	https://www.nrces.in/ndhm/fhir/r4/index.html	Standard

Überprüfung der in einer Ressource angegebenen Profile FHIR

Damit ein FHIR Profil validiert werden kann, fügen Sie es mithilfe des im Implementierungsleitfaden URL angegebenen Profils dem `profile` Element der einzelnen Ressourcen hinzu.

FHIRProfile werden validiert, wenn Sie Ihrem Datenspeicher eine neue Ressource hinzufügen. Um eine neue Ressource hinzuzufügen, können Sie den tartFHIRImport API S-Job-Vorgang verwenden,

eine POST Anfrage zum Hinzufügen einer neuen Ressource stellen PUT oder eine vorhandene Ressource aktualisieren.

Example — Um zu sehen, auf welches FHIR Profil in einer Ressource verwiesen wird

Das Profil URL wird dem `profile` Element im `"meta" : "profile"` Schlüssel-Wert-Paar hinzugefügt. Diese Ressource wurde aus Gründen der Übersichtlichkeit gekürzt.

```
            {
             "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 — Wie referenziert man ein nicht standardmäßig unterstütztes Profil FHIR

Zur Validierung anhand eines unterstützten, nicht standardmäßigen Profils (z. B. CarinBB 1.0.0) - füge das Profil URL mit Version (getrennt durch '|') und das Basisprofil URL zum Element hinzu. `meta.profile` Diese Beispielressource wurde aus Gründen der Übersichtlichkeit gekürzt.

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

Vorinstallierte Datentypen

HealthLake wird nur SYNTHEA als vorinstallierter Datentyp unterstützt. Synthea ist ein synthetischer Patientengenerator, der die Krankengeschichte modellgenerierter Patienten modelliert. Es handelt sich HealthLake um ein Open-Source-Git-Repository, mit dem FHIR R4-konforme Ressourcenpakete generiert werden können, sodass Benutzer Modelle testen können, ohne tatsächliche Patientendaten zu verwenden.

Die folgenden Ressourcentypen sind in vorinstallierten Datenspeichern verfügbar.

Unterstützte Synthea-Ressourcentypen

AllergyIntolerance	Ort
CarePlan	MedicationAdministration
CareTeam	MedicationRequest
Antrag	Beobachtung
Bedingung	Organisation
Gerät	Patient
DiagnosticReport	Praktiker
Begegnung	PractitionerRole
ExplanationofBenefit	Verfahren
ImagingStudy	Herkunft
Immunisierung	

Berechtigungen einrichten, um mit der Verwendung zu beginnen AWS HealthLake

In diesem Kapitel richten Sie mit dem die AWS Management Console erforderlichen Berechtigungen ein, um mit der Verwendung AWS HealthLake und Erstellung eines Datenspeichers zu beginnen. Um Berechtigungen für die Erstellung eines Datenspeichers einzurichten, erstellen Sie einen IAM Benutzer oder eine Rolle, der bzw. die ein Data HealthLake Lake-Administrator und -Administrator ist. Sie machen diesen Benutzer in AWS Lake Formation zum Data Lake-Administrator. Der Data Lake-Administrator gewährt Lake Formation Zugriff auf Ressourcen, die für die Verwendung von Amazon Athena zur Abfrage eines Datenspeichers erforderlich sind.

Nachdem Sie einen Datenspeicher in erstellt haben HealthLake, können Sie Berechtigungen für den Import oder Export von Dateien in den Datenspeicher einrichten. Informationen zum Einrichten von Berechtigungen für den Import von Dateien finden Sie unterBerechtigungen für Importjobs einrichten. Informationen zum Einrichten von Berechtigungen für den Export von Dateien finden Sie unterBerechtigungen für Exportaufträge einrichten.

Themen

- Melden Sie sich an für eine AWS-Konto
- Erstellen eines Benutzers mit Administratorzugriff
- Konfigurieren Sie einen IAM Benutzer oder eine Rolle zur Verwendung HealthLake (Administrator) IAM
- Fügen Sie einen Benutzer oder eine Rolle als Data Lake-Administrator in Lake Formation hinzu (IAMAdministrator)

Melden Sie sich an für eine AWS-Konto

Wenn Sie noch keine haben AWS-Konto, führen Sie die folgenden Schritte aus, um eine zu erstellen.

Um sich für eine anzumelden AWS-Konto

1. Öffnen Sie https://portal.aws.amazon.com/billing/die Anmeldung.
2. Folgen Sie den Online-Anweisungen.

   Bei der Anmeldung müssen Sie auch einen Telefonanruf entgegennehmen und einen Verifizierungscode über die Telefontasten eingeben.

Wenn Sie sich für eine anmelden AWS-Konto, Root-Benutzer des AWS-Kontoswird eine erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Als bewährte Sicherheitsmethode weisen Sie einem Administratorbenutzer Administratorzugriff zu und verwenden Sie nur den Root-Benutzer, um Aufgaben auszuführen, die Root-Benutzerzugriff erfordern.

AWS sendet Ihnen nach Abschluss des Anmeldevorgangs eine Bestätigungs-E-Mail. Du kannst jederzeit deine aktuellen Kontoaktivitäten einsehen und dein Konto verwalten, indem du zu https://aws.amazon.com/gehst und Mein Konto auswählst.

Erstellen eines Benutzers mit Administratorzugriff

Nachdem Sie sich für einen angemeldet haben AWS-Konto, sichern Sie Ihren Root-Benutzer des AWS-Kontos AWS IAM Identity Center, aktivieren und erstellen Sie einen Administratorbenutzer, sodass Sie den Root-Benutzer nicht für alltägliche Aufgaben verwenden.

Sichern Sie Ihre Root-Benutzer des AWS-Kontos

1. Melden Sie sich AWS Management Consoleals Kontoinhaber an, indem Sie Root-Benutzer auswählen und Ihre AWS-Konto E-Mail-Adresse eingeben. Geben Sie auf der nächsten Seite Ihr Passwort ein.

   Hilfe bei der Anmeldung mit dem Root-Benutzer finden Sie unter Anmelden als Root-Benutzer im AWS-Anmeldung Benutzerhandbuch zu.
2. Aktivieren Sie die Multi-Faktor-Authentifizierung (MFA) für Ihren Root-Benutzer.

   Anweisungen finden Sie im Benutzerhandbuch unter Aktivieren eines virtuellen MFA Geräts für Ihren AWS-Konto IAM Root-Benutzer (Konsole).

Erstellen eines Benutzers mit Administratorzugriff

1. Aktivieren Sie IAM Identity Center.

   Anweisungen finden Sie unter Aktivieren AWS IAM Identity Center im AWS IAM Identity Center Benutzerhandbuch.
2. Gewähren Sie einem Benutzer in IAM Identity Center Administratorzugriff.

Ein Tutorial zur Verwendung von IAM-Identity-Center-Verzeichnis als Identitätsquelle finden Sie unter Benutzerzugriff mit der Standardeinstellung konfigurieren IAM-Identity-Center-Verzeichnis im AWS IAM Identity Center Benutzerhandbuch.

Anmelden als Administratorbenutzer

- Um sich mit Ihrem IAM Identity Center-Benutzer anzumelden, verwenden Sie die Anmeldung, URL die an Ihre E-Mail-Adresse gesendet wurde, als Sie den IAM Identity Center-Benutzer erstellt haben.

  Hilfe bei der Anmeldung mit einem IAM Identity Center-Benutzer finden Sie im AWS-Anmeldung Benutzerhandbuch unter Anmeldung beim AWS Zugriffsportal.

Weiteren Benutzern Zugriff zuweisen

1. Erstellen Sie in IAM Identity Center einen Berechtigungssatz, der der bewährten Methode zur Anwendung von Berechtigungen mit den geringsten Rechten folgt.

   Anweisungen hierzu finden Sie unter Berechtigungssatz erstellen im AWS IAM Identity Center Benutzerhandbuch.
2. Weisen Sie Benutzer einer Gruppe zu und weisen Sie der Gruppe dann Single Sign-On-Zugriff zu.

   Eine genaue Anleitung finden Sie unter Gruppen hinzufügen im AWS IAM Identity Center Benutzerhandbuch.

Konfigurieren Sie einen IAM Benutzer oder eine Rolle zur Verwendung HealthLake (Administrator) IAM

> Persona: Administrator IAM
>
> Ein Benutzer, der IAM Benutzer und Rollen erstellen und Data Lake-Administratoren hinzufügen kann.

Diese Schritte in diesem Thema müssen von einem IAM Administrator ausgeführt werden.

Um Ihren HealthLake Datenspeicher mit Athena zu verbinden, müssen Sie einen IAM Benutzer oder eine Rolle erstellen, die ein Data Lake-Administrator und ein HealthLake Administrator ist. Dieser neue Benutzer oder diese neue Rolle gewährt Zugriff auf Ressourcen, die sich in einem Datenspeicher über AWS Lake Formation befinden, und die `AmazonHealthLakeFullAccess` AWS verwaltete Richtlinie wird dem Benutzer oder der Rolle hinzugefügt.

> **Important**
>
> Ein IAM Benutzer oder eine Rolle, die ein Data Lake-Administrator ist, kann keine neuen Data Lake-Administratoren erstellen. Um einen weiteren Data Lake-Administrator hinzuzufügen, müssen Sie einen IAM Benutzer oder eine Rolle verwenden, der bzw. der `AdministratorAccess` Zugriff gewährt wurde.

Um einen Administrator zu erstellen

1. Fügen Sie die **`AmazonHealthlakeFullAccess`** IAM AWS verwaltete Richtlinie einem Benutzer oder einer Rolle in Ihrer Organisation hinzu.

   Falls Sie mit der Erstellung eines IAM Benutzers nicht vertraut sind, finden Sie weitere Informationen unter IAM Benutzer erstellen und AWS IAMRichtlinien im Überblick im IAMBenutzerhandbuch.
2. Gewähren Sie dem IAM Benutzer oder der Rolle Zugriff auf AWS Lake Formation.

   - Fügen Sie einem Benutzer oder einer Rolle in Ihrer Organisation die folgende IAM AWS verwaltete Richtlinie hinzu: **`AWSLakeFormationDataAdmin`**

     > **Note**
     >
     > Die `AWSLakeFormationDataAdmin` Richtlinie gewährt Zugriff auf alle Ressourcen von AWS Lake Formation. Es wird empfohlen, immer die Mindestberechtigungen zu verwenden, die für die Ausführung Ihrer Aufgabe erforderlich sind. Weitere Informationen finden Sie unter IAMBewährte Methoden im IAMBenutzerhandbuch.
3. Fügen Sie dem Benutzer oder der Rolle die folgende Inline-Richtlinie hinzu. Weitere Informationen finden Sie unter Inline-Richtlinien im IAMBenutzerhandbuch.

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

Weitere Informationen zu dieser `AWSLakeFormationDataAdmin` Richtlinie finden Sie unter Lake Formation Personas and IAM Permissions Reference im AWS Lake Formation Developer Guide.

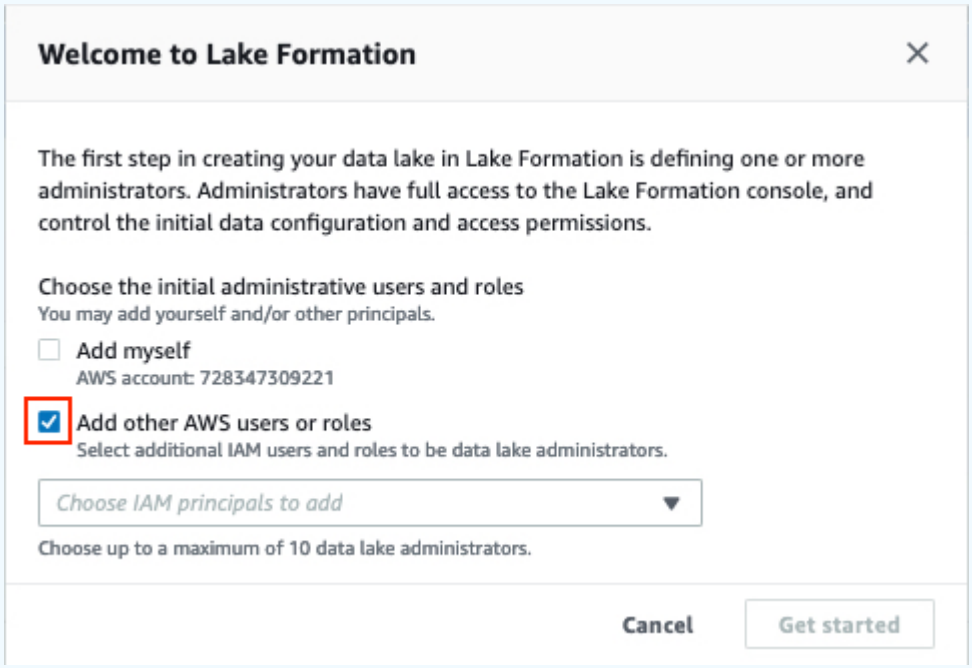
Fügen Sie einen Benutzer oder eine Rolle als Data Lake-Administrator in Lake Formation hinzu (IAMAdministrator)

Als Nächstes muss der IAM Administrator den in Schritt 1 erstellten Benutzer oder die Rolle als Data Lake-Administrator in Lake Formation hinzufügen.

Um einen IAM Benutzer oder eine Rolle als Data Lake-Administrator hinzuzufügen

1. Öffnen Sie die AWS Lake Formation Formation-Konsole: https://console.aws.amazon.com/lakeformation/

> **Note**
>
> Wenn Sie Lake Formation zum ersten Mal besuchen, wird ein Dialogfeld Willkommen bei Lake Formation angezeigt, in dem Sie aufgefordert werden, einen Lake Formation-Administrator zu definieren.



2. Weisen Sie dem neuen Benutzer oder der neuen Rolle die Rolle eines AWS Lake Formation Data Lake-Administrators zu.
   - Option 1: Wenn Sie das Dialogfeld Willkommen bei Lake Formation erhalten haben.
     1. Wählen Sie Weitere AWS Benutzer oder Rollen hinzufügen aus.
     2. Wählen Sie den Abwärtspfeil (▼).
     3. Wählen Sie den HealthLake Administrator aus, der auch Lake Formation-Administratoren sein soll.
     4. Wählen Sie Erste Schritte.
   - Option 2: Verwenden Sie den Navigationsbereich (☰).
     1. Wählen Sie den Navigationsbereich (☰).
     2. Wählen Sie unter Berechtigungen die Option Administrative Rollen und Aufgaben aus.
     3. Wählen Sie im Abschnitt Data Lake-Administratoren die Option Administratoren auswählen aus.

4. Wählen Sie im Dialogfeld Data Lake-Administratoren verwalten den Abwärtspfeil (▼) aus.
5. Wählen Sie als Nächstes die HealthLake Administratoren, Benutzer oder Rollen aus, die Sie auch Lake Formation-Administratoren sein möchten, oder suchen Sie nach ihnen.
6. Wählen Sie Save (Speichern) aus.

3. Ändern Sie die Standardsicherheitseinstellungen, die von Lake Formation verwaltet werden sollen. Die HealthLake Datenspeicherressourcen müssen nicht von Lake Formation verwaltet werdenIAM. Informationen zur Aktualisierung finden Sie unter Ändern des Standardberechtigungsmodells im AWS Lake Formation Developer Guide.

Erstellen eines Datenspeichers in AWS HealthLake

Nachdem Sie den Berechtigungen einrichten, um mit der Verwendung zu beginnen AWS HealthLake Vorgang abgeschlossen haben, können Sie einen Datenspeicher erstellen. In verwenden Sie einen Datenspeicher AWS HealthLake, um Daten im HL7 FHIR (R4) -Format zu speichern. In den Themen dieses Kapitels wird beschrieben, wie Sie einen Datenspeicher erstellen.

Um sowohl analysefähige Datenspeicher zu erstellen als auch Zugriff auf diese in Athena zu gewähren, fügen Sie die `AWSLakeFormationDataAdmin` verwaltete Richtlinie Ihrem IAM Benutzer, Ihrer Gruppe oder Rolle hinzu. Die `AWSLakeFormationDataAdmin` Richtlinie ermöglicht es Ihnen, Data Lake-Administratoren zu erstellen und Zugriff auf Datenspeicher in Athena zu gewähren. Informationen zum Einstellen von Berechtigungen finden Sie unterBerechtigungen einrichten, um mit der Verwendung zu beginnen AWS HealthLake.

HealthLake ist auch in integriert AWS CloudTrail. Sie können CloudTrail es verwenden, um eine Aufzeichnung der Aktionen bereitzustellen, die von einem Benutzer, einer Rolle oder einem AWS Dienst in ausgeführt wurden HealthLake. CloudTrail erfasst alle API Aufrufe und Konsolenaktionen HealthLake als Ereignisse. Weitere Informationen hierzu finden Sie unter Protokollieren von AWS HealthLake API-Aufrufen mit AWS CloudTrail.

Weitere Informationen zu den Ressourcentypen von Fast Healthcare Interoperability Resources (FHIR), die von unterstützt werden HealthLake, finden Sie unterUnterstützte FHIR Ressourcentypen in AWS HealthLake.

> ⚠ Amazon Athena Athena-Kompatibilität
>
> HealthLake Datumsspeicher, die vor dem 14. November 2022 erstellt wurden, können keine SQL Abfragen mit Athena durchführen. Um die Athena-Suchfunktionen für Ihren bereits vorhandenen Datenspeicher zu verwenden, migrieren Sie die Daten zunächst in einen neuen Datenspeicher. Weitere Informationen zur Migration bereits vorhandener Datenspeicher finden Sie unter. Migration eines vorhandenen Datenspeichers zur Verwendung von Amazon Athena

Nachdem Sie einen Datenspeicher erstellt haben, können Sie seine Eigenschaften, einschließlich seines Status, mit den API_DescribeFHIRDatastoreOperationen API_ListFHIRDatastoresoder.html API abrufen. Sie finden den Status des Datenspeichers und andere Details auch auf der Seite Datenspeicher in der HealthLake Konsole.

Ein HealthLake Datenspeicher kann die folgenden Status haben:

- Erstellen — Ihr Datenspeicher wird gerade erstellt.
- Aktiv — Ihr Datenspeicher ist aktiv. Sie können Daten daraus importieren und exportieren. Sie können auch die FHIR Ressourcen verwalten und durchsuchen, die Sie im Datenspeicher gespeichert haben.
- Löschen — Ihr Datenspeicher wird gelöscht.
- Gelöscht — Ihr Datenspeicher wurde gelöscht.

Themen

Einen Datenspeicher erstellen (AWS Management Console)

⚠ HealthLake Unterschiede auf der Konsole

Die HealthLake Konsole unterstützt nicht das Erstellen eines Datenspeichers, SMART der FHIR aktiviert ist. Um einen SMART Datenspeicher mit FHIR aktivierter Aktivierung zu erstellen, müssen Sie den AWS CLI oder einen der AWS unterstützten Datenspeicher verwendenSDKS. Weitere Informationen hierzu finden Sie unter SMARTIntegrieren FHIR mit AWS HealthLake. Außerdem unterscheidet die Konsole nicht zwischen den beiden Arten von Datenspeichern, die unterstützt werden, HealthLake wenn Sie die Detailseite eines einzelnen Datenspeichers aufrufen.

Um einen HealthLake Datenspeicher zu erstellen

1. Öffnen Sie die HealthLake Konsole zu https://console.aws.amazon.com//healthlake/Hause.
2. Öffnen Sie den Navigationsbereich (←).
3. Wählen Sie dann Datenspeicher aus.
4. Wählen Sie als Nächstes Datenspeicher erstellen.
5. Geben Sie im Abschnitt Datenspeicher-Einstellungen für den Namen des Datenspeichers einen Namen an.

6. (Optional) Aktivieren Sie im Bereich Datenspeicher-Einstellungen für Beispieldaten vorab laden das Kontrollkästchen, um Synthea-Daten vorab zu laden.

   - Bei Synthea-Daten handelt es sich um einen vorgeladenen Beispieldatensatz. Weitere Informationen finden Sie unter Vorinstallierte Datentypen.

7. Wählen Sie im Abschnitt Datenspeicherverschlüsselung entweder Eigenen AWS Schlüssel verwenden (Standard) oder Anderen AWS KMS Schlüssel auswählen (erweitert) aus.

8. Im Abschnitt Tags — optional können Sie Ihrem Datenspeicher Tags hinzufügen.

   - Weitere Informationen zum Taggen Ihres Datenspeichers finden Sie unterHinzufügen eines Tags zu einem Datenspeicher.

9. Wählen Sie als Nächstes „Datenspeicher erstellen“. Der Status Ihrer Datenspeicher ist auf der Seite Datenspeicher verfügbar.

Einen Datenspeicher erstellen (AWS CLI und AWS SDKs)

Sie können die folgenden Codebeispiele verwenden, um einen HealthLake Datenspeicher zu erstellen.

AWS CLI

Das folgende Beispiel zeigt die Verwendung der `CreateFHIRDatastore` Operation mit dem AWS CLI. Um das Beispiel auszuführen, müssen Sie den installieren AWS CLI. Wenn Sie Ihren Datenspeicher erstellen, wird für die Verschlüsselung im Ruhezustand standardmäßig ein AWS eigener KMS Schlüssel verwendet, sofern nicht anders angegeben. Weitere Informationen zur Verschlüsselung HealthLake finden Sie REST unter. Verschlüsselung bei REST für AWS HealthLake

Das Beispiel ist für Unix, Linux und macOS formatiert. Ersetzen Sie unter Windows den umgekehrten Schrägstrich (\) als Unix-Fortsetzungszeichen am Ende jeder Zeile durch ein Caret (). ^

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

Bei Erfolg erhalten Sie die folgende Antwort. JSON Wenn Ihr Datenspeicher bereit ist, Daten aufzunehmen, ändert sich der Status auf`ACTIVE`. Weitere Informationen zum Importieren von Daten in Ihren HealthLake Datenspeicher finden Sie unterDateien in einen HealthLake Datenspeicher importieren.

```
{
    "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/"
}
```

Um eine Liste aller Datenspeicher/Datenspeicher anzuzeigen, können Sie den `ListFHIRDataStore` Vorgang verwenden. Sie können auch eine Liste der aktiven Datenspeicher in der HealthLake Konsole sehen.

Python (boto3)

Das folgende Beispiel zeigt, wie Sie mithilfe der `create_fhir_datastore` Operation einen HealthLake Datenspeicher erstellen. Wenn Sie Ihren Datenspeicher erstellen, wird bei der Verschlüsselung im Ruhezustand standardmäßig ein AWS eigener AWS KMS Schlüssel verwendet, sofern nicht anders angegeben. Weitere Informationen zur Verschlüsselung HealthLake finden Sie REST unter. Verschlüsselung bei REST für 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')
```

Ein Datenspeicher kann einen von vier Status haben. `list_fhir_datastores`Dient zum Anzeigen einer Liste Ihrer HealthLake Datenspeicher unabhängig vom Status. Dieses Beispiel zeigt, wie Sie anhand des Status eines Datenspeichers filtern können.

```
import boto3

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

Weitere Informationen finden Sie `list_fhir_datastore`in der Boto3-Dokumentation.

Dateien in einen HealthLake Datenspeicher importieren

Nachdem Sie den Vorgang abgeschlossen habenErstellen eines Datenspeichers in AWS HealthLake, können Sie Dateien aus einem Amazon Simple Storage Service (Amazon S3) -Bucket in den Datenspeicher importieren. Um Dateien zu importieren, starten Sie einen Importauftrag mit der HealthLake Konsole oder dem `StartFHIRImportJob` API Vorgang.

Wenn Sie einen Importauftrag erstellen, geben Sie den Speicherort Ihrer Eingabedaten in Amazon S3, einen Amazon S3 S3-Bucket-Speicherort für Ausgabeprotokolldateien, eine IAM Rolle, die HealthLake Zugriff auf Ihre Buckets gewährt, und einen kundeneigenen AWS Key Management Service Schlüssel AWS an. HealthLake verwendet diesen Schlüssel, um Ihre Daten am Quellspeicherort zu verschlüsseln, und wird verwendet, um sie zu entschlüsseln, damit HealthLake sie importiert werden können. Informationen zum Einrichten von Berechtigungen für Importaufträge finden Sie unter. Berechtigungen für Importjobs einrichten Weitere Informationen zum Erstellen und Verwenden von AWS KMS Schlüsseln finden Sie unter Creating Keys im AWSKey Management Service Developer Guide.

HealthLake akzeptiert Eingabedateien im durch Zeilenumbruch getrennten Format JSON (`.ndjson`), wobei jede Zeile aus einer gültigen Ressource besteht. FHIR Sie können die API Operationen `DescribeFHIRImportJob` und verwenden, um laufende `ListFHIRImportJobs` Importaufträge zu beschreiben und aufzulisten.

HealthLake Generiert für jeden Importauftrag eine `manifest.json` Datei. In diesem Protokoll werden sowohl die Erfolge als auch die Fehlschläge eines Importauftrags beschrieben. HealthLake gibt die Datei in den Amazon S3 S3-Bucket aus, den Sie bei der Erstellung eines Importauftrags angeben. Weitere Informationen finden Sie unter JSONManifest-Datei.

Sie können Import- oder Exportaufträge in die Warteschlange stellen. Diese asynchronen Import- oder Exportaufträge werden nach dem Prinzip FIFO (First In First Out) verarbeitet. Sie können FHIR Ressourcen erstellen, lesen, aktualisieren oder löschen, während ein Import- oder Exportauftrag ausgeführt wird.

Nachdem Sie einen Datenspeicher mit vorgeladenen Daten gefüllt oder Daten importiert haben, können Sie mit der Abfrage Ihres Datenspeichers SQL in Amazon Athena beginnen. Weitere Informationen finden Sie unter AWS HealthLake Datenspeicher SQL in Amazon Athena abfragen.

Themen

- Starten Sie einen Importjob in HealthLake
- JSONManifest-Datei
- Beispiel: Starten und Überwachen von Importaufträgen mit dem AWS CLI

Berechtigungen für Importjobs einrichten

Bevor Sie Dateien in einen Datenspeicher importieren, müssen Sie die HealthLake Erlaubnis für den Zugriff auf Ihre Eingabe- und Ausgabe-Buckets in Amazon S3 erteilen. Um HealthLake Zugriff zu gewähren, erstellen Sie eine IAM Servicerolle für HealthLake, fügen der Rolle eine Vertrauensrichtlinie hinzu, um Rollenübernahmeberechtigungen zu gewähren HealthLake , und fügen der Rolle eine Berechtigungsrichtlinie hinzu, die ihr Zugriff auf Ihre Amazon S3 S3-Buckets gewährt.

Wenn Sie einen Importauftrag erstellen, geben Sie den Amazon-Ressourcennamen (ARN) dieser Rolle für die an`DataAccessRoleArn`. Weitere Informationen zu IAM Rollen und Vertrauensrichtlinien finden Sie unter IAMRollen.

Nachdem Sie die Berechtigungen eingerichtet haben, können Sie Dateien mit einem Importauftrag in Ihren Datenspeicher importieren. Weitere Informationen finden Sie unter Starten Sie einen Importjob in HealthLake.

So richten Sie Importberechtigungen ein

1. Falls dies noch nicht geschehen ist, erstellen Sie einen Amazon S3 S3-Ziel-Bucket für Ausgabeprotokolldateien. Der Amazon S3 S3-Bucket muss sich in derselben AWS Region wie der Service befinden, und Block Public Access muss für alle Optionen aktiviert sein. Weitere Informationen finden Sie unter Blockieren des öffentlichen Zugriffs mit Amazon S3. Für die Verschlüsselung muss auch ein Amazon-eigener oder kundeneigener KMS Schlüssel verwendet werden. Weitere Informationen zur Verwendung von KMS Schlüsseln finden Sie unter Amazon Key Management Service.
2. Erstellen Sie eine Datenzugriffs-Servicerolle für HealthLake und erteilen Sie dem HealthLake Service die Erlaubnis, diese zu übernehmen. Beachten Sie dabei die folgende Vertrauensrichtlinie. HealthLake verwendet dies, um den Amazon S3 S3-Ausgabe-Bucket zu schreiben.

```
{
    "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. Fügen Sie der Datenzugriffsrolle eine Berechtigungsrichtlinie hinzu, die ihr den Zugriff auf den Amazon S3 S3-Bucket ermöglicht. `amzn-s3-demo-bucket`Ersetzen Sie es durch den Namen Ihres Buckets.

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

Starten Sie einen Importjob in HealthLake

Nachdem Sie einen Datenspeicher erstellt und Berechtigungen für Importaufträge eingerichtet haben (Berechtigungen für Importjobs einrichten), können Sie mit dem Import von Dateien mit einem Importauftrag beginnen. Sie können einen Importauftrag entweder mit der AWS HealthLake Konsole oder mit dem AWS HealthLake Importvorgang API starten start-fhir-import-jobAPI.

Themen

- Dateien mit API Operationen importieren
- Einen Importjob starten (Konsole)

Dateien mit API Operationen importieren

Voraussetzungen

Wenn Sie die AWS HealthLake API Operationen verwenden, müssen Sie zuerst eine AWS Identity and Access Management (IAM) -Richtlinie erstellen und sie einer IAM Rolle zuordnen. Weitere Informationen zu IAM Rollen und Vertrauensrichtlinien finden Sie unter IAMRichtlinien und Berechtigungen. Kunden müssen außerdem einen KMS Schlüssel für die Verschlüsselung verwenden. Weitere Informationen zur Verwendung von KMS Keys finden Sie unter Amazon Key Management Service.

Gehen Sie wie folgt vor, um Dateien (API) zu importieren.

1. Laden Sie Ihre Daten in einen Amazon S3 S3-Bucket hoch.

2. Verwenden Sie die `start-fhir-import-job API` API Operation. Wenn Sie den Job starten, geben Sie den Namen des Amazon S3 S3-Buckets an, der die Eingabedateien enthält, den KMS Schlüssel, den Sie für die Verschlüsselung verwenden möchten, und die Konfiguration der Ausgabedaten.
3. Um mehr über einen FHIR Importauftrag zu erfahren, verwenden Sie den describe-fhir-import-jobVorgang, um die ID, den Namen, ARN die Startzeit, die Endzeit und den aktuellen Status des Auftrags abzurufen. Wird verwendet list-fhir-import-job, um alle Importaufträge und ihren Status anzuzeigen.

Einen Importjob starten (Konsole)

Um Dateien mit der Konsole zu importieren, laden Sie Ihre Daten in einen Amazon S3 S3-Bucket hoch.

Gehen Sie wie folgt vor, um Dateien zu importieren.

1. Laden Sie Ihre Daten in einen Amazon S3 S3-Bucket hoch.
2. Öffnen Sie die HealthLake Konsole zu https://console.aws.amazon.com//healthlake/Hause.
3. Gehen Sie zur Datenspeicher-Detailseite für Ihren Datenspeicher und wählen Sie Import aus.
4. Geben Sie Ihren Amazon S3 S3-Bucket an und erstellen oder identifizieren Sie entweder die IAM Rolle und den KMS Schlüssel, den Sie verwenden möchten.
5. Wählen Sie Daten importieren.

JSONManifest-Datei

HealthLake Generiert für jeden Importauftrag eine `manifest.json` Datei. HealthLake gibt die Datei in den Amazon S3 S3-Bucket aus, den Sie bei der Erstellung eines Importauftrags angeben.

Die `manifest.json` Datei beschreibt sowohl die Erfolge als auch die Fehlschläge eines Importjobs. Protokolldateien sind in zwei Ordnern mit dem Namen `SUCCESS` und organisiert`FAILURE`. Eine Ausgabedatei kann vertrauliche Informationen enthalten. Wenn Sie also einen Importauftrag erstellen, müssen Sie sowohl einen Amazon S3 S3-Ausgabe-Bucket als auch einen AWS KMS Schlüssel für die Verschlüsselung angeben.

Im Folgenden finden Sie ein Beispiel für die `manifest.json` Ausgabedatei. Wir empfehlen, diese Datei als ersten Schritt zur Behebung eines fehlgeschlagenen Importauftrags zu verwenden. Sie enthält Einzelheiten zu jeder Datei und zu den Ursachen für den Fehlschlag des Importauftrags.

```
  {
  "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
}
```

Beispiel: Starten und Überwachen von Importaufträgen mit dem AWS CLI

Das folgende Beispiel zeigt, wie Sie mit AWS Command Line Interface dem einen Importjob starten und überwachen können. Sie können auch die start-fhir-import-job API verwenden.

```
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
```

Wenn der Importjob beginnt, erhalten Sie die folgende Bestätigung.

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

Um den Status eines Importauftrags zu überwachen oder seine Konfigurationseigenschaften zu ermitteln, verwenden Sie den Befehl describe-fhir-import-jobAPIoder den AWS CLI Befehl, wie im folgenden Beispiel gezeigt.

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

Als Antwort erhalten Sie die folgenden Informationen.

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

Um eine Liste aller Importaufträge anzuzeigen, verwenden Sie den Befehl list-fhir-import-jobsAPIoder den AWS CLI Befehl, wie im folgenden Beispiel gezeigt. Sie können einen oder mehrere Filter hinzufügen, um die Ergebnisse einzuschränken.

```
    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)
```

Als Antwort erhalten Sie die folgenden Informationen.

```
{
    "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
```

Dateien aus einem HealthLake Datenspeicher exportieren

Nachdem Sie einen Datenspeicher erstellt, Daten gespeichert und importiert haben (oder wenn Sie vorgeladene Beispieldaten verwenden), können Sie die Daten in einen Amazon S3 S3-Bucket exportieren. Gehen Sie wie folgt vor, um HealthLake Daten aus Ihrem Datenspeicher zu exportieren.

- Stellen Sie mithilfe der `StartFHIRExportJob` API Operation AWS SDKs und eine Exportanforderung HealthLake.
  - Dieser Vorgang unterstützt nur das Stellen einer systemweiten Exportanforderung.
- Stellen Sie eine Exportanforderung mit der `export` Syntax unter Verwendung von. HealthLake FHIR REST API
  - Dieser Vorgang unterstützt systemweite, Patienten- und Gruppenexportanfragen. Sie können auch Parameter anwenden, um die Daten in der Exportanforderung weiter zu filtern.

> **Important**
>
> HealthLake SDKExportanfragen mithilfe von `StartFHIRExportJob` API Operation und FHIR REST API Exportanfragen mithilfe von `StartFHIRExportJobWithPost` API Operation haben separate IAM Aktionen. Für jede IAM Aktion, SDK Export mit `StartFHIRExportJob` und FHIR REST API Export mit`StartFHIRExportJobWithPost`, können Berechtigungen zum Erlauben/Verweigern separat behandelt werden. Wenn Sie möchten, dass SDK sowohl FHIR REST API Exporte als auch Exporte eingeschränkt werden, stellen Sie sicher, dass Sie für jede IAM Aktion die entsprechenden Berechtigungen verweigern.

Beide Operationen unterstützen nur den Export Ihrer Dateien in einen Amazon S3 (S3) -Bucket. Alle Dateien aus Ihrem HealthLake Datenspeicher werden als durch Zeilenumbrüche getrennte Dateien JSON (`.ndjson`) exportiert, wobei jede Zeile aus einer gültigen Ressource besteht. FHIR

Für beide Operationen ist eine Servicerolle erforderlich. Darin HealthLake muss der Service Principal definiert sein, und Sie müssen einen Amazon Simple Storage Service (S3) -Bucket definieren, in den Sie Ihre Dateien exportieren möchten. Weitere Informationen hierzu finden Sie unter Berechtigungen für Exportaufträge einrichten.

Sie können Import- oder Exportaufträge in die Warteschlange stellen. Diese asynchronen Import- oder Exportaufträge werden nach dem Prinzip FIFO (First In First Out) verarbeitet. Sie können FHIR Ressourcen erstellen, lesen, aktualisieren oder löschen, während ein Import- oder Exportauftrag ausgeführt wird.

Informationen zum Exportieren von Dateien aus Ihrem HealthLake Datenspeicher finden Sie in den folgenden Abschnitten.

- Berechtigungen für Exportaufträge einrichten
- Exportieren von Dateien aus Ihrem Datenspeicher mit der HealthLake Konsole oder AWS SDKs
- Exportieren von Daten aus Ihrem HealthLake Datenspeicher mit FHIR REST API Operationen

Berechtigungen für Exportaufträge einrichten

Bevor Sie Dateien aus einem Datenspeicher exportieren, müssen Sie die HealthLake Erlaubnis für den Zugriff auf Ihren Ausgabe-Bucket in Amazon S3 erteilen. Um HealthLake Zugriff zu gewähren, erstellen Sie eine IAM Servicerolle für HealthLake, fügen der Rolle eine Vertrauensrichtlinie hinzu, um Rollenübernahmeberechtigungen zu gewähren HealthLake , und fügen der Rolle eine Berechtigungsrichtlinie hinzu, die ihr Zugriff auf Ihren Amazon S3 S3-Bucket gewährt.

Wenn Sie bereits eine Rolle für HealthLake in erstellt habenBerechtigungen für Importjobs einrichten, können Sie sie wiederverwenden und ihr die in diesem Thema aufgeführten zusätzlichen Berechtigungen für Ihren Amazon S3 S3-Exportbucket gewähren. Weitere Informationen zu IAM Rollen und Vertrauensrichtlinien finden Sie unter IAMRichtlinien und Berechtigungen.

> **Important**
>
> HealthLake SDKExportanfragen mithilfe von `StartFHIRExportJob` API Operation und FHIR REST API Exportanfragen mithilfe von `StartFHIRExportJobWithPost` API Operation haben separate IAM Aktionen. Für jede IAM Aktion, SDK Export mit `StartFHIRExportJob` und FHIR REST API Export mit`StartFHIRExportJobWithPost`, können Berechtigungen zum Erlauben/Verweigern separat behandelt werden. Wenn Sie möchten, dass SDK sowohl FHIR REST API Exporte als auch Exporte eingeschränkt werden, stellen Sie sicher, dass Sie für jede IAM Aktion die entsprechenden Berechtigungen verweigern. Wenn Sie Benutzern vollen Zugriff auf gewähren HealthLake, sind keine Änderungen der IAM Benutzerberechtigungen erforderlich.

Der Benutzer oder die Rolle, der bzw. die die Berechtigungen einrichtet, muss berechtigt sein, Rollen zu erstellen, Richtlinien zu erstellen und Richtlinien an Rollen anzuhängen. Die folgende IAM Richtlinie gewährt diese Berechtigungen.

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

Um Exportberechtigungen einzurichten

1. Falls dies noch nicht geschehen ist, erstellen Sie einen Amazon S3 S3-Ziel-Bucket für die Daten, die Sie aus Ihrem Datenspeicher exportieren möchten. Der Amazon S3 S3-Bucket muss sich in derselben AWS Region wie der Service befinden, und Block Public Access muss für alle Optionen aktiviert sein. Weitere Informationen finden Sie unter Blockieren des öffentlichen Zugriffs mit Amazon S3. Für die Verschlüsselung muss auch ein Amazon-eigener oder kundeneigener KMS Schlüssel verwendet werden. Weitere Informationen zur Verwendung von KMS Schlüsseln finden Sie unter Amazon Key Management Service.
2. Falls Sie dies noch nicht getan haben, erstellen Sie eine Datenzugriffs-Servicerolle für HealthLake und erteilen Sie dem HealthLake Service die Erlaubnis, diese zu übernehmen. Beachten Sie dabei die folgende Vertrauensrichtlinie. HealthLake verwendet dies, um den Amazon S3 S3-Ausgabe-Bucket zu schreiben. Wenn Sie bereits einen in erstellt habenBerechtigungen für Importjobs einrichten, können Sie ihn wiederverwenden und ihm im nächsten Schritt Berechtigungen für Ihren Amazon S3 S3-Bucket erteilen.

   ```
   {
       "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. Fügen Sie der Datenzugriffsrolle eine Berechtigungsrichtlinie hinzu, die ihr den Zugriff auf Ihren Amazon S3 S3-Ausgabe-Bucket ermöglicht. `amzn-s3-demo-bucket`Ersetzen Sie es durch den Namen Ihres Buckets.

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

Exportieren von Dateien aus Ihrem Datenspeicher mit der HealthLake Konsole oder AWS SDKs

Nachdem Sie den Vorgang abgeschlossen habenBerechtigungen für Exportaufträge einrichten, können Sie Dateien aus Ihrem Datenspeicher in einen Amazon Simple Storage Service (Amazon S3) -Bucket exportieren. Um Dateien aus einem Datenspeicher zu exportieren, starten Sie einen Exportjob in HealthLake. Ein Exportauftrag exportiert Dateien aus Ihrem Datenspeicher in einem durch Zeilenumbruch getrennten Format JSON (`.ndjson`), wobei jede Zeile aus einer gültigen Ressource besteht. FHIR Wenn Sie einen Exportauftrag starten, müssen Sie einen AWS KMS Schlüssel für die Verschlüsselung angeben. Weitere Informationen zum Erstellen eines KMS Schlüssels finden Sie unter Schlüssel erstellen im AWSKey Management Service Developer Guide.

In den folgenden Themen wird beschrieben, wie Sie einen Exportauftrag mit der AWS HealthLake Konsole und anschließend AWS SDKs mit dem start-fhir-export-jobAPIVorgang starten.

Themen

- Exportieren von Dateien aus Ihrem Datenspeicher (Konsole)
- Exportieren von Dateien aus Ihrem Datenspeicher (AWS SDKs)

Exportieren von Dateien aus Ihrem Datenspeicher (Konsole)

Gehen Sie wie folgt vor, um Dateien zu exportieren (Konsole).

1. Erstellen Sie einen S3-Ausgabe-Bucket in derselben Region wie HealthLake.

2. Um einen neuen Exportauftrag zu starten, identifizieren Sie den Amazon S3 S3-Ausgabe-Bucket und erstellen oder identifizieren Sie die IAM Rolle, die Sie verwenden möchten. Weitere Informationen zu IAM Rollen und Vertrauensrichtlinien finden Sie unter IAMRollen. Verwenden Sie auch eine KMS Schlüsselverschlüsselung. Weitere Informationen zur Verwendung von KMS Schlüsseln finden Sie unter Amazon Key Management Service.
3. Um den Status Ihres Exportauftrags zu sehen, verwenden Sie `ListFHIRExportJobs` API Operation.

Exportieren von Dateien aus Ihrem Datenspeicher (AWS SDKs)

Um Dateien aus Ihrem Datenspeicher mit dem zu exportieren AWS SDKs, verwenden Sie den start-fhir-export-jobVorgang. Der folgende Code zeigt, wie Sie einen Exportjob mit dem SDK für Python (Boto3) starten.

```
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'])
```

Um die ID, den NamenARN, die Startzeit, die Endzeit und den aktuellen Status eines FHIR Exportauftrags abzurufen, verwenden Sie. describe-fhir-export-job Wird verwendet list-fhir-export-jobs, um alle Exportaufträge und ihren Status aufzulisten.

Der folgende Code zeigt, wie Sie die Eigenschaften eines bestimmten Exportjobs mit dem SDK für Python (Boto3) abrufen.

```
import boto3
```

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

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

Exportieren von Daten aus Ihrem HealthLake Datenspeicher mit FHIR REST API Operationen

Nachdem Sie den Vorgang abgeschlossen habenBerechtigungen für Exportaufträge einrichten, können Sie Daten mithilfe von FHIR REST API Vorgängen aus Ihrem HealthLake Datenspeicher exportieren. Um eine Exportanforderung mithilfe von stellen zu können FHIR RESTAPI, müssen Sie über einen IAM Benutzer, eine Gruppe oder eine Rolle mit den erforderlichen Berechtigungen verfügen, diese im `$export` Rahmen der `POST` Anfrage angeben und Anforderungsparameter in den Hauptteil Ihrer Anfrage aufnehmen. Gemäß der FHIR Spezifikation muss der FHIR Server `GET` Anfragen unterstützen und kann `POST` Anfragen unterstützen. Um zusätzliche Parameter zu unterstützen, ist ein Body erforderlich, um den Export zu starten. Daher werden POST Anfragen HealthLake unterstützt.

> **Important**
>
> HealthLake Datenspeicher, die vor dem 1. Juni 2023 erstellt wurden, unterstützen nur FHIR REST API basierte Exportauftragsanforderungen für systemweite Exporte.
> HealthLake Datenspeicher, die vor dem 1. Juni 2023 erstellt wurden, unterstützen nicht das Abrufen des Status eines Exports mithilfe einer `GET` Anfrage am Endpunkt eines Datenspeichers.

Alle Exportanfragen, die Sie mit dem stellen, FHIR REST API werden im `ndjson` Format zurückgegeben und in einen Amazon S3 S3-Bucket exportiert. Jedes S3-Objekt wird nur einen einzigen FHIR Ressourcentyp enthalten.

Sie können Exportanfragen gemäß den AWS Kontingenten in eine Warteschlange stellen. Weitere Informationen zu den damit verbundenen Service Quotas finden Sie unterAWS HealthLake Endpunkte und Kontingente. HealthLake

HealthLake unterstützt die folgenden drei Arten von Massenexport-Endpunktanforderungen.

Typ	Beschreibungen	Syntax
Systemexport	Exportieren Sie alle Daten vom HealthLake FHIR Server.	`POST https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/$export`
Alle Patienten	Exportieren Sie alle Daten, die sich auf alle Patienten beziehen, einschließlich der Ressourcentypen, die dem Ressourcentyp Patient zugeordnet sind.	`POST https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/Patient/$export`
Gruppe von Patienten	Exportieren Sie alle Daten, die sich auf eine Patientengruppe beziehen, für die eine Gruppen-ID angegeben wurde.	`POST https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/Group/ID/$export`

Bevor Sie beginnen

Erfüllen Sie die folgenden Anforderungen, um mithilfe des Formulars FHIR REST API für eine Exportanfrage zu stellen HealthLake.

- Sie müssen einen Benutzer, eine Gruppe oder eine Rolle eingerichtet haben, die über die erforderlichen Berechtigungen verfügt, um die Exportanfrage zu stellen. Weitere Informationen hierzu finden Sie unter Autorisieren einer Anfrage `export`.
- Sie müssen eine Servicerolle erstellt haben, die HealthLake Zugriff auf den Amazon S3 S3-Bucket gewährt, in den Ihre Daten exportiert werden sollen. Die Servicerolle muss auch HealthLake als Dienstprinzipal angegeben werden. Weitere Informationen zum Einrichten von Berechtigungen finden Sie unterBerechtigungen für Exportaufträge einrichten.

Autorisieren einer Anfrage **`export`**

Um eine erfolgreiche Exportanfrage mit dem zu stellen FHIR RESTAPI, autorisieren Sie Ihren Benutzer, Ihre Gruppe oder Rolle, indem Sie entweder IAM oder OAuth2 .0 verwenden. Sie müssen außerdem über eine Servicerolle verfügen.

Autorisieren einer Anfrage mithilfe von IAM

Wenn Sie eine `$export` Anfrage stellen, müssen für den Benutzer, die Gruppe oder die Rolle `CancelFHIRExportJobWithDelete` IAM Aktionen `StartFHIRExportJobWithPostDescribeFHIRExportJobWithGet`, und in der Richtlinie enthalten sein.

> **Important**
>
> HealthLake SDKExportanfragen mithilfe von `StartFHIRExportJob` API Operation und FHIR REST API Exportanfragen mithilfe von `StartFHIRExportJobWithPost` API Operation haben separate IAM Aktionen. Für jede IAM Aktion, SDK Export mit `StartFHIRExportJob` und FHIR REST API Export mit`StartFHIRExportJobWithPost`, können Berechtigungen zum Erlauben/Verweigern separat behandelt werden. Wenn Sie möchten, dass SDK sowohl FHIR REST API Exporte als auch Exporte eingeschränkt werden, stellen Sie sicher, dass Sie für jede IAM Aktion die entsprechenden Berechtigungen verweigern.

Autorisieren einer Anfrage mit SMART on FHIR (OAuth2.0)

Wenn Sie eine `$export` Anfrage für einen SMART FHIR aktivierten HealthLake Datenspeicher stellen, müssen Ihnen die entsprechenden Bereiche zugewiesen werden. Weitere Informationen zu unterstützten Bereichen finden Sie unter. HealthLake FHIRressourcenspezifische Bereiche des Datenspeichers

Eine **`export`** Anfrage stellen

In diesem Abschnitt werden die erforderlichen Schritte beschrieben, die Sie ausführen müssen, wenn Sie eine Exportanfrage mithilfe von stellen FHIR RESTAPI.

Um zu vermeiden, dass Ihr AWS Konto versehentlich belastet wird, empfehlen wir, Ihre Anfragen zu testen, indem Sie eine `POST` Anfrage stellen, ohne die `export` Syntax anzugeben.

Um die Anfrage zu stellen, müssen Sie wie folgt vorgehen:

1. Geben Sie `export` in der `POST` Anfrage URL einen unterstützten Endpunkt an.
2. Geben Sie die erforderlichen Header-Parameter an.
3. Geben Sie einen Anforderungstext an, der die erforderlichen Parameter definiert.

Schritt 1: Geben Sie **`export`** in der **`POST`** Anfrage URL einen unterstützten Endpunkt an

HealthLake unterstützt drei Arten von Massenexport-Endpunktanfragen. Um eine Massenexportanfrage zu stellen, müssen Sie eine `POST` Anfrage auf einem der drei unterstützten Endpunkte stellen. Die folgenden Beispiele zeigen, wie Sie `export` in der Anfrage URL angeben.

- `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 dieser `POST` Anforderungszeichenfolge können Sie die folgenden unterstützten Suchparameter verwenden.

Unterstützte Suchparameter

HealthLake unterstützt die folgenden Suchmodifikatoren in Massenexportanfragen.

Diese Beispiele beinhalten Sonderzeichen, die vor dem Absenden Ihrer Anfrage codiert werden müssen.

Name	Erforderlich?	Beschreibung	Beispiel
`_outputFormat`	Nein	Das Format für die angeforderten Massendatendateien, die generiert werden sollen.	

Name	Erforderlich?	Beschreibung	Beispiel
		Zulässige Werte sind`application/fhir+ndjson` ,`application/ndjson` ,`ndjson`.	
`_type`	Nein	Eine Reihe von durch Kommas getrennten FHIR Ressourcentypen, die Sie in Ihren Exportauftrag aufnehmen möchten. Wir empfehlen die Aufnahme, `_type` da sich dies negativ auf die Kosten auswirken kann, wenn alle Ressourcen exportiert werden.	`&_type=MedicationStatement,Observation`
`_since`	Nein	Ressourcentypen, die am oder nach dem Datums- und Zeitstempel geändert wurden. Wenn für einen Ressourcentyp kein Datum der letzten Aktualisierung angegeben ist, wird er in Ihre Antwort aufgenommen.	`&_since=2024-05-09T00%3A00%3A00Z`

Schritt 2: Geben Sie die erforderlichen Header-Parameter an

Um eine Exportanforderung mit dem zu stellen FHIR RESTAPI, müssen Sie die folgenden zwei Header-Parameter angeben.

- Inhaltstyp: `application/fhir+json`
- Bevorzugen Sie: `respond-async`

Als Nächstes müssen Sie die erforderlichen Elemente im Anfragetext angeben.

Schritt 3: Geben Sie einen Anforderungstext an, der die erforderlichen Parameter definiert.

Für die Exportanforderung ist auch ein Hauptteil im `JSON` Format erforderlich. Der Körper kann die folgenden Parameter enthalten.

Schlüssel	Erforderlich?	Beschreibung	Wert
DataAccessRoleArn	Ja	Ein ARN Mitglied einer HealthLake Servicerolle. Die verwendete Servicerolle muss HealthLake als Dienstprinzipal angegeben werden.	`arn:aws:iam:: 444455556666 :role/your-healthlake-service-role`
JobName	Nein	Der Name der Exportanforderung.	**your-export-job-name**
S3Uri	Ja	Teil eines OutputDataConfig Schlüssels. Der S3 URI des Ziel-Buckets, in den Ihre exportierten Daten heruntergeladen werden.	`s3://DOC-EXAMPLE-DESTINATION-BUCKET/ EXPORT-JOB /`

Schlüssel	Erforderlich?	Beschreibung	Wert
KmsKeyId	Ja	Teil eines OutputDataConfig Schlüssels. Der AWS KMS Schlüssel, ARN der zum Sichern des Amazon S3 S3-Buckets verwendet wurde.	arn:aws:kms: **region-of-bucket:123456789012** :key/**1234abcd-12ab-34cd-56ef-1234567890ab**

Example — Der Hauptteil einer Exportanfrage, die mit dem FHIR REST API

Um eine Exportanfrage mithilfe von zu stellen FHIR RESTAPI, müssen Sie einen Hauptteil angeben, wie im Folgenden dargestellt.

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

Wenn Ihre Anfrage erfolgreich ist, erhalten Sie die folgende Antwort.

Header der Antwort

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

Hauptteil der Antwort

```
{
  "datastoreId": "your-data-store-id",
```

```
  "jobStatus": "SUBMITTED",
  "jobId": "your-export-request-job-id"
}
```

Verwaltung Ihrer Exportanfrage

Nachdem Sie eine erfolgreiche Exportanfrage gestellt haben, können Sie diese Anfrage verwaltenexport, indem Sie den Status einer aktuellen Exportanfrage beschreiben und `export` eine aktuelle Exportanfrage stornieren.

Wenn Sie eine Exportanfrage mithilfe von stornieren RESTAPI, wird Ihnen nur der Teil der Daten in Rechnung gestellt, der bis zum Absenden der Stornierungsanforderung exportiert wurde.

In den folgenden Themen wird beschrieben, wie Sie den Status einer aktuellen Exportanfrage abrufen oder diese stornieren können.

Eine Exportanfrage stornieren

Um eine Exportanfrage zu stornieren, stellen Sie eine `DELETE` Anfrage und geben Sie die Job-ID in der Anfrage anURL.

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

Wenn Ihre Anfrage erfolgreich ist, erhalten Sie Folgendes.

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

Wenn Ihre Anfrage nicht erfolgreich ist, erhalten Sie Folgendes.

```
{
  "resourceType": "OperationOutcome",
  "issue": [
```

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

Beschreibung einer Exportanfrage

Um den Status einer Exportanfrage abzurufen, stellen Sie eine `GET` Anfrage, indem Sie `export` und Ihr verwenden**`export-request-job-id`**.

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

Die JSON Antwort wird ein `ExportJobProperties` Objekt enthalten. Sie kann die folgenden Schlüssel/Wert-Paare enthalten.

Name	Erforderlich?	Beschreibung	Wert
DataAccessRoleArn	Nein	Ein Mitglied ARN einer HealthLake Servicerolle. Die verwendete Servicerolle muss HealthLake als Dienstprinzipal angegeben werden.	`arn:aws:iam:: 444455556666 :role/your-healthlake-service-role`
SubmitTime	Nein	Das Datum, zu dem ein Exportauftrag gesendet wurde.	`Apr 21, 2023 5:58:02`
EndTime	Nein	Die Uhrzeit, zu der ein Exportauftrag abgeschlossen wurde.	`Apr 21, 2023 6:00:08 PM`
JobName	Nein	Der Name der Exportanforderung.	**`your-export-job-name`**

Name	Erforderlich?	Beschreibung	Wert
JobStatus	Nein		Gültige Werte für sind: `SUBMITTED | IN_PROGRESS | COMPLETED_WITH_ERRORS | COMPLETED | FAILED`
S3Uri	Ja	Teil eines OutputDataConfigObjekts. Der Amazon S3 URI des Ziel-Buckets, in den Ihre exportierten Daten heruntergeladen werden.	`s3://DOC-EXAMPLE-DESTINATION-BUCKET/` **`EXPORT-JOB /`**
KmsKeyId	Ja	Teil eines OutputDataConfigObjekts. Der AWS KMS Schlüssel, ARN der zum Sichern des Amazon S3 S3-Buckets verwendet wurde.	`arn:aws:kms:` **`region-of-bucket:123456789012`** `:key/`**`1234abcd-12ab-34cd-56ef-1234567890ab`**

Example : Hauptteil einer Exportanfrage zur Beschreibung, die mit dem FHIR REST API

Bei Erfolg erhalten Sie die folgende JSON Antwort.

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

Löschen eines Datenspeichers in HealthLake

Das Löschen eines Datenspeichers ist ein asynchroner Vorgang. Nach dem Start ändert sich der Status in Löschen. Ein Datenspeicher behält den Status Gelöscht bei, bis alle FHIR Daten aus dem Datenspeicher und die erforderliche zugrunde liegende Infrastruktur ebenfalls entfernt wurden.

Nachdem die Daten und die Infrastruktur entfernt wurden, ändert sich der Status Ihres HealthLake Datenspeichers in Gelöscht. Nach dem Löschen sind die Details zu Ihren Datenspeichern nur verfügbar, wenn Sie die `ListFHIRDataStores` Operationen `DescribeFHIRDataStore` und sieben Tage lang verwenden. Nach sieben Tagen erscheint der gelöschte Datenspeicher nicht mehr in den Ergebnissen.

Um einen Datenspeicher erfolgreich zu löschen, muss die IAM Aktion der IAM Richtlinie des Benutzers, der Gruppe oder der Rolle, die die Anfrage gestellt `glue:DeleteDatabase` hat, hinzugefügt werden. Diese IAM Aktion ist nicht Teil der AWS verwalteten Richtlinie,`AmazonHealthLakeFullAccess`.

Sie können einen Datenspeicher mit dem AWS Management Console AWS SDKs, oder dem löschen AWS CLI.

Themen

Löschen eines Datenspeichers (Konsole)

Um einen Datenspeicher mit der Konsole zu löschen, wählen Sie Ihren Datenspeicher auf der Seite Datenspeicher aus und klicken Sie auf Löschen.

Um einen HealthLake Datenspeicher zu löschen

1. Öffnen Sie die HealthLake Konsole zu https://console.aws.amazon.com//healthlake/Hause.
2. Öffnen Sie den Navigationsbereich (←).
3. Wählen Sie dann Datenspeicher aus.
4. Wählen Sie auf der Seite Datenspeicher die Option neben dem Datenspeicher aus, den Sie löschen möchten.

5. Wählen Sie dann Löschen
6. Geben Sie im Dialogfeld ein, **`delete`** um zu bestätigen, dass Sie den ausgewählten Datenspeicher löschen möchten.
7. Wählen Sie dann Löschen aus. Dann ändert sich der Status Ihres Datenspeichers von Aktiv zu Löschen.

Löschen eines Datenspeichers (AWS SDKsund AWS CLI)

Sie können die folgenden Codebeispiele verwenden, um einen HealthLake Datenspeicher zu löschen.

AWS CLI

Die folgenden Beispiele zeigen die Verwendung der `DeleteFHIRDatastore` Operation mit dem AWS CLI. Um das Beispiel auszuführen, müssen Sie den installieren AWS CLI.

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

Bei Erfolg erhalten Sie die folgende JSON Antwort.

```
{
    "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)

Das AWS SDK for Python unterstützt die `describe_fhir_datastore` Methode, die einen einzigen Parameter akzeptiert`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)
```

Bei Erfolg wird ein Python-Wörterbuch zurückgegeben.

```
{'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}}
```

Um Details zu mehr als einem Datenspeicher gleichzeitig zurückzugeben, verwenden Sie `ListFHIRDatastore`

verwenden Sie den `DeleteFHIRDataStore` Befehl mithilfe von, AWS CLI wie im folgenden Beispiel gezeigt. Sie können einen Datenspeicher auch mit der delete-fhir-datastore APIoder der Konsole löschen. Durch das Löschen eines Datenspeichers werden alle im Datenspeicher und in der zugrunde liegenden Infrastruktur enthaltenen FHIR Ressourcenversionen entfernt. Protokolle,

die sich auf einen gelöschten Datenspeicher beziehen, werden gemäß den HIPAA Richtlinien im Dienstkonto aufbewahrt.

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

Wie in der folgenden JSON Beispielantwort gezeigt, ändert sich der Status auf `DELETING` "", um zu bestätigen, dass der Datenspeicher und sein Inhalt gerade gelöscht werden.

```
{
  "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)"
}
```

FHIRRESTAPIInteraktionen mit einem HealthLake Datenspeicher verwenden

In AWS HealthLake verwenden Sie REST API Interaktionen mit Fast Healthcare Interoperability Resources (FHIR), um FHIR Ressourcen in Ihrem Datenspeicher zu verwalten und zu durchsuchen. FHIRRESTAPIInteraktionen werden verwendet, um Interaktionen zum Erstellen, Lesen, Aktualisieren und Löschen (CRUD) für Ressourcen in einem Datenspeicher durchzuführen. Sie können komplexe Suchzeichenfolgen auch mit einer `POST` HTTP Oder-Anfrage `GET` erstellen, da HealthLake eine Teilmenge der von FHIR -unterstützten Suchoperationen unterstützt wird.

Aus Konformitätsgründen werden FHIR Ressourcentypen anhand der R4-Ressource validiert. HL7 FHIR StructureDefinition Um die entsprechenden Funktionen FHIR eines aktiven HealthLake Datenspeichers zu ermitteln, stellen Sie eine `GET` Anfrage, bei der die in der angegeben `metadata` istURL, wie folgt.

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

Bei erfolgreicher Ausführung erhalten Sie einen `200` HTTP Antwortcode und die Capability Statement für Ihren HealthLake Datenspeicher. Weitere Informationen finden Sie `CapabilityStatement`in der HL7FHIRR4-Dokumentation.

In der folgenden Tabelle sind FHIR Interaktionen aufgeführt, die von AWS HealthLake unterstützt werden.

FHIRInteraktionen, die unterstützt werden von AWS HealthLake

FHIRInteraktion	Beschreibung
Interaktionen mit dem gesamten System	
`capabilities`	Holen Sie sich eine Leistungsbeschreibung für das System
`batch/transaction`	Aktualisieren, erstellen oder löschen Sie eine Reihe von Ressourcen in einer einzigen Interaktion
Interaktionen auf Typebene	

FHIRInteraktion	Beschreibung
`create`	Erstellen Sie eine neue Ressource mit einer vom Server zugewiesenen ID
`search`	Suchen Sie anhand einiger Filterkriterien nach einem Ressourcentyp
`history`	Rufen Sie den Änderungsverlauf für einen bestimmten Ressourcentyp ab
Interaktionen auf Instanzebene	
`read`	Lesen Sie den aktuellen Status einer Ressource
`history`	Lesen Sie die Änderungshistorie für eine bestimmte Ressource
`vread`	Lesen Sie den Status einer bestimmten Version der Ressource
`update`	Aktualisieren Sie eine Ressource anhand ihrer ID (oder erstellen Sie sie, wenn sie neu ist)
`delete`	Ressource löschen

Themen

Unterstützte FHIR Ressourcentypen in AWS HealthLake

In der folgenden Tabelle sind die FHIR R4-Ressourcentypen aufgeführt, die von AWS HealthLake unterstützt werden. Weitere Informationen finden Sie im Ressourcenindex in der HL7FHIRR4-Dokumentation.

FHIRR4-Ressourcentypen werden unterstützt von HealthLake

Account	DetectedIssue	Rechnung	Praktiker
ActivityDefinition	Gerät	Bibliothek	PractitionerRole
AdverseEvent	DeviceDefinition	Verknüpfung	Verfahren
AllergyIntolerance	DeviceMetric	Auflisten	Herkunft
Ernennung	DeviceUseStatement	Ort	Fragebogen
AppointmentResponse	DeviceRequest	Measure (Maß)	QuestionnaireResponse
AuditEvent- Siehe Hinweis	DiagnosticReport	MeasureReport	RelatedPerson
Binär	DocumentManifest	Medien	RequestGroup
BodyStructure	DocumentReference	Medikamente	ResearchStudy
Paket — siehe Hinweis	EffectEvidenceSynthesis	MedicationAdministration	ResearchSubject
CapabilityStatement	Begegnung	MedicationDispense	RiskAssessment
CarePlan	Endpunkt	MedicationKnowledge	RiskEvidenceSynthesis
CareTeam	EpisodeOfCare	MedicationRequest	Plan
ChargeItem	EnrollmentRequest	MedicationStatement	ServiceRequest
ChargeItemDefinition	EnrollmentResponse	MessageHeader	Slot
Antrag	ExplanationOfBenefit	MolecularSequence	Exemplar
ClaimResponse	FamilyMemberHistory	NutritionOrder	StructureDefinition
Kommunikation	Flag	Beobachtung	StructureMap

CommunicationRequest	Ziel	OperationOutcome	Substanz
Composition	Gruppe	Organisation	SupplyDelivery
ConceptMap	GuidanceResponse	OrganizationAffiliation	SupplyRequest
Bedingung	HealthcareService	Parameter	Aufgabe
Zustimmung	ImagingStudy	Patientin	ValueSet
Vertrag	Immunisierung	PaymentNotice	VisionPrescription
Deckung	ImmunizationEvaluation	PaymentReconciliation	VerificationResult
CoverageEligibilityRequest	ImmunizationRecommendation	Person	
CoverageEligibilityResponse	InsurancePlan	PlanDefinition	

⚠ FHIRSpezifikationen und HealthLake

- Mit diesen Ressourcentypen können Sie keine `POST` Anfragen stellen`GET`: Binary OperationOutcome, Bundle und Parameters.
- AuditEvent— Eine AuditEvent Ressource kann erstellt oder gelesen, aber nicht aktualisiert oder gelöscht werden.
- Bundle — Es gibt mehrere Möglichkeiten, Bundle-Anfragen zu HealthLake verwalten. Weitere Details finden Sie unter Verwaltung mehrerer FHIR Ressourcen mit Bundle.
- VerificationResult— Dieser Ressourcentyp wird nur für Datenspeicher unterstützt, die nach dem 09. Dezember 2023 erstellt wurden.

Ausführen von Erstellungs-, Lese-, Aktualisierungs- und Löschvorgängen (CRUD) für HealthLake Datenspeicher

Sie verwenden zwar systemeigene AWS Aktionen, wenn Sie Datenspeicher verwalten, Daten importieren und exportieren, aber Sie verwenden vier FHIR HTTP Hauptoperationen, um FHIR Ressourcen innerhalb eines HealthLake Datenspeichers zu erstellen (POSTGET), zu lesen (PUT), zu aktualisieren (DELETE) und zu löschen (). In den folgenden Themen wird beschrieben, wie Sie mithilfe der FHIR REST API Dienste die Vorgänge Erstellen, Lesen, Aktualisieren und Löschen (CRUD) in Ihrem HealthLake Datenspeicher ausführen. Sie müssen einen Signaturprozess mit Signature Version 4 verwenden, um HealthLake API Anfragen zu authentifizieren, die über einen HTTP Client gesendet werden. Weitere Informationen finden Sie unter Signaturprozess für Signature Version 4 in der Allgemeine AWS-Referenz.

Themen

Eine Ressource erstellen mit **POST**

Sie verwenden eine POST Anfrage, um eine neue Ressource in einem HealthLake Datenspeicher zu erstellen. POSTAnfragen erfordern nicht, dass Sie ein id Element angeben. Die HealthLake Server geben den HTTP Statuscode 201 Erstellt zurück, wenn eine Ressource erfolgreich erstellt wurde.

> **Note**
>
> Wenn Sie eine POST Anfrage für den DocumentReference Ressourcentyp stellen, werden die vorhandenen Erweiterungen nicht geändert. AWS HealthLake Fügt stattdessen die neuen Erweiterungen zusammen mit den vorhandenen Erweiterungen zu Ihrem Datenspeicher hinzu. Weitere Informationen darüber, wie natürliche Sprachverarbeitung (NLP) für den DocumentReference Ressourcentyp HealthLake verwendet wird, um wertvolle medizinische Daten zu extrahieren, finden Sie unterVerwenden der automatisierten

Ressourcengenerierung auf der Grundlage der Verarbeitung natürlicher Sprache (NLP) des FHIR DocumentReference Ressourcentyps in AWS HealthLake.

Example Eine **`Patient`** Ressource mithilfe einer **`POST`** Anfrage erstellen.

Um eine HealthLake POST Datenspeicheranforderung zu erstellen, verwenden Sie den Endpunkt Ihres Datenspeichers und geben Sie einen JSON Anforderungstext an. Um den Endpunkt eines Datenspeichers zu finden, suchen Sie in der HealthLake Konsole unter Datenspeicher oder verwenden Sie die escribeFHIRDatastore Operation D in der AWS HealthLake APIReferenz.

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"
}
```

JSON-Antwort

Um die Erstellung der Patientenressource zu bestätigen, erhalten Sie HTTP den Statuscode 201 Erstellt und die folgende JSON Antwort.

```
{
  "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"
  }
}
```

Lesen einer Ressource mit **GET**

Dieses Beispiel zeigt Ihnen, wie Sie mithilfe einer GET Anfrage eine FHIR Patientenressource lesen.

Example Eine bestimmte **Patient** Ressource mithilfe einer **GET** Anfrage lesen.

Verwenden Sie den Endpunkt Ihres HealthLake Datenspeichers, um eine GET Datenspeicheranforderung zu erstellen. Um den Endpunkt eines Datenspeichers zu finden, suchen Sie in der HealthLake Konsole unter Datenspeicher oder verwenden Sie die escribeFHIRDatastore Operation D in der AWS HealthLake APIReferenz.

Sie müssen auch den Ressourcentyp **Patient** und einen gültigen Bezeichner angeben**2de04858-ba65-44c1-8af1-f2fe69a977d9**.

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

JSON-Antwort

Bei Erfolg erhalten Sie einen 200 HTTP Statuscode und die folgende JSON Antwort.

```
{
```

```
    "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"
}
```

Aktualisierung einer Ressource mit **PUT**

Das folgende Beispiel zeigt Ihnen, wie Sie Details PUT zu einem Patienten im FHIR Ressourcentyp Patient aktualisieren können. Wenn Sie eine PUT Anfrage für eine Ressource stellen, die noch nicht erstellt wurde, wird außerdem eine erste Version erstellt.

Ihre Anfrage gibt entweder einen 200 HTTP Statuscode zurück, wenn die Ressource aktualisiert wurde, oder sie gibt einen 201 HTTP Statuscode zurück, wenn eine neue Ressource erstellt wurde.

> **Note**
>
> Wenn Sie eine PUT Anfrage für den DocumentReference Ressourcentyp stellen, werden die vorhandenen Erweiterungen nicht geändert. AWS HealthLake Fügt stattdessen die neuen Erweiterungen zusammen mit den vorhandenen Erweiterungen zu Ihrem Datenspeicher hinzu. Weitere Informationen darüber, wie natürliche Sprachverarbeitung (NLP) für den DocumentReference Ressourcentyp HealthLake verwendet wird, um wertvolle medizinische Daten zu extrahieren, finden Sie unterVerwenden der automatisierten

Ressourcengenerierung auf der Grundlage der Verarbeitung natürlicher Sprache (NLP) des FHIR DocumentReference Ressourcentyps in AWS HealthLake.

Example Aktualisieren eines **`Patient`** Ressourcentyps mithilfe einer **`PUT`** Anfrage

Wenn Sie eine PUT Anfrage stellen, benötigen Sie den Endpunkt des Datenspeichers, den Namen des Ressourcentyps, den Sie aktualisieren möchten, eine Kennung und einen JSON Anforderungstext.

Wenn Sie eine neue Ressource erstellen, verwendet sie den angegebenen Bezeichner, um die neue Ressource zu erstellen. PUT

PUT Request

Beispielstruktur einer gültigen PUT Anfrage:

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

JSON Request Body

Ein JSON Beispieltext, der zur Aktualisierung der angegebenen Patientenressource verwendet wird.

```
{
  "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"
}
```

JSON-Antwort

Zur Bestätigung der Änderung erhalten Sie als Antwort die folgenden JSON Informationen:

```
{
 "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"
 }
}
```

Bedingtes Update

Die bedingte Aktualisierung ermöglicht die Aktualisierung einer vorhandenen Ressource anhand einiger Identifikations-Suchkriterien und nicht anhand der logischen ID. Wenn der Server dieses Update verarbeitet, führt er eine Suche mithilfe seiner Standardsuchfunktionen für den Ressourcentyp durch, um eine einzige logische ID für diese Anfrage aufzulösen.

Welche Aktion ausgeführt wird, hängt davon ab, wie viele Treffer gefunden wurden:

- Keine Treffer, keine ID im Anfragetext angegeben: Der Server erstellt die Ressource.
- Keine Treffer, die ID wurde angegeben und die Ressource ist noch nicht mit der ID vorhanden: Der Server behandelt die Interaktion als „Update as Create“ -Interaktion.
- Keine Treffer, ID angegeben und bereits vorhanden: Der Server lehnt das Update mit einem `409 Conflict` Fehler ab.
- Ein Treffer, keine Ressourcen-ID angegeben ODER (Ressourcen-ID angegeben und sie stimmt mit der gefundenen Ressource überein): Der Server führt das Update anhand der entsprechenden Ressource wie oben beschrieben durch. Wenn die Ressource aktualisiert wurde, gibt der Server SHALL ein; `200 OK`
- One Match, angegebene Ressourcen-ID, entspricht aber nicht der gefundenen Ressource: Der Server gibt einen `409 Conflict` Fehler zurück, der darauf hinweist, dass die Client-ID-Spezifikation ein Problem war, vorzugsweise mit OperationOutcome
- Mehrere Treffer: Der Server gibt einen `412 Precondition Failed` Fehler zurück, der darauf hinweist, dass die Kriterien des Clients nicht selektiv genug waren, vorzugsweise mit einem OperationOutcome

Example — Aktualisieren Sie eine Patientenressource mit dem Namen Peter, dem Geburtsdatum dem 1. Januar 2000 und der Telefonnummer 1234567890:

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

Löschen einer Ressource mit **DELETE**

Um eine Ressource in Ihrem HealthLake Datenspeicher zu löschen, müssen Sie eine `DELETE` HTTP Anfrage stellen.

Example Löschen eines bestimmten **`Patient`** Ressourcentyps mithilfe einer **`DELETE`** Anfrage.

Verwenden Sie den Endpunkt des Datenspeichers, um eine `DELETE` Anfrage zu erstellen. Um den Endpunkt eines Datenspeichers zu finden, suchen Sie in der HealthLake Konsole unter Datenspeicher oder verwenden Sie die escribeFHIRDatastoreD-Operation aus der AWS HealthLake APIReferenz.

Sie müssen auch den Ressourcentyp und eine gültige ID angeben.

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

HTTP-Antwort

Bei Erfolg erhalten Sie einen `204` HTTP Statuscode, der bestätigt, dass sich die Ressource nicht mehr im Datenspeicher befindet. Wenn eine Löschanforderung fehlschlägt, erhalten Sie einen HTTP Statuscode der Serie 400, der angibt, warum die `DELETE` Anforderung fehlgeschlagen ist.

Verwaltung mehrerer FHIR Ressourcen mit Bundle

In der HL7 FHIR R4-Spezifikation sind Bundles einfach eine Sammlung von Ressourcen. HealthLake unterstützt die Erstellung eines Bundle-Ressourcentyps in einer FHIR REST API Anfrage und die Verwendung einer Bundle-Transaktion, um mehrere CRUD Operationen in einer einzigen FHIR REST API Anfrage auszuführen. In einer Bundle-Transaktion müssen Sie den Bundle-Typ wie `batch` in der FHIR REST API Anfrage angeben.

Alle Bundle-Anfragen werden von aufgezeichnet AWS CloudTrail. Weitere Informationen zur Verwendung von CloudTrail with HealthLake finden Sie unterProtokollieren von AWS HealthLake API-Aufrufen mit AWS CloudTrail.

HL7FHIRR4-Ressourcen (extern)

- Die vollständige Spezifikation finden Sie unter Ressourcentyp: Paket im FHIRDokumentationsindex.
- Informationen zu Batch-Interaktionen mit dem FHIR REST API finden Sie unter Batch-Interaktionen mithilfe von FHIR REST API im FHIR Dokumentationsindex.

In den folgenden Abschnitten wird beschrieben, wie eine FHIR REST API Anfrage strukturiert wird, um entweder eine neue Bundle-Ressource zu erstellen oder Ressourcen mithilfe von Bundle-Transaktionen einzeln zu verarbeiten.

⚠ Unterschiede zwischen der HealthLake Konsole AWS CLI, dem und dem AWS SDKs

Die HealthLake Konsole unterstützt nur Operationen vom Typ Bundle, bei denen der Bundle-Ressourcentyp in der FHIR REST API Anfrage angegeben istURL.

Ausführen mehrerer CRUD Operationen mithilfe von FHIR Bundles

Wenn in Ihrer Anfrage kein Ressourcentyp angegeben istURL, wird die FHIR REST API Anfrage als einzelne Datenspeichertransaktionen analysiert. Jeder im JSON Hauptteil angegebene CRUD Vorgang wird ausgewertet und ein bestimmter HTTP Statuscode zurückgegeben. HealthLake unterstützt den Bundle-Typ`batch`.

Gehen Sie wie folgt vor, um mehrere CRUD Operationen in einer einzigen FHIR REST API Anfrage auszuführen:

Die folgende Liste zeigt gekürzte Teile eines Anforderungstexts, der in einer FHIR REST API Bundle-Anfrage verwendet wird. Einen vollständigen Anfragetext finden Sie unter Erstellen einer Bundle-Anfrage mit mehreren CRUD Vorgängen.

1. Geben Sie in Ihrer `POST` Anfrage keinen Ressourcentyp an:

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

2. Geben Sie im Anfragetext den Bundle-Typ an als `"type": "batch"`
3. Geben Sie im Anfragetext ressourcenspezifische Daten für jede CRUD Interaktion an, beginnend mit dem `resource` Schlüssel.
4. Jede CRUD Operation wird `request` im Anforderungstext wie folgt als angegeben:

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

In der JSON Antwort erhalten Sie einen HTTP Statuscode für jeden in der Anfrage angegebenen CRUD Vorgang.

HealthLake schränkt Bundle-Transaktionen ein

- Weitere Informationen zu den Beschränkungen, die für HealthLake Bundles gelten, finden Sie unterAWS HealthLake Endpunkte und Kontingente.

Im Folgenden finden Sie ein Beispiel für einen Bundle-Vorgang, der mehrere CRUD Operationen enthält.

Example — Erstellen einer Bundle-Anfrage mit mehreren CRUD Vorgängen.

Um eine FHIR REST API Anfrage zu stellen, die mehrere CRUD Operationen ausführt, müssen Sie eine POST Anfrage über Ihren Datenspeicher-Endpunkt stellen und einen JSON Anforderungstext angeben.

Sie finden den Endpunkt Ihres Datenspeichers in der HealthLake Konsole unter Datenspeicher oder mithilfe der escribeFHIRDatastore Operation D in der AWS HealthLake APIReferenz.

POST Request

Stellen Sie eine POST Anfrage über den Endpunkt Ihres Datenspeichers. Verwenden Sie die nächste Registerkarte, JSONRequest Body, um die erforderlichen Elemente des Anforderungstexts zu sehen.

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

JSON Request Body

Im Hauptteil der Anfrage müssen Sie die folgenden Schlüssel/Wert-Paare zusammen mit allen anderen FHIR ressourcenspezifischen Daten zu den einzelnen Anfragen angeben. CRUD Das erste Beispiel zeigt einen gekürzten JSON Anfragetext, der die erforderlichen Elemente hervorhebt. Das zweite Beispiel zeigt einen vollständigen JSON Anfragetext.

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

Hier ist ein vollständiges Beispiel, das die Erstellung eines neuen `Patient` `Medication` Ressourcentyps zeigt.

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

JSON-Antwort

Um die Erstellung der in der Beispiel-Bundle-Transaktion angegebenen Ressourcen zu bestätigen, erhalten 201 Sie für jeden eingeschlossenen CRUD Vorgang HTTP den Statuscode Created. Wenn

ein CRUD Vorgang fehlschlägt, erhalten Sie HTTP den Serienstatus 400, der angibt, warum die einzelne Anforderung fehlgeschlagen ist.

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

Ressourcen als Bundle-Ressourcentyp gruppieren

Um einen neuen Bundle-Ressourcentyp zu erstellen, müssen Sie `Bundle` in der FHIR REST API Anfrage einen gültigen JSON Text angeben und bereitstellen, der die Ressourcen enthält, die Sie gruppieren möchten.

Wenn Bundle in der Anfrage angegeben istURL, wird der Inhalt des JSON Anforderungstextes unverändert in Ihrem HealthLake Datenspeicher gespeichert. Daher können für die einzelnen Ressourcentypen keine CRUD Operationen ausgeführt werden. Bündeln dieses Typs wird eine einzige neue Ressourcen-ID zugewiesen. Da die Ressourcen unverändert gespeichert werden, können Sie für einzelne Ressourcen, die im Ressourcentyp Bundle gespeichert sind, keine `POST` Anfragen stellen `GET` oder Anfragen stellen.

> **Note**
>
> Die HL7 FHIR R4-Spezifikation unterstützt auch das Gruppieren von Ressourcen mithilfe von Group, Composition und List. Wenn Sie diese Ressourcentypen erstellen, sind die einzelnen Ressourcen nicht direkt enthalten. Stattdessen verwenden sie das `Reference` Element, um auf die einzelnen Ressourcen zu verweisen. Die Verwendung dieser Ressourcentypen ermöglicht es Ihnen daher, die einzelnen Ressourcen, die in ihnen enthalten sind, zu ändern.

Um einen `Bundle` Ressourcentyp zu erstellen, müssen Sie ihn in Ihrer `POST` Anfrage angeben und eine JSON Aufzählung der Ressourcen bereitstellen, die Sie einbeziehen möchten.

Example — Erstellen einer Bundle-Ressource mithilfe einer Anfrage **`POST`**

Gehen Sie wie folgt vor, um eine `bundle` Ressource zu erstellen

1. Formatieren FHIR REST API Sie eine Anfrage wie folgt:

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

2. Geben Sie den JSON Text an, der die Ressourcen angibt, die Sie gruppieren möchten. In diesem Beispiel werden zwei Patientenressourcen gruppiert.

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

Durchsuchen Ihres HealthLake Datenspeichers mithilfe der FHIR REST API Operationen

HealthLake unterstützt das Durchsuchen Ihres Datenspeichers mithilfe der im FHIR Standard bereitgestellten REST API Operationen. In diesem Abschnitt finden Sie Beispiele dafür, wie Sie `POST` Anfragen für mehrere verschiedene Ressourcentypen stellen `GET` können.

> **Note**
>
> Für Anfragen, die personenbezogene Daten (PII) oder geschützte Gesundheitsinformationen (PHI) betreffen, wird empfohlen, `POST` Anfragen zu verwenden. In einer `POST` Anfrage PII oder PHI wird als Teil des Anfragetextes hinzugefügt und bei der Übertragung verschlüsselt.

Die FHIR Spezifikation unterstützt mehrere Suchparametertypen, HealthLake unterstützt jedoch nur eine Teilmenge. Weitere Informationen erhalten Sie unter Unterstützte Suchparametertypen und Erweiterte Suchparameter, unterstützt von HealthLake.

Durchsucht Ihren Datenspeicher mithilfe der FHIR REST API Operationen.

- Unterstützte Suchparametertypen
- Erweiterte Suchparameter, unterstützt von HealthLake
  - _include
  - _revinclude
  - _summary
  - _elements
  - _total
  - _sort
  - _count
  - Chaining and Reverse Chaining(_has)
- Unterstützte Suchmodifikatoren
- Unterstützte Suchvergleicher
- Suchparameter werden nicht unterstützt von HealthLake
- Suche mit POST Beispielen
- Suche mit GET Beispielen

Unterstützte Suchparametertypen

Die folgende Tabelle zeigt die unterstützten Suchparametertypen in HealthLake.

Unterstützte Suchparametertypen

Suchparameter	Beschreibung
_id	Ressourcen-ID (nicht vollständigURL)
_lastUpdated	Datum der letzten Aktualisierung. Es liegt im Ermessen des Servers, die Genauigkeit der Grenzen festzulegen.
_tag	Suche nach einem Ressourcen-Tag.
_Profil	Sucht nach allen Ressourcen, die mit einem Profil gekennzeichnet sind.
_Sicherheit	Sucht nach Sicherheitslabels, die auf diese Ressource angewendet wurden.
_Quelle	Sucht danach, woher die Ressource stammt.
_text	Suchen Sie nach der Erzählung der Ressource.
createdAt	Suchen Sie nach der benutzerdefinierten Erweiterung -createdAt.

> **Note**
>
> Die folgenden Suchparameter werden nur für Datenspeicher unterstützt, die nach dem 09. Dezember 2023 erstellt wurden: _security, _source, _text,. createdAt

Die folgende Tabelle zeigt Beispiele dafür, wie Abfragezeichenfolgen auf der Grundlage bestimmter Datentypen für einen bestimmten Ressourcentyp geändert werden können. Aus Gründen der Übersichtlichkeit wurden Sonderzeichen in der Spalte mit den Beispielen nicht codiert. Um eine

erfolgreiche Abfrage durchzuführen, stellen Sie sicher, dass die Abfragezeichenfolge richtig codiert wurde.

Typen von Suchparametern	Details	Beispiele

Zahl	Sucht in einer angegebenen Ressource nach einem numerischen Wert. Signifikante Zahlen werden beobachtet.
	
	Die Anzahl der signifikanten Ziffern ist spezifisch für den Suchparameterwert, ausgenommen führende Nullen.
	
	Vergleichspräfixe sind zulässig.
		`[parameter]=100`
		
		`[parameter]=1e2`
		
		`[parameter]=lt100`

Datum/ DateTime	Sucht nach einem bestimmten Datum oder einer bestimmten Uhrzeit. Das erwartete Format ist `yyyy-mm-ddThh:mm:ss[Z|(+|-)hh:mm]` , kann aber variieren.
	
	Akzeptiert die folgenden Datentypen: `date`, `dateTime`, `instant`, `Period` und `Timing`. Weitere Informationen zur Verwendung dieser Datentypen bei Suchen finden Sie unter Datum im FHIRDokumentationsindex.
	
	Vergleichspräfixe sind zulässig.
		`[parameter]=eq2013-01-14`
		
		`[parameter]=gt2013-01-14T10:00`
		
		`[parameter]=ne2013-01-14`

Typen von Suchparametern	Details	Beispiele

String	Sucht nach einer Zeichenfolge unter Berücksichtigung von Groß- und Kleinschreibung.
	
	Unterstützt beide `HumanName` `Address` Typen. Weitere Informationen finden Sie unter den HumanName Datentypeinträgen und den `Address`Datentypeinträgen im FHIRDokumentationsindex.
	
	Die erweiterte Suche wird mithilfe von `:text` Modifikatoren unterstützt.
		`[base]/Patient?given=eve`
		
		`[base]/Patient?given:contains=eve`

Typen von Suchparametern	Details	Beispiele
Token	Sucht nach einer close-to-exact Übereinstimmung mit einer Zeichenfolge und wird häufig mit einem Paar medizinischer Codewerte verglichen. Die Berücksichtigung von Groß- und Kleinschreibung hängt vom Codesystem ab, das bei der Erstellung einer Abfrage verwendet wird. Subsumtionsbasierte Abfragen können dazu beitragen, Probleme im Zusammenhang mit der Berücksichtigung von Groß- und Kleinschreibung zu reduzieren. Aus Gründen der Übersichtlichkeit wurde das | nicht codiert.	`[parameter]=[system]|[code]` : `[system]` Bezieht sich hier auf ein Kodierungssystem und `[code]` bezieht sich auf den Codewert, der in diesem spezifischen System gefunden wurde. `[parameter]=[code]` : Hier entspricht Ihre Eingabe entweder einem Code oder einem System. `[parameter]=|[code]` : Hier entspricht Ihre Eingabe einem Code, und die Systemeigenschaft hat keinen Bezeichner.

Zusammengesetzt	Sucht mithilfe der Modifikatoren $ und der , Operation nach mehreren Parametern innerhalb eines einzigen Ressourcentyps.
	
	Vergleichspräfixe sind zulässig.
		`/Patient?language=FR,NL&language=EN`
		
		`Observation?component-code-value-quantity=http://loinc.org|8480-6$lt60`
		
		`[base]/Group?characteristic-value=gender$mixed`

Typen von Suchparametern	Details	Beispiele

Quantity (Menge)	Sucht nach einer Zahl, einem System und einem Code als Werte. Eine Zahl ist erforderlich, System und Code sind jedoch optional. Basiert auf dem Datentyp Menge. Weitere Informationen finden Sie unter Menge im FHIRDokumentationsindex.
	
	Verwendet die folgende angenommene Syntax `[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`

Referenz	Sucht nach Verweisen auf andere Ressourcen.	`[base]/Observation?subject=Patient/23`
		
		`test`

URI	Sucht nach einer Zeichenfolge, die eine bestimmte Ressource eindeutig identifiziert.	`[base]/ValueSet?url=http://acme.org/fhir/ValueSet/123`
Spezial	Suchanfragen basieren auf integrierten medizinischen Erweiterungen. NLP	

Erweiterte Suchparameter, unterstützt von HealthLake

HealthLake unterstützt die folgenden erweiterten Suchparameter.

Name	Beschreibung	Beispiel	Funktion
`_includ`	Wird verwendet, um anzufordern, dass zusätzliche Ressourcen in einer Suchanfrage zurückgegeben werden. Es gibt Ressourcen zurück, auf die von der Zielressourceninstanz verwiesen wird.	`Encounter?_include=Encounter:subject`	
`_revinc` `de`	Wird verwendet, um anzufordern, dass zusätzliche Ressourcen in einer Suchanfrage zurückgegeben werden. Es gibt Ressourcen zurück, die auf die primäre Ressourceninstanz verweisen.	`Patient?_id=` **`patient-identifier`** `&_revinclude=Encounter:patient`	
`_summar`	Die Zusammenfassung kann verwendet werden, um eine Teilmenge der Ressource anzufordern.	`Patient?_summary=text`	Die folgenden Zusammenfassungsparameter werden unterstützt:`_summary=true` ,, `_summary=false` `_summary=text` ,`_summary=data` .
`_elemen`	Fordert an, dass ein bestimmter Satz von Elementen als Teil einer Ressource in den Suchergebnissen zurückgegeben wird.	`Patient?_elements=identifier,active,link`	

Name	Beschreibung	Beispiel	Funktion
`_total`	Gibt die Anzahl der Ressourcen zurück, die den Suchparametern entsprechen.	`Patient?_total=accurate`	Support `_total=accurate` ,`_total=none` .
`_sort`	Geben Sie die Sortierreihenfolge der zurückgegebenen Suchergebnisse mithilfe einer durch Kommas getrennten Liste an. Das `-` Präfix kann für jede Sortierregel in der kommagetrennten Liste verwendet werden, um die absteigende Reihenfolge anzugeben.	`Observation?_sort=status,-date`	Support Sortierung nach Feldern mit Typen`Number, String, Quantity, Token, URI, Reference` . Sortieren nach `Date` wird nur für Datenspeicher unterstützt, die nach dem 09. Dezember 2023 erstellt wurden. Support bis zu 5 Sortierregeln.
`_count`	Steuern Sie, wie viele Ressourcen pro Seite des Suchpakets zurückgegeben werden.	`Patient?_count=100`	Die maximale Seitengröße beträgt 100.
`chainin`	Suchen Sie nach Elementen von Ressourcen, auf die verwiesen wird. Der `.` leitet die verkettete Suche auf das Element in der referenzierten Ressource weiter.	`DiagnosticReport?subject:Patient.name=peter`	
`reverse chainin (_has)`	Sucht auf der Grundlage der Ressourcenelemente, die auf sie verweisen, nach einer Ressource.	`Patient?_has:Observation:patient:code=1234-5`	

_include

Durch die Verwendung `_include` in einer Suchabfrage können auch zusätzliche angegebene FHIR Ressourcen zurückgegeben werden. Wird verwendet`_include`, um Ressourcen einzubeziehen, die weiterverknüpft sind.

Example — Wird verwendet**`_include`**, um die Patienten oder die Gruppe von Patienten zu finden, bei denen Husten diagnostiziert wurde

Sie würden nach dem `Condition` Ressourcentyp suchen, indem Sie den Diagnosecode für Husten `_include` angeben und dann angeben, dass auch die `subject` Diagnose zurückgegeben werden soll. Beim `Condition` Ressourcentyp `subject` handelt es sich entweder um den Ressourcentyp für den Patienten oder den Ressourcentyp für die Gruppe.

Aus Gründen der Übersichtlichkeit wurden Sonderzeichen im Beispiel nicht codiert. Um eine erfolgreiche Abfrage durchzuführen, stellen Sie sicher, dass die Abfragezeichenfolge richtig codiert wurde.

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

_revinclude

Durch die Verwendung `_revinclude` in einer Suchabfrage können auch zusätzliche angegebene FHIR Ressourcen zurückgegeben werden. Wird verwendet`_revinclude`, um Ressourcen einzubeziehen, die rückwärts verknüpft sind.

Example — Wird verwendet**`_revinclude`**, um verwandte Ressourcentypen „Begegnung“ und „Beobachtung“ einzubeziehen, die mit einem bestimmten Patienten verknüpft sind

Um diese Suche durchzuführen, würden Sie zunächst die Person definieren, `Patient` indem Sie ihre Kennung im `_id` Suchparameter angeben. Anschließend würden Sie mithilfe der Struktur `Encounter:patient` und zusätzliche FHIR Ressourcen angeben0`bservation:patient`.

Aus Gründen der Übersichtlichkeit wurden die Sonderzeichen im Beispiel nicht codiert. Um eine erfolgreiche Abfrage durchzuführen, stellen Sie sicher, dass die Abfragezeichenfolge richtig codiert wurde.

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

```
Patient?_id=patient-
identifier&_revinclude=Encounter:patient&_revinclude=Observation:patient
```

_summary

Die Verwendung `_summary` in einer Suchabfrage ermöglicht es dem Benutzer, eine Teilmenge der FHIR Ressource anzufordern. Es kann einen der folgenden Werte enthalten:`true, text, data, false`. Alle anderen Werte werden als ungültig behandelt. Die zurückgegebenen Ressourcen werden `'SUBSETTED'` in meta.tag mit gekennzeichnet, um darauf hinzuweisen, dass die Ressourcen unvollständig sind.

- `true`: Gibt alle unterstützten Elemente zurück, die in der Basisdefinition der Ressource (n) als „Zusammenfassung“ gekennzeichnet sind.
- `text`: Gibt nur die Elemente 'text', 'id', 'meta' und nur die obligatorischen Elemente der obersten Ebene zurück.
- `data`: Gibt alle Teile außer dem Element „Text“ zurück.
- `false`: Gibt alle Teile der Ressource (n) zurück

Kann in einer einzelnen Suchanfrage `_summary=text` nicht mit `_include` oder mit `_revinclude` Suchparametern kombiniert werden.

Example — Ruft das „Text“ -Element der Patientenressourcen in einem Datenspeicher ab.

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

_elements

Durch die Verwendung `_elements` in einer Suchabfrage können bestimmte FHIR Ressourcenelemente angefordert werden. Die zurückgegebenen Ressourcen werden `'SUBSETTED'` in meta.tag mit gekennzeichnet, um darauf hinzuweisen, dass die Ressourcen unvollständig sind.

Der `_elements` Parameter besteht aus einer durch Kommas getrennten Liste von Basiselementnamen, z. B. Elementen, die auf der Stammebene der Ressource definiert sind. Es dürfen nur die aufgelisteten Elemente zurückgegeben werden. Wenn `_elements` Parameterwerte ungültige Elemente enthalten, ignoriert der Server sie und gibt obligatorische Elemente und gültige Elemente zurück.

`_elements`gilt nicht für eingeschlossene Ressourcen (zurückgegebene Ressourcen, deren Suchmodus ist`include`).

Kann in einer einzelnen Suchanfrage `_elements` nicht mit `_summary` Suchparametern kombiniert werden.

Example — Ruft die Elemente „Identifier“, „Active“ und „Link“ von Patientenressourcen in Ihrem HealthLake Datenspeicher ab.

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

`_total`

Bei Verwendung `_total` in einer Suchabfrage wird die Anzahl der Ressourcen zurückgegeben, die den angeforderten Suchparametern entsprechen. HealthLake gibt die Gesamtzahl der übereinstimmenden Ressourcen (zurückgegebene Ressourcen, deren Suchmodus ist`match`) in `Bundle.total` der Suchantwort zurück.

`_total`unterstützt die `accurate` none Parameterwerte. `_total=estimate`wird nicht unterstützt. Alle anderen Werte werden als ungültig behandelt. `_total`gilt nicht für die eingeschlossenen Ressourcen (zurückgegebene Ressourcen, deren Suchmodus ist`include`).

Example — Ruft die Gesamtzahl der Patientenressourcen in einem Datenspeicher ab:

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

`_sort`

Durch die Verwendung `_sort` in der Suchabfrage werden die Ergebnisse in einer bestimmten Reihenfolge angeordnet. Die Ergebnisse werden auf der Grundlage der durch Kommas getrennten Liste von Sortierregeln in Prioritätsreihenfolge sortiert. Bei den Sortierregeln sollte es sich um gültige Suchparameter handeln. Alle anderen Werte werden als ungültig behandelt.

In einer einzigen Suchanfrage können Sie bis zu 5 Sortier-Suchparameter verwenden. Sie können optional ein `-` Präfix verwenden, um die absteigende Reihenfolge anzugeben. Der Server sortiert standardmäßig in aufsteigender Reihenfolge.

Die unterstützten Parametertypen für die Sortiersuche sind:`Number, String, Date, Quantity, Token, URI, Reference`. Wenn sich ein Suchparameter auf ein verschachteltes Element bezieht,

wird dieser Suchparameter für die Sortierung nicht unterstützt. Beispiel: Die Suche nach „Name“ des Ressourcentyps Patient bezieht sich auf Patient. Ein Name-Element mit HumanName Datentyp wird als verschachtelt betrachtet. Daher wird die Sortierung von Patientenressourcen nach „Name“ nicht unterstützt.

Example — Rufen Sie Patientenressourcen in einem Datenspeicher ab und sortieren Sie sie in aufsteigender Reihenfolge nach dem Geburtsdatum:

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

`_count`

Der Parameter `_count` ist als Anweisung an den Server definiert, wie viele Ressourcen auf einer einzigen Seite zurückgegeben werden sollen.

Die maximale Seitengröße ist 100. Alle Werte, die größer als 100 sind, sind ungültig. `_count=0`wird nicht unterstützt.

Example — Suchen Sie nach der Patientenressource und legen Sie die Größe der Suchseite auf 25 fest:

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

Chaining and Reverse Chaining(_has)

Verkettung und umgekehrte Verkettung FHIR bieten eine effizientere und kompaktere Methode zum Abrufen miteinander verbundener Daten, wodurch die Notwendigkeit mehrerer separater Abfragen reduziert wird und der Datenabruf für Entwickler und Benutzer komfortabler wird.

Wenn eine Rekursionsebene mehr als 100 Ergebnisse zurückgibt, HealthLake wird 4xx zurückgegeben, um den Datenspeicher vor Überlastung und mehrfachen Paginierungen zu schützen.

Example — Verkettung — Ruft alle Daten ab, die sich auf einen Patienten beziehen DiagnosticReport , wobei der Patientenname Peter ist.

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

Example — Umgekehrte Verkettung — Ruft Patientenressourcen ab, wobei die Patientenressource durch mindestens eine Beobachtung referenziert wird, wobei die Beobachtung den Code 1234 hat und wobei sich die Beobachtung auf die Patientenressource im Patientensuchparameter bezieht.

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

Unterstützte Suchmodifikatoren

Suchmodifikatoren werden mit Feldern verwendet, die auf Zeichenfolgen basieren. Alle verwendeten Suchmodifikatoren verwenden boolesche Logik HealthLake . Sie könnten beispielsweise angeben`:contains`, dass ein größeres Zeichenkettenfeld eine kleine Zeichenfolge enthalten soll, damit es in Ihre Suchergebnisse aufgenommen wird.

Unterstützte Suchmodifikatoren

Suchmodifikator	Typ
: fehlt	Alle Parameter außer `Composite`
:exakt	String
:enthält	String
:nicht	Token
:Text	Token
:Bezeichner	Referenz

Unterstützte Suchvergleicher

Sie können Suchkomparatoren verwenden, um die Art der Übereinstimmung in einer Suche zu steuern. Sie können Komparatoren verwenden, wenn Sie nach Zahlen-, Datums- und Mengenfeldern suchen. In der folgenden Tabelle sind Suchkomparatoren und ihre Definitionen aufgeführt, die von unterstützt werden. HealthLake

Unterstützte Suchvergleicher

Suchkomparator	Beschreibung
eq	Der Wert für den Parameter in der Ressource entspricht dem angegebenen Wert.
Ein	Der Wert für den Parameter in der Ressource entspricht nicht dem angegebenen Wert.
gt	Der Wert für den Parameter in der Ressource ist größer als der angegebene Wert.
lt	Der Wert für den Parameter in der Ressource ist kleiner als der angegebene Wert.
ge	Der Wert für den Parameter in der Ressource ist größer oder gleich dem angegebenen Wert.
le	Der Wert für den Parameter in der Ressource ist kleiner oder gleich dem angegebenen Wert.
sa	Der Wert für den Parameter in der Ressource beginnt nach dem angegebenen Wert.
eb	Der Wert für den Parameter in der Ressource endet vor dem angegebenen Wert.

Suchparameter werden nicht unterstützt von HealthLake

Eine vollständige Liste der unterstützten Suchparameter finden Sie in der FHIRSuchparameter-Registry. HealthLake unterstützt alle Suchparameter mit Ausnahme der in der Tabelle aufgeführten.

Suchparameter werden nicht unterstützt

Zusammenstellung des Pakets	Lage in der Nähe
Bundle-ID	C onsent-source-reference
Bundle-Botschaft	Vertragspatient

Typ des Pakets	Inhalt der Ressource
Zeitstempel des Pakets	Ressourcen-Abfrage

Suche mit POST Beispielen

Sie können einen HealthLake Datenspeicher durchsuchen, indem Sie POST Anfragen stellen. Sie können Abfrageparameter entweder im URI oder in einem Anforderungstext angeben, aber Sie können nicht beide in einer einzigen Anfrage verwenden.

Die Beispiele in diesem Thema folgen dieser bewährten Methode.

> **Note**
>
> Für Anfragen, die personenbezogene Daten (PII) oder geschützte Gesundheitsinformationen (PHI) betreffen, wird empfohlen, `POST` Anfragen zu verwenden. In einer `POST` Anfrage PII oder PHI wird als Teil des Anfragetextes hinzugefügt und bei der Übertragung verschlüsselt.

Wenn POST Sie eine Anfrage mit einem Parameter im Anforderungstext stellen, verwenden Sie `Content-Type: application/x-www-form-urlencoded` ihn als Teil des Headers.

In diesem Thema finden Sie Beispiele dafür, wie Sie POST mithilfe der folgenden Ressourcentypen suchen können.

- Alter: Alter ist kein definierter Ressourcentyp inFHIR. Stattdessen wird das Alter als Teil des Ressourcentyps Patient erfasst. Um anhand eines bestimmten Alters oder einer bestimmten Altersgruppe nach einer Patientengruppe zu suchen, verwenden Sie the section called “Unterstützte Suchvergleicher” a. Weitere Informationen finden Sie unter Ressourcentyp: Patient im FHIRDokumentationsindex.
- Zustand: Dieser Ressourcentyp speichert Details zu klinischen Konzepten wie einer Diagnose, Situationen, einem klinischen Zustand und Problemen, die Anlass zu Besorgnis geben. Weitere Informationen finden Sie im FHIRDokumentationsindex unter Ressourcentyp: Zustand. HealthLake erstellt neue Bedingungen auf der Grundlage von Dokumenten in DocumentReference. Diese Ergänzungen sind standardmäßig ausgeschlossen, wenn eine `POST` Anfrage gestellt wird. Um sie einzubeziehen, müssen Sie bei Ihrer Suche einen gültigen Bezeichner für eine Bedingungsressource angeben.

- DocumentReferenceUm sie einzubeziehen, müssen Sie in Ihrer Suche einen gültigen Bezeichner für eine Bedingungsressource HealthLake angeben. ----SEP----:Dieser Ressourcentyp wird unterstützt von. Dieser Ressourcentyp unterstützt das Verweisen auf Dokumente aller Art. Weitere Informationen finden Sie unter Ressourcentyp: DocumentReference im FHIRDokumentationsindex. HealthLake bietet auch eine integrierte Verarbeitung natürlicher Sprache (NLP) von Dokumenten in der DocumentReference. Weitere Informationen hierzu finden Sie unter Verwenden der automatisierten Ressourcengenerierung auf der Grundlage der Verarbeitung natürlicher Sprache (NLP) des FHIR DocumentReference Ressourcentyps in AWS HealthLake.
- Standort: Dieser Ressourcentyp umfasst sowohl zufällige Standorte (ein Ort, der ohne vorherige Benennung oder Genehmigung für medizinische Zwecke genutzt wird) als auch spezielle, formell festgelegte Standorte. Weitere Informationen finden Sie im FHIRDokumentationsindex unter Ressourcentyp: Standort.
- Beobachtung: Messungen und einfache Aussagen über einen Patienten, ein Gerät oder eine andere Person. HealthLake erstellt neue Beobachtungsressourcen auf der Grundlage von Dokumenten, die in der DocumentReference Ressource gefunden wurden. Weitere Informationen darüber, wie neue Ressourcen HealthLake erstellt werden, finden Sie unterVerwenden der automatisierten Ressourcengenerierung auf der Grundlage der Verarbeitung natürlicher Sprache (NLP) des FHIR DocumentReference Ressourcentyps in AWS HealthLake. Diese Ergänzungen sind standardmäßig ausgeschlossen, wenn eine `POST` Anfrage gestellt wird. Um sie einzubeziehen, müssen Sie bei Ihrer Suche einen gültigen Bezeichner für eine Beobachtungsressource angeben. Weitere Informationen finden Sie unter Ressourcentyp: Beobachtung im FHIRDokumentationsindex.

Jede Registerkarte enthält Beispiele für die Suche nach dem angegebenen Ressourcentyp. Es enthält ein Beispiel dafür, wie die Anfrage im Hauptteil der Anfrage spezifiziert wird.

Age

Gehen Sie wie folgt vor, um eine auf dem `Patient` Ressourcentyp `POST` basierende Suchanfrage zu stellen. Bei dieser Suche wird der `eq` Suchkomparator verwendet, um nach Personen zu suchen, die 1997 geboren wurden.

Sie müssen eine Anfrage URL und einen Anfragetext angeben. Hier ist ein Beispiel für eine AnfrageURL.

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

Um das Jahr 1997 in der Suche anzugeben, würden Sie dem Anfragetext das folgende Element hinzufügen.

```
birthdate=eq1997
```

JSON-Antwort

Bei Erfolg erhalten Sie einen `200` HTTP Antwortcode und eine ähnliche JSON Antwort.

Condition

Verwenden Sie Folgendes, um eine `POST` Anfrage für den `Condition` Ressourcentyp zu stellen. Bei dieser Suche werden Stellen in Ihrem HealthLake Datenspeicher gefunden, die den medizinischen Code enthalten`72892002`.

Sie müssen eine Anfrage URL und einen Anfragetext angeben. Hier ist ein Beispiel für eine AnfrageURL.

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

Um den medizinischen Code anzugeben, nach dem Sie suchen möchten, fügen Sie dieses JSON Element dem Anfragetext hinzu.

```
code=72892002
```

JSON-Antwort

Bei Erfolg erhalten Sie einen `200` HTTP Antwortcode. Die folgende JSON Antwort wurde aus Gründen der Übersichtlichkeit gekürzt.

```
{
 "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

Um die Ergebnisse der HealthLake integrierten Verarbeitung natürlicher Sprache (NLP) zu sehen, wenn POST Sie eine Anfrage für den `DocumentReference` Ressourcentyp stellen, formatieren Sie eine Anfrage wie folgt.

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

Informationen zur Angabe des DocumentReference Elements, auf das Sie verweisen möchten, finden Sie unterSuchparameter. Sie geben diese im Anfragetext als anJSON.

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

Diese Abfragezeichenfolge verwendet mehrere Suchparameter für die Suche nach den API Vorgängen von Amazon Comprehend Medical, die zur Generierung der integrierten medizinischen NLP Ergebnisse verwendet werden.

Location

Verwenden Sie Folgendes, um eine `POST` Anfrage für den `Location` Ressourcentyp zu stellen. Diese Suche findet Orte in Ihrem HealthLake Datenspeicher, die den Stadtnamen Boston als Teil der Adresse enthalten.

Sie müssen eine Anfrage URL und einen Anfragetext angeben. Hier ist ein Beispiel für eine AnfrageURL.

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

Um Boston in der Suche anzugeben, fügen Sie dem Anfragetext das folgende Element hinzu:

```
address=Boston
```

JSON-Antwort

Bei Erfolg erhalten Sie einen `200` HTTP Antwortcode. Die JSON Antwort wurde aus Gründen der Übersichtlichkeit gekürzt.

```
{
 "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

Verwenden Sie Folgendes, um eine auf dem `Observation` Ressourcentyp `POST` basierende Suchanfrage zu stellen. Diese Suche verwendet den `value-concept` Suchparameter, um nach medizinischem Code zu suchen,`266919005`. Dieser Status zeigt an`Never smoker`.

Sie müssen eine Anfrage URL und einen Anfragetext angeben. Hier ist ein Beispiel für eine AnfrageURL.

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

Um den Status anzugeben`Never smoker`, legen Sie ihn `value-concept=266919005` im Hauptteil von festJSON.

```
value-concept=266919005
```

JSON-Antwort

Bei Erfolg erhalten Sie einen 200 HTTP Antwortcode. Die folgende JSON Antwort wurde aus Gründen der Übersichtlichkeit gekürzt.


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

Suche mit GET Beispielen

Sie können einen HealthLake Datenspeicher durchsuchen, indem Sie GET Anfragen stellen. HealthLake unterstützt nur die Bereitstellung von Abfrageparametern als Teil von und nicht als Teil eines Anforderungstexts. URI

> **Note**
>
> Für Anfragen, die personenbezogene Daten (PII) oder geschützte Gesundheitsinformationen (PHI) betreffen, wird empfohlen, POST Anfragen zu verwenden. In einer POST Anfrage PII oder PHI wird als Teil des Anfragetextes hinzugefügt und bei der Übertragung verschlüsselt.

Das Thema enthält Beispiele für die Suche GET mithilfe unterstützter Ressourcentypen in HealthLake.

- Alter: Alter ist kein definierter Ressourcentyp inFHIR. Stattdessen wird das Alter als Teil des Ressourcentyps des Patienten erfasst. Um anhand eines bestimmten Alters oder einer bestimmten Altersgruppe nach einer Patientengruppe zu suchen, müssen Sie a verwendenthe section called “Unterstützte Suchvergleicher”. Weitere Informationen finden Sie unter Ressourcentyp: Patient im FHIRDokumentationsindex.

- Zustand: Dieser Ressourcentyp speichert Details zu klinischen Konzepten wie einer Diagnose, Situationen, einem klinischen Zustand und Problemen, die Anlass zu Besorgnis geben. Weitere Informationen finden Sie im FHIRDokumentationsindex unter Ressourcentyp: Zustand. HealthLake erstellt neue Bedingungen auf der Grundlage von Dokumenten, die in der gefunden wurden DocumentReference. Diese Ergänzungen sind standardmäßig ausgeschlossen, wenn eine `POST` Anfrage gestellt wird. Um sie einzubeziehen, müssen Sie bei Ihrer Suche einen gültigen Bezeichner für eine Bedingungsressource angeben.
- DocumentReferenceUm sie einzubeziehen, müssen Sie in Ihrer Suche einen gültigen Bezeichner für eine Bedingungsressource HealthLake angeben. ----SEP----:Dieser Ressourcentyp wird unterstützt von. Dieser Ressourcentyp unterstützt das Verweisen auf Dokumente aller Art. Weitere Informationen finden Sie unter Ressourcentyp: DocumentReference im FHIRDokumentationsindex. HealthLake bietet auch eine integrierte Verarbeitung natürlicher Sprache (NLP) von Dokumenten in der DocumentReference. Weitere Informationen hierzu finden Sie unter Verwenden der automatisierten Ressourcengenerierung auf der Grundlage der Verarbeitung natürlicher Sprache (NLP) des FHIR DocumentReference Ressourcentyps in AWS HealthLake.
- Standort: Dieser Ressourcentyp umfasst sowohl zufällige Standorte (ein Ort, der ohne vorherige Benennung oder Genehmigung für medizinische Zwecke genutzt wird) als auch spezielle, formell festgelegte Standorte. Weitere Informationen finden Sie unter Ressourcentyp: Standort im FHIRDokumentationsindex.
- Beobachtung: Messungen und einfache Aussagen über einen Patienten, ein Gerät oder eine andere Person. HealthLake erstellt neue Beobachtungsressourcen auf der Grundlage von Dokumenten, die in der DocumentReference Ressource gefunden wurden. Weitere Informationen zum HealthLake Erstellen neuer Ressourcen finden Sie unterVerwenden der automatisierten Ressourcengenerierung auf der Grundlage der Verarbeitung natürlicher Sprache (NLP) des FHIR DocumentReference Ressourcentyps in AWS HealthLake. Diese Ergänzungen sind standardmäßig ausgeschlossen, wenn eine `POST` Anfrage gestellt wird. Um sie einzubeziehen, müssen Sie bei Ihrer Suche einen gültigen Bezeichner für eine Beobachtungsressource angeben. Weitere Informationen finden Sie unter Ressourcentyp: Beobachtung im FHIRDokumentationsindex.

Jede Registerkarte zeigt ein Beispiel für die Suche nach dem angegebenen Ressourcentyp. Es enthält ein Beispiel dafür, wie die Anfrage in der und die URI zugehörige JSON Antwort spezifiziert werden.

Age

Gehen Sie wie folgt vor, um eine auf dem `Patient` Ressourcentyp `GET` basierende Suchanfrage zu stellen. Bei dieser Suche wird der `eq` Suchkomparator verwendet, um nach Personen zu suchen, die 1997 geboren wurden.

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

JSON-Antwort

Bei Erfolg erhalten Sie einen `200` HTTP Antwortcode.

Condition

Verwenden Sie Folgendes, um eine `GET` Anfrage für den `Condition` Ressourcentyp zu stellen. Bei dieser Suche werden Stellen in Ihrem HealthLake Datenspeicher gefunden, die den medizinischen Code enthalten`72892002`.

Sie müssen eine Anfrage URL und einen Anfragetext angeben. Hier ist ein Beispiel für eine AnfrageURL.

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

JSON-Antwort

Bei Erfolg erhalten Sie einen `200` HTTP Antwortcode. Die folgende JSON Antwort wurde aus Gründen der Übersichtlichkeit gekürzt.

```
{
 "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

Dieses Beispiel zeigt, wie eine Suchanfrage für den `DocumentReference` Ressourcentyp für Patienten erstellt wird, bei denen Streptokokken diagnostiziert wurden und denen auch Amoxicillin verschrieben wurde.

```
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
```

Bei Erfolg erhalten Sie die folgende Antwort. 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

Verwenden Sie Folgendes, um eine GET Anfrage für den Location Ressourcentyp zu stellen. Diese Suche findet Orte in Ihrem HealthLake Datenspeicher, die den Stadtnamen Boston als Teil der Adresse enthalten.

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

JSON-Antwort

Bei Erfolg erhalten Sie einen 200 HTTP Antwortcode. Die JSON Antwort wurde aus Gründen der Übersichtlichkeit gekürzt.

```
{
    "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

Gehen Sie wie folgt vor, um eine auf dem `Observation` Ressourcentyp `GET` basierende Suchanfrage zu stellen. Diese Suche verwendet den `value-concept` Suchparameter, um nach medizinischem Code zu suchen,`266919005`. Dieser Status zeigt an`Never smoker`.

Sie müssen eine Anfrage URL und eine Abfragezeichenfolge angeben. Hier ist ein Beispiel für eine AnfrageURL.

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

Um den Status anzugeben`Never smoker`, legen Sie ihn `value-concept=266919005` als Abfragezeichenfolge fest.

JSON-Antwort

Bei Erfolg erhalten Sie einen `200` HTTP Antwortcode. Die folgende JSON Antwort wurde aus Gründen der Übersichtlichkeit gekürzt.

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

FHIRRessourcenverlauf lesen

Die FHIR `history` Interaktion ruft den Verlauf einer bestimmten FHIR Ressource in einem HealthLake Datenspeicher ab. Mithilfe dieser Interaktion können Sie feststellen, wie sich der Inhalt einer FHIR Ressource im Laufe der Zeit verändert hat. In Abstimmung mit Audit-Logs ist es auch nützlich, den Status einer Ressource vor und nach der Änderung zu überprüfen.

> **Note**
>
> FHIRDiese Ressource `history` ist standardmäßig für alle HealthLake Datenspeicher aktiviert, die nach dem 25.10.2024 erstellt wurden. Wenn Ihr Datenspeicher vor diesem Datum erstellt wurde, können Sie ein Supportticket einreichen, um die FHIR `history` Interaktion zu aktivieren. Erstellen Sie einen Fall mit AWS Support Center Console. Um Ihren Fall zu erstellen, melden Sie sich bei Ihrem an AWS-Konto und wählen Sie Fall erstellen.

Die `history` Interaktion wird mit dem `HTTP GET` Befehl ausgeführt. Die FHIR Interaktionen `create`, `update`, und `delete` führen zu einer historischen Version der Ressource, die gespeichert werden soll. HealthLake unterstützt die folgenden Suchparameter für die FHIR `history` Interaktion.

HealthLake unterstützte Suchparameter für die FHIR **`history`** Interaktion

Suchparameter	Beschreibung
`_count : integer`	Die maximale Anzahl von Suchergebnissen auf einer Seite. Der Server gibt die angeforde rte Anzahl oder die maximale Anzahl von

Suchparameter	Beschreibung
	Suchergebnissen zurück, die standardm äßig für den Datenspeicher zulässig sind, je nachdem, welcher Wert niedriger ist.
`_since : instant`	Schließt nur Ressourcenversionen ein, die zu oder nach dem angegebenen Zeitpunkt erstellt wurden.
`_at : date(Time)`	Schließt nur Ressourcenversionen ein, die zu einem bestimmten Zeitpunkt während des im Datums- und Uhrzeitwert angegebenen Zeitraums aktuell waren. Weitere Informationen finden Sie `date`in der HL7FHIRRESTfulAPID okumentation.

Das folgende Beispiel gibt 100 historische Suchergebnisse pro Seite für eine FHIR `Patient` Ressource in zurück HealthLake. Scrollen Sie über die Schaltfläche Kopieren, um den gesamten URL Pfad anzuzeigen. URLDas hat die Form:

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

Der zurückgegebene Inhalt einer Verlaufsinteraktion ist in einer FHIR Ressource enthalten `Bundle`, deren Typ auf gesetzt ist`history`. Sie enthält den angegebenen Versionsverlauf, sortiert nach den ältesten Versionen zuletzt, und enthält gelöschte Ressourcen. Weitere Informationen zur `history` Interaktion finden Sie `history`in der HL7 FHIR RESTful API Dokumentation.

> **Note**
>
> Sie können sich `history` für bestimmte FHIR Ressourcentypen abmelden. Um sich abzumelden, erstellen Sie einen Fall mit AWS Support Center Console. Um Ihren Fall zu erstellen, melden Sie sich bei Ihrem an AWS-Konto und wählen Sie Fall erstellen.

Den versionsspezifischen Ressourcenverlauf FHIR lesen

Die FHIR `vread` Interaktion führt einen versionsspezifischen Lesevorgang einer Ressource in einem Datenspeicher durch. HealthLake Mithilfe dieser Interaktion können Sie den Inhalt einer FHIR Ressource so anzeigen, wie er zu einem bestimmten Zeitpunkt in der Vergangenheit war.

HealthLake erklärt, dass es die Versionierung `CapabilityStatement.rest.resource.versioning`für jede unterstützte Ressource unterstützt. Alle HealthLake Datenspeicher enthalten `Resource.meta.versionId` (`vid`) für alle Ressourcen.

Wenn FHIR `history` Interaktion aktiviert ist (standardmäßig für Datenspeicher, die nach dem 25.10.2024 erstellt wurden, oder auf Anfrage für ältere Datenspeicher), beinhaltet die `Bundle` Antwort den `vid` als Teil von. `location` Im folgenden Beispiel `vid` wird der als Zahl angezeigt. 1 Das vollständige Beispiel finden Sie unter Beispiel bundle/bundle-response (). JSON

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

Die `vread` Interaktion wird mit dem Befehl ausgeführt. `HTTP GET` Die folgende `vread` Interaktion gibt eine einzelne Instanz mit dem für die Ressource angegebenen Inhalt für die Version der FHIR `Patient` Ressourcenmetadaten zurück, die von der angegeben wurde`vid`. Scrollen Sie über die Schaltfläche Kopieren, um den gesamten URL Pfad im folgenden Beispiel anzuzeigen. URLDas hat die Form:

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

> **Note**
>
> Wenn Sie `vread` beim Lesen einer FHIR Ressource die `history` Interaktion ohne verwenden, wird HealthLake immer die neueste Version der Metadaten der Ressource zurückgegeben.

Weitere Informationen zur `vread` Interaktion finden Sie `vread`in der HL7FHIRAPIRestful-Dokumentation.

Patientendaten mit der Operation Patient $everything abrufen FHIR REST API

Die Operation Patient $everything wird verwendet, um eine FHIR Patientenressource zusammen mit allen anderen Ressourcen, die sich auf diesen Patienten beziehen, abzufragen. Dieser Vorgang kann verwendet werden, um einem Patienten Zugriff auf seine gesamte Patientenakte zu gewähren oder es einem Anbieter zu ermöglichen, einen Massendatendownload für einen Patienten durchzuführen. HealthLake unterstützt $everything für eine bestimmte Patienten-ID.

> **Note**
>
> Die Operation Patient $everything wird derzeit für Datenspeicher unterstützt, die nach dem 27. Februar 2024 erstellt wurden.

Ruft alle Ressourcen ab, die sich auf einen Patienten beziehen

Patient $everything ist eine REST API Operation, die wie in den folgenden Beispielen gezeigt aufgerufen werden kann.

GET Request

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

> **Note**
>
> Die antwortenden Ressourcen sind nach Ressourcentyp und Ressourcen-ID sortiert.
> Die Antwort ist immer mit Bundle.total gefüllt.

$everything Parameter für den Patienten

HealthLake unterstützt die folgenden Abfrageparameter

Parameter	Details
start	Ruft alle Patientendaten ab einem bestimmten Startdatum ab.
end	Ruft alle Patientendaten vor einem bestimmten Enddatum ab.
since	Lassen Sie alle Patientendaten nach einem bestimmten Datum aktualisieren.
_Typ	Rufen Sie Patientendaten für bestimmte Ressourcentypen ab.
_Anzahl	Rufen Sie Patientendaten ab und geben Sie die Seitengröße an.

Example - Ruft alle Patientendaten ab einem bestimmten Startdatum ab

Patient $everything kann den `start` Filter verwenden, um nur Daten nach einem bestimmten Datum abzufragen.

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

Example - Ruft alle Patientendaten vor einem bestimmten Enddatum ab

Patient $everything kann den `end` Filter verwenden, um nur Daten vor einem bestimmten Datum abzufragen.

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

Example - Alle Patientendaten werden nach einem bestimmten Datum aktualisiert

Patient $everything kann den `since` Filter verwenden, um nur Daten abzufragen, die nach einem bestimmten Datum aktualisiert wurden.

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

Example - Rufen Sie Patientendaten für bestimmte Ressourcentypen ab

Patient $everything kann den `_type` Filter verwenden, um bestimmte Ressourcentypen anzugeben, die in die Antwort aufgenommen werden sollen. In einer durch Kommas getrennten Liste können mehrere Ressourcentypen angegeben werden.

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

Example - Rufen Sie Patientendaten ab und geben Sie die Seitengröße an

Patient $everything kann das verwenden`_count`, um die Seitengröße einzustellen.

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

Patient $everything **`start`** und Attribute **`end`**

HealthLake unterstützt die folgenden Ressourcenattribute für die Start- und Endabfrageparameter.

Ressource	"Resource"-Element
Account	Konto. servicePeriod. Start
AdverseEvent	AdverseEvent. Datum
AllergyIntolerance	AllergyIntolerance.recordedDate
Ernennung	Termin. Start
AppointmentResponse	AppointmentResponse. starten
AuditEvent	AuditEvent.Zeitraum.Beginn

Ressource	"Resource"-Element
Basic	Basic. Erstellt
BodyStructure	NEIN_ DATE
CarePlan	CarePlan.Zeitraum.Beginn
CareTeam	CareTeam.Zeitraum.Beginn
ChargeItem	ChargeItem. occurrenceDateTime, ChargeItem. occurrencePeriod. starten, ChargeItem. occurrenceTiming. Ereignis
Antrag	Anspruch erheben. billablePeriod. starten
ClaimResponse	ClaimResponse. erstellt
ClinicalImpression	ClinicalImpression. Datum
Kommunikation	Mitteilung. Gesendet
CommunicationRequest	CommunicationRequest. occurrenceDateTime, CommunicationRequest. occurrencePeriod. starten
Komposition	Zusammensetzung.Datum
Bedingung	Zustand. recordedDate
Zustimmung	Zustimmung. dateTime
Deckung	Deckung. Zeitraum. Start

Ressource	"Resource"-Element
CoverageEligibilityRequest	CoverageEligibilityRequest. erstellt
CoverageEligibilityResponse	CoverageEligibilityResponse. erstellt
DetectedIssue	DetectedIssue. identifiziert
DeviceRequest	DeviceRequest.authoredOn
DeviceUseStatement	DeviceUseStatement.recordedOn
DiagnosticReport	DiagnosticReport. wirksam
DocumentManifest	DocumentManifest. geschaffen
DocumentReference	DocumentReference.kontext.period.start
Begegnung	Begegnung. Zeit. Start
EnrollmentRequest	EnrollmentRequest. erstellt
EpisodeOfCare	EpisodeOfCare.Zeitraum.Start
ExplanationOfBenefit	ExplanationOfBenefit. billablePeriod. starten

Ressource	"Resource"-Element
FamilyMemberHistory	NEIN_ DATE
Flag	Flaggen.Zeitraum.Start
Ziel	Ziel. statusDate
Gruppe	NEIN_ DATE
ImagingStudy	ImagingStudy. gestartet
Immunisierung	Immunisierung. Aufgenommen
ImmunizationEvaluation	ImmunizationEvaluation. Datum
ImmunizationRecommendation	ImmunizationRecommendation.datum
Rechnung	Rechnung.Datum
Auflisten	Liste.Datum
MeasureReport	MeasureReport.Zeitraum.Beginn
Medien	Medien/ausgegeben
MedicationAdministration	MedicationAdministration. wirksam
MedicationDispense	MedicationDispense.whenPrepared

Ressource	"Resource"-Element
MedicationRequest	MedicationRequest.authoredOn
MedicationStatement	MedicationStatement.dateAsserted
MolecularSequence	NEIN_ DATE
NutritionOrder	NutritionOrder.dateTime
Beobachtung	Beobachtung. Wirksam
Patientin	NEIN_ DATE
Person	NEIN_ DATE
Verfahren	Verfahren. Durchgeführt
Herkunft	Provenienz. occurredPeriod.start, Provenienz. occurredDateTime
QuestionnaireResponse	QuestionnaireResponse. verfasst
RelatedPerson	NEIN_ DATE
RequestGroup	RequestGroup.authoredOn
ResearchSubject	ResearchSubject. Zeitraum
RiskAssessment	RiskAssessment. occurrenceDateTime, RiskAssessment. occurrencePeriod. starten

Ressource	"Resource"-Element
Plan	Zeitplan. planningHorizon
ServiceRequest	ServiceRequest.authoredOn
Exemplar	Exemplar. receivedTime
SupplyDelivery	SupplyDelivery. occurrenceDateTime, SupplyDelivery. occurrencePeriod. starten, SupplyDelivery. occurrenceTiming. Ereignis
SupplyRequest	SupplyRequest.authoredOn
VisionPrescription	VisionPrescription.dateWritten

Exportieren von Daten aus Ihrem HealthLake Datenspeicher mit $export

Um eine Exportanforderung zu stellen, indem FHIR REST API Sie die Angabe `$export` als Teil der `POST` Anfrage verwenden und Anforderungsparameter in den Hauptteil Ihrer Anfrage aufnehmen. Gemäß der FHIR Spezifikation muss der FHIR Server `GET` Anfragen unterstützen und kann `POST` Anfragen unterstützen. Um zusätzliche Parameter zu unterstützen, ist ein Body erforderlich, um den Export zu starten. Daher werden POST Anfragen HealthLake unterstützt.

> **Important**
>
> HealthLake Datenspeicher, die vor dem 1. Juni 2023 erstellt wurden, unterstützen nur FHIR REST API basierte Exportauftragsanforderungen für systemweite Exporte.
> HealthLake Datenspeicher, die vor dem 1. Juni 2023 erstellt wurden, unterstützen nicht das Abrufen des Status eines Exports mithilfe einer `GET` Anfrage am Endpunkt eines Datenspeichers.

Alle Exportanfragen, die Sie mit dem stellen, FHIR REST API werden im `ndjson` Format zurückgegeben und in einen Amazon S3 S3-Bucket exportiert. Jedes S3-Objekt wird nur einen einzigen FHIR Ressourcentyp enthalten.

Sie können jeweils eine einzige Exportanforderung für jedes AWS Konto stellen. Weitere Informationen zu den damit verbundenen Service Quotas finden Sie unterAWS HealthLake Endpunkte und Kontingente. HealthLake

Weitere Informationen darüber, wie Sie mithilfe von eine Exportanfrage stellen FHIR RESTAPI, finden Sie unterExportieren von Daten aus Ihrem HealthLake Datenspeicher mit FHIR REST API Operationen.

AWS HealthLake Datenspeicher SQL in Amazon Athena abfragen

Wenn Sie einen HealthLake Datenspeicher erstellen, wird die stark verschachtelte FHIR Datenstruktur in Amazon Athena aufgenommen und automatisch in Iceberg-Tabellen umgewandelt, mit denen abgefragt werden kann. SQL Die Gewährung des Zugriffs auf diese neue Ressource wird mithilfe von AWS Lake Formation verwaltet. Jeder FHIR Ressourcentyp wird in Athena als einzelne Tabelle dargestellt.

Important

Für Datenspeicher, die vor dem 14. November 2022 erstellt wurden, müssen Sie Ihren vorhandenen Datenspeicher auf einen neuen migrieren, um ihn abfragen zu könnenSQL. Weitere Informationen dazu finden Sie unter Migration eines vorhandenen Datenspeichers zur Verwendung von Amazon Athena.

Note

Nach dem 20. Februar 2023 verwenden HealthLake Datenspeicher standardmäßig keine integrierte Verarbeitung natürlicher Sprache (NLP). Wenn Sie daran interessiert sind, diese Funktion in Ihrem Datenspeicher zu aktivieren, finden Sie Wie aktiviere ich die HealthLake integrierte Funktion zur Verarbeitung natürlicher Sprache? weitere Informationen im Kapitel Fehlerbehebung.

Um einen HealthLake Datenspeicher zu erstellen, müssen Sie Ihrem IAM Benutzer oder Ihrer HealthLake Administratorrolle zusätzliche IAM Richtlinien und eine Servicerolle hinzufügen. Weitere Informationen zum Einrichten von Berechtigungen finden Sie unterBerechtigungen einrichten, um mit der Verwendung zu beginnen AWS HealthLake.

HealthLake Datenspeicher werden als Iceberg-Tabellen in Athena aufgenommen. Weitere Informationen zur Funktionsweise von Eisberg-Tabellen in Athena finden Sie unter Verwenden von Eisberg-Tabellen im Athena-Benutzerhandbuch.

HealthLake unterstützt `READ` den Betrieb Ihrer HealthLake Datenspeicher Datenspeicher in Athena. Weitere Informationen zu den Vorgängen Create, Read, Update und Delete (CRUD) mithilfe der FHIR

REST API Operationen finden FHIRRESTAPIInteraktionen mit einem HealthLake Datenspeicher verwenden Sie unter Weitere Informationen darüber, wie sich CRUD Operationen auf Ihre Daten in Athena auswirken.

In den Themen dieses Kapitels wird beschrieben, wie Sie Ihren HealthLake Datenspeicher mit Athena verbinden, wie Sie ihn mithilfe von SQL Athena abfragen und wie Sie Ergebnisse für weitere Analysen mit anderen AWS Diensten verbinden.

Inhalt

Ihren Datenspeicher mit Amazon Athena verbinden

> **Important**
>
> Nach dem 14. November 2022 haben sich die IAM Zugangsvoraussetzungen HealthLake geändert. Um in Athena sowohl Datenspeicher zu erstellen als auch Zugriff darauf zu gewähren, müssen Sie die `AWSLakeFormationDataAdmin` verwaltete Richtlinie zu Ihrem IAM Benutzer, Ihrer Gruppe oder Rolle hinzugefügt haben. Sie können die `AWSLakeFormationDataAdmin` Richtlinie verwenden, um Data Lake-Administratoren zu erstellen und Zugriff auf Datenspeicher in Athena zu gewähren.

In diesem Thema werden die notwendigen Schritte beschrieben, um einen Athena-Benutzer, eine Gruppe oder eine Athena-Rolle zu erstellen und ihnen Zugriff auf FHIR Ressourcen in einem HealthLake Datenspeicher zu gewähren.

- Einem Benutzer, einer Gruppe oder einer Rolle Zugriff auf einen HealthLake Datenspeicher gewähren (AWS Lake Formation Console)
- Einrichtung eines Athena-Kontos

Einem Benutzer, einer Gruppe oder einer Rolle Zugriff auf einen HealthLake Datenspeicher gewähren (AWS Lake Formation Console)

> Persona: Administrator HealthLake
>
> Die HealthLake Administrator-Persona ist ein Data Lake-Administrator in AWS Lake Formation. Sie gewähren Zugriff auf HealthLake Datenspeicher in Lake Formation.

Für jeden erstellten Datenspeicher sind zwei Einträge in der AWS Lake Formation Formation-Konsole sichtbar. Ein Eintrag ist ein Ressourcenlink. Namen von Ressourcenlinks werden immer kursiv angezeigt. Jeder Ressourcenlink wird mit dem Namen und dem Besitzer der verknüpften gemeinsam genutzten Ressource angezeigt. Für alle HealthLake Datenspeicher ist der Besitzer der gemeinsam genutzten Ressource das HealthLake Dienstkonto. Der andere Eintrag ist der HealthLake Datenspeicher im HealthLake Dienstkonto. Die Schritte in diesem Verfahren verwenden den Datenspeicher, der die Ressourcenverknüpfung darstellt.

Weitere Informationen zu Ressourcenlinks finden Sie unter So funktionieren Ressourcenlinks in Lake Formation im AWS Lake Formation Developer Guide.

Damit ein Benutzer, eine Gruppe oder eine Rolle Daten in Athena abfragen kann, müssen Sie die Describe-Berechtigung für die Ressourcendatenbank erteilen. Anschließend müssen Sie den Tabellen die Optionen Select und Describe gewähren.

STEP1: Um DESCRIBEBerechtigungen für eine HealthLake Data Store-Resource Link-Datenbank zu gewähren

1. Öffnen Sie die AWS Lake Formation Formation-Konsole: https://console.aws.amazon.com/lakeformation/
2. Wählen Sie in der primären Navigationsleiste Datenbanken aus.
3. Wählen Sie auf der Seite Datenbanken das Optionsfeld neben dem Namen des Datenspeichers, der kursiv gedruckt ist.
4. Wählen Sie Aktionen (▾).
5. Wählen Sie Gewähren.
6. Wählen Sie auf der Seite Datenberechtigungen gewähren unter Principals die Option IAMBenutzer oder Rollen aus.

7. Verwenden Sie unter IAMBenutzer oder Rollen den Abwärtspfeil (▼) oder suchen Sie nach dem IAM Benutzer, der Rolle oder der Gruppe, zu dem Sie in Athena Abfragen stellen möchten.
8. Wählen Sie unter LF-Tags oder Katalogressourcen die Option Benannte Datenkatalogressourcen aus.
9. Wählen Sie unter Datenbanken mit dem Abwärtspfeil (▼) die HealthLake Datenspeicher-Datenbank aus, auf die Sie gemeinsam zugreifen möchten.
10. Wählen Sie auf der Karte Berechtigungen für Ressourcenlinks unter Berechtigungen für Ressourcenverknüpfungen die Option Beschreiben aus.

Wenn die Erteilung erfolgreich war, wird das Banner „Genehmigung erteilt“ angezeigt. Um die soeben erteilte Berechtigung einzusehen, wählen Sie Data Lake-Berechtigungen aus. Suchen Sie den Benutzer, die Gruppe und die Rolle in der Tabelle. In der Spalte Berechtigungen wird Describe aufgeführt.

Jetzt müssen Sie Grant on target verwenden, um Select und Describe für alle Tabellen in der Datenbank zu gewähren.

STEP2: Gewähren Sie Zugriff auf alle Tabellen in einem HealthLake Datenspeicher-Ressourcenlink

1. Öffnen Sie die AWS Lake Formation Formation-Konsole: https://console.aws.amazon.com/lakeformation/
2. Wählen Sie in der primären Navigationsleiste Datenbanken aus.
3. Wählen Sie auf der Seite Datenbanken das Optionsfeld neben dem Namen des Datenspeichers, der kursiv gedruckt ist.
4. Wählen Sie Aktionen (▼).
5. Wählen Sie Grant on target aus.
6. Wählen Sie auf der Seite Datenberechtigungen gewähren unter Principals die Option IAMBenutzer oder Rollen aus.
7. Verwenden Sie unter IAMBenutzer oder Rollen den Abwärtspfeil (▼) oder suchen Sie nach dem IAM Benutzer, der Gruppe oder der Rolle, zu der Sie in Athena Abfragen stellen möchten.
8. Wählen Sie auf der Karte LF-Tags oder Katalogressourcen die Option Benannte Datenkatalogressourcen aus.
9. Wählen Sie unter Datenbanken mit dem Abwärtspfeil (▼) die HealthLake Datenspeicher-Datenbank aus, auf die Sie Zugriff gewähren möchten.

10. Wählen Sie unter Tabellen die Option Alle Tabellen aus, um alle Tabellen für einen HealthLake Benutzer freizugeben.
11. Wählen Sie auf der Karte Tabellenberechtigungen unter Tabellenberechtigungen die Option Describe and Select aus.
12. Wählen Sie Gewähren.

Nachdem Sie „Gewähren“ ausgewählt haben, wird das Erfolgsbanner „Berechtigungen gewähren“ angezeigt. Der angegebene Benutzer kann jetzt Abfragen an einem HealthLake Datenspeicher in Athena stellen.

Erste Schritte mit Athena

> **HealthLake Nutzer**
>
> Der HealthLake Benutzer verwendet die Athena-Konsole oder AWS SDKs fragt einen HealthLake Datenspeicher ab AWS CLI, der ihm vom HealthLake Administrator zur Verfügung gestellt wurde.

Um einen Datenspeicher mit Athena abzufragen, müssen Sie die folgenden drei Dinge tun.

- Gewähren Sie dem IAM Benutzer oder der Rolle über Lake Formation Zugriff auf den HealthLake Datenspeicher. Weitere Informationen hierzu finden Sie unter Einem Benutzer, einer Gruppe oder einer Rolle Zugriff auf einen HealthLake Datenspeicher gewähren (AWS Lake Formation Console).
- Erstellen Sie eine Arbeitsgruppe für Ihren HealthLake Datenspeicher.
- Benennen Sie einen Amazon S3 S3-Bucket zum Speichern Ihrer Abfrageergebnisse.

Um mit Athena zu beginnen, fügen Sie Ihrem Benutzer, Ihrer Gruppe oder Rolle die FullAccess AWS verwalteten Richtlinien AmazonAthenaFullAccessund AmazonS3 hinzu. Die Verwendung einer AWS verwalteten Richtlinie ist eine hervorragende Möglichkeit, um mit der Nutzung eines neuen Dienstes zu beginnen. Beachten Sie, dass AWS-verwaltete Richtlinien möglicherweise keine Berechtigungen mit den geringsten Berechtigungen für Ihre spezifischen Anwendungsfälle gewähren, da sie für alle AWS-Kunden verfügbar sind. Wenn Sie Berechtigungen mit IAM Richtlinien festlegen, gewähren Sie nur die Berechtigungen, die für die Ausführung einer Aufgabe erforderlich sind. Weitere Informationen über die Verwendung der geringsten Rechte IAM und deren Anwendung finden Sie unter Anwenden von Berechtigungen mit den geringsten Rechten im Benutzerhandbuch. IAM

> **Important**
>
> Um einen HealthLake Datenspeicher in Athena abzufragen, müssen Sie die Athena-Engine Version 3 verwenden.

Arbeitsgruppen sind Ressourcen, und daher können Sie mithilfe IAM von Richtlinien den Zugriff auf bestimmte Arbeitsgruppen steuern. Weitere Informationen finden Sie im Athena-Benutzerhandbuch unter Verwendung von Arbeitsgruppen zur Kontrolle des Abfragezugriffs und der Kosten.

Weitere Informationen zur Einrichtung von Arbeitsgruppen finden Sie https://docs.aws.amazon.com/athena/latest/ug/workgroups-procedure.html im Athena-Benutzerhandbuch.

> **Note**
>
> Die Region, in der sich Ihr Amazon S3 S3-Bucket befindet, und die Athena-Konsole müssen übereinstimmen.

Bevor Sie eine Abfrage ausführen können, muss in Amazon S3 ein Speicherort für das Abfrageergebnis angegeben werden, oder Sie müssen eine Arbeitsgruppe verwenden, für die ein Bucket angegeben wurde und deren Konfiguration die Client-Einstellungen überschreibt. Ausgabedateien werden automatisch für jede Abfrage gespeichert, die ausgeführt wird.

Weitere Informationen zur Angabe von Speicherorten für Abfrageergebnisse in der Athena-Konsole finden Sie unter Angeben eines Speicherorts für Abfrageergebnisse mithilfe der Athena-Konsole im Amazon Athena Athena-Benutzerhandbuch.

Beispiele für die Abfrage Ihres HealthLake Datenspeichers in Athena finden Sie unterFragen Sie Ihren HealthLake Datenspeicher ab mit SQL.

Fragen Sie Ihren HealthLake Datenspeicher ab mit SQL

> **Note**
>
> Nach dem 20. Februar 2023 verwenden HealthLake Datenspeicher standardmäßig keine integrierte Verarbeitung natürlicher Sprache (NLP). Wenn Sie daran interessiert sind, diese Funktion in Ihrem Datenspeicher zu aktivieren, finden Sie Wie aktiviere ich die HealthLake

integrierte Funktion zur Verarbeitung natürlicher Sprache? weitere Informationen im Kapitel Fehlerbehebung.
Alle Beispiele in diesem Thema verwenden fiktionalisierte Daten, die mit Synthea erstellt wurden. Weitere Informationen zum Erstellen eines mit Synthea-Daten vorinstallierten Datenspeichers finden Sie unter. Erstellen eines Datenspeichers in AWS HealthLake

Wenn Sie Ihren HealthLake Datenspeicher in Athena importieren, wird jeder Ressourcentyp aus Ihrem HealthLake Datenspeicher in eine Tabelle konvertiert. Diese Tabellen können einzeln oder als Gruppe mithilfe SQL von basierten Abfragen abgefragt werden. Aufgrund der Struktur der Datenspeicher werden Ihre Daten als mehrere verschiedene Datentypen in Athena importiert. Weitere Informationen zum Erstellen von SQL Abfragen, die auf diese Datentypen zugreifen können, finden Sie unter Abfragen von Arrays mit komplexen Typen und verschachtelten Strukturen im Amazon Athena Athena-Benutzerhandbuch.

Für jedes Element in einem Ressourcentyp definiert die FHIR Spezifikation eine Kardinalität. Die Kardinalität eines Elements definiert die Unter- und Obergrenzen, wie oft dieses Element vorkommen kann. Bei der Erstellung einer SQL Abfrage müssen Sie dies berücksichtigen. Schauen wir uns zum Beispiel einige Elemente im Ressourcentyp: Patient an.

- Element: Name Die FHIR Spezifikation legt die Kardinalität als fest. `0..*`

  Das Element wird als Array erfasst.

```
[{
 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
```

```
}]
```

Um in Athena zu sehen, wie ein Ressourcentyp aufgenommen wurde, suchen Sie unter Tabellen und Ansichten danach. Um auf Elemente in diesem Array zuzugreifen, können Sie die Punktnotation verwenden. Hier ist ein einfaches Beispiel, das auf die Werte für `given` und zugreifen würde`family`.

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

- Element: MaritalStatus Die FHIR Spezifikation legt die Kardinalität als fest. `0..1`

  Dieses Element wird erfasst als. JSON

```
{
 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
}
```

  Um in Athena zu sehen, wie ein Ressourcentyp aufgenommen wurde, suchen Sie unter Tabellen und Ansichten danach. Um auf Schlüssel-Wert-Paare in der zuzugreifenJSON, können Sie die

Punktnotation verwenden. Da es sich nicht um ein Array handelt, ist kein Array-Index erforderlich. Hier ist ein einfaches Beispiel, das auf den Wert für zugreifen würde`text`.

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

Weitere Informationen zum Zugreifen und Suchen JSON finden Sie unter Abfragen JSON im Athena-Benutzerhandbuch.

Die Abfrageanweisungen der Athena Data Manipulation Language (DML) basieren auf Trino. Athena unterstützt nicht alle Funktionen von Trino, und es gibt signifikante Unterschiede. Weitere Informationen finden Sie unter DMLAbfragen, Funktionen und Operatoren im Amazon Athena Athena-Benutzerhandbuch.

Darüber hinaus unterstützt Athena mehrere Datentypen, auf die Sie bei der Erstellung von Abfragen Ihres HealthLake Datenspeichers stoßen können. Weitere Informationen zu Datentypen in Athena finden Sie unter Datentypen in Amazon Athena im Amazon Athena Athena-Benutzerhandbuch.

Weitere Informationen zur Funktionsweise von SQL Abfragen in Athena finden Sie in der SQLReferenz für Amazon Athena im Amazon Athena Athena-Benutzerhandbuch.

Jede Registerkarte zeigt Beispiele dafür, wie Sie mit Athena nach den angegebenen Ressourcentypen und zugehörigen Elementen suchen können.

Element: Extension

Das Element `extension` wird verwendet, um benutzerdefinierte Felder in einem Datenspeicher zu erstellen.

Dieses Beispiel zeigt Ihnen, wie Sie auf die Funktionen des `extension` Elements zugreifen, das im `Patient` Ressourcentyp gefunden wurde.

Wenn Ihr HealthLake Datenspeicher in Athena importiert wird, werden die Elemente eines Ressourcentyps unterschiedlich analysiert. Da es sich bei der Struktur um `element` eine Variable handelt, kann sie im Schema nicht vollständig spezifiziert werden. Um diese Variabilität zu handhaben, werden die Elemente innerhalb des Arrays als Zeichenketten übergeben.

In der Tabellenbeschreibung von finden Sie das Element`Patient`, das als `extension` beschrieben wird`array<string>`, was bedeutet, dass Sie mithilfe eines Indexwerts auf die

Elemente des Arrays zugreifen können. Um auf die Elemente der Zeichenfolge zuzugreifen, müssen Sie jedoch verwendenjson_extract.

Hier ist ein einzelner Eintrag aus dem extension Element, das in der Patiententabelle gefunden wurde.

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

Obwohl dies gültig istJSON, behandelt Athena es als Zeichenfolge.

Dieses SQL Abfragebeispiel zeigt, wie Sie eine Tabelle erstellen können, die die patient-birthPlace Elemente patient-mothersMaidenName und enthält. Um auf diese Elemente zuzugreifen, müssen Sie verschiedene Array-Indizes verwenden und json_extract.

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

Weitere Informationen zu Abfragen, die das beinhaltenJSON, finden Sie unter Extrahieren von Daten aus JSON im Amazon Athena Athena-Benutzerhandbuch.

Element: birthDate (Age)

Das Alter ist kein Element des Ressourcentyps Patient in. FHIR Hier sind zwei Beispiele für Suchanfragen, die nach Alter filtern.

Da das Alter kein Element ist, verwenden wir das `birthDate` für die SQL Abfragen. Um zu sehen, wie ein Element aufgenommen wurdeFHIR, suchen Sie unter Tabellen und Ansichten nach dem Tabellennamen. Sie können sehen, dass es vom Typ Zeichenfolge ist.

Beispiel 1: Berechnung eines Werts für das Alter

In dieser SQL Beispielabfrage verwenden wir ein integriertes SQL Tool, `current_date` `year` um diese Komponenten zu extrahieren. Dann subtrahieren wir sie, um das tatsächliche Alter eines Patienten in einer Spalte mit dem Namen `age` zurückzugeben.

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

Beispiel 2: Filterung nach Patienten, die früher geboren wurden `2019-01-01` und es sind`male`.

Die SQL Abfrage zeigt Ihnen, wie Sie die `CAST` Funktion verwenden, um das `birthDate` Element als Typ `DATE` umzuwandeln, und wie Sie anhand von zwei Kriterien in der `WHERE` Klausel filtern. Da das Element standardmäßig als Typstring aufgenommen wird, müssen wir `CAST` es als Typ eingeben`DATE`. Dann können Sie den < Operator verwenden, um es mit einem anderen Datum zu vergleichen,`2019-01-01`. Mithilfe `AND` von können Sie der `WHERE` Klausel ein zweites Kriterium hinzufügen.

```
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

Dieses Beispiel zeigt Suchen nach Orten innerhalb des Ressourcentyps Standort, bei denen der Stadtname Attleboro lautet.

```
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

In der Ressourcentyp-Bedingung werden Diagnosedaten gespeichert, die sich auf Probleme beziehen, die ein gewisses Maß an Besorgnis ausgelöst haben. HealthLakeDie integrierte medizinische Verarbeitung natürlicher Sprache (NLP) generiert neue `Condition` Ressourcen auf der Grundlage der im DocumentReference Ressourcentyp enthaltenen Angaben. Wenn neue Ressourcen generiert werden, wird das Tag HealthLake `SYSTEM_GENERATED` an das `meta` Element angehängt. Diese SQL Beispielabfrage zeigt, wie Sie die Bedingungstabelle durchsuchen und Ergebnisse zurückgeben können, bei denen die `SYSTEM_GENERATED` Ergebnisse entfernt wurden.

Weitere Informationen HealthLake zur integrierten Verarbeitung natürlicher Sprache (NLP) finden Sie unterVerwenden der automatisierten Ressourcengenerierung auf der Grundlage der Verarbeitung natürlicher Sprache (NLP) des FHIR DocumentReference Ressourcentyps in AWS HealthLake.

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

Sie können auch innerhalb eines bestimmten Zeichenkettenelements suchen, um Ihre Anfrage weiter zu filtern. Das `modifierextension` Element enthält Details darüber, welche `DocumentReference` Ressource zum Generieren einer Reihe von Bedingungen verwendet wurde. Auch hier müssen Sie verwenden, `json_extract` um auf die verschachtelten JSON Elemente zuzugreifen, die als Zeichenfolge in Athena importiert werden.

Diese SQL Beispielabfrage zeigt, wie Sie nach all dem suchen können`Condition`, was auf der Grundlage einer bestimmten Abfrage generiert wurde. `DocumentReference` Verwenden Sie

diese OptionCAST, um das JSON Element als Zeichenfolge festzulegen, sodass Sie es LIKE zum Vergleichen verwenden können.

```
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

Der Ressourcentyp Observation speichert Messwerte und einfache Aussagen zu einem Patienten, einem Gerät oder einer anderen Person. HealthLakeDie integrierte Verarbeitung natürlicher Sprache (NLP) generiert neue `Observation` Ressourcen auf der Grundlage von Details, die in einer `DocumentReference` Ressource gefunden wurden. Diese SQL Beispielabfrage beinhaltet `WHERE meta.tag[1] is NULL` auskommentiert, was bedeutet, dass die `SYSTEM_GENERATED` Ergebnisse enthalten sind.

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

Diese Spalte wurde als importiert `struct`. Daher können Sie mithilfe der Punktnotation auf die darin enthaltenen Elemente zugreifen.

Resource type: MedicationStatement

MedicationStatement ist ein FHIR Ressourcentyp, mit dem Sie Details zu Medikamenten speichern können, die ein Patient eingenommen hat, einnimmt oder in future einnehmen wird. HealthLakeDie integrierte medizinische Verarbeitung natürlicher Sprache (NLP) generiert neue MedicationStatement Ressourcen auf der Grundlage von Dokumenten, die im DocumentReference Ressourcentyp gefunden wurden. Wenn neue Ressourcen generiert werden, wird das Tag HealthLake `SYSTEM_GENERATED` an das `meta` Element angehängt. Diese SQL Beispielabfrage zeigt, wie Sie eine Abfrage erstellen, die nach einem einzelnen Patienten anhand seiner Kennung filtert und Ressourcen findet, die von HealthLake s integrated NLP hinzugefügt wurden.

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

Weitere Informationen zu HealthLake Integrated Medical NLP finden Sie unterVerwenden der automatisierten Ressourcengenerierung auf der Grundlage der Verarbeitung natürlicher Sprache (NLP) des FHIR DocumentReference Ressourcentyps in AWS HealthLake.

SQLBeispielabfragen mit komplexer Filterung

> **Note**
>
> Nach dem 20. Februar 2023 verwenden HealthLake Datenspeicher standardmäßig keine integrierte Verarbeitung natürlicher Sprache (NLP). Wenn Sie daran interessiert sind, diese Funktion in Ihrem Datenspeicher zu aktivieren, finden Sie Wie aktiviere ich die HealthLake integrierte Funktion zur Verarbeitung natürlicher Sprache? weitere Informationen im Kapitel Fehlerbehebung.

Zu den Beispielen in diesem Thema gehören SQL Abfragen für die HealthLake Integration mit Athena, die komplexe Filterung verwenden.

Example Erstellung von Filterkriterien, die auf demografischen Daten basieren

Bei der Erstellung einer Patientenkohorte ist es wichtig, die richtigen demografischen Daten der Patienten zu ermitteln. Diese Beispielabfrage zeigt, wie Sie die Trino-Punktnotation verwenden und `json_extract` Daten in Ihrem HealthLake Datenspeicher filtern können.

```
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;
```

Mit der Athena-Konsole können Sie die Ergebnisse weiter sortieren und herunterladen.

Example Filter für einen Patienten und die damit verbundenen Erkrankungen erstellen

Diese Beispielabfrage zeigt, wie Sie alle verwandten Erkrankungen für die in einem HealthLake Datenspeicher gefundenen Patienten suchen und sortieren können.

```
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;
```

Sie können die Athena-Konsole verwenden, um diese Ergebnisse weiter zu sortieren oder sie zur weiteren Analyse herunterzuladen.

Example Filter für einen Patienten und die damit verbundenen Beobachtungen erstellen

Diese Beispielabfrage zeigt, wie Sie alle zugehörigen Beobachtungen für die in einem HealthLake Datenspeicher gefundenen Patienten suchen und sortieren können.

```
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 Erstellen von Filterbedingungen für einen Patienten und die damit verbundenen Verfahren

Die Verknüpfung von Verfahren und Patienten ist ein wichtiger Aspekt des Gesundheitswesens. Diese SQL Abfrage zeigt, wie Sie dazu die Ressourcentypen Patient und Verfahren in Athena verwenden können. Diese SQL Abfrage gibt alle Patienten und die zugehörigen Verfahren zurück, die sich in Ihrem HealthLake Datenspeicher befinden.

```
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;
```

Jetzt können Sie die Ergebnisse mit der Athena-Konsole für weitere Analysen herunterladen oder sie sortieren, um die Ergebnisse besser zu verstehen.

Example Erstellung von Filterbedingungen für einen Patienten und die zugehörigen Verschreibungen

Es ist wichtig, eine aktuelle Liste der Medikamente zu sehen, die Patienten einnehmen. Mit Athena können Sie eine SQL Abfrage schreiben, die sowohl den Patienten- als auch den MedicationRequest Ressourcentyp verwendet, der sich in Ihrem HealthLake Datenspeicher befindet.

Diese SQL Abfrage verbindet den Patienten und die in Athena importierten MedicationRequest Tabellen. Außerdem werden die Rezepte mithilfe der Punktnotation in ihre einzelnen Einträge unterteilt.

```
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
```

Sie können die Athena-Konsole verwenden, um die Ergebnisse zu sortieren oder sie zur weiteren Analyse herunterzuladen.

Example Im MedicationStatement Ressourcentyp gefundene Medikamente werden angezeigt

Die Beispielabfrage zeigt Ihnen, wie Sie die in Athena JSON importierten verschachtelten Dateien mithilfe von organisieren. SQL Die Abfrage verwendet das `meta` Element, um anzugeben, wann ein Medikament durch HealthLake die integrierte Verarbeitung natürlicher Sprache () NLP hinzugefügt wurde. Weitere Informationen HealthLake zur Integration mit Amazon Comprehend Medical finden Sie unter. Verwenden der automatisierten Ressourcengenerierung auf der Grundlage der Verarbeitung natürlicher Sprache (NLP) des FHIR DocumentReference Ressourcentyps in AWS HealthLake Es wird auch verwendet`json_extract`, um nach Daten innerhalb des String-Arrays zu suchen. JSON

```
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;
```

Sie können die Athena-Konsole verwenden, um diese Ergebnisse herunterzuladen oder zu sortieren.

Example Filtern Sie nach einem bestimmten Krankheitstyp

Das Beispiel zeigt, wie Sie eine Gruppe von Patienten im Alter von 18 bis 75 Jahren finden können, bei denen Diabetes diagnostiziert wurde.

```
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%');
```

Jetzt können Sie die Ergebnisse mit der Athena-Konsole sortieren oder zur weiteren Analyse herunterladen.

AWS HealthLake und VPC Schnittstellenendpunkte ()AWS PrivateLink

Sie können eine private Verbindung zwischen Ihrem VPC und herstellen, AWS HealthLake indem Sie einen VPCSchnittstellenendpunkt erstellen. VPCSchnittstellenendpunkte basieren auf einer Technologie AWS PrivateLink, mit der Sie privat zugreifen HealthLake können, APIs ohne dass ein Internet-Gateway, ein NAT Gerät, eine Verbindung oder AWS Direct Connect eine VPN Verbindung erforderlich ist. Instanzen in Ihrem System benötigen VPC keine öffentlichen IP-Adressen, um mit ihnen zu kommunizieren HealthLake;. APIs Der Verkehr zwischen Ihnen VPC und HealthLake; verlässt das Amazon-Netzwerk nicht.

Jeder Schnittstellenendpunkt wird durch eine oder mehrere Elastic-Network-Schnittstellen in Ihren Subnetzen dargestellt.

Weitere Informationen finden Sie unter Interface VPC Endpoints (AWS PrivateLink) im VPCAmazon-Benutzerhandbuch.

Überlegungen zu Endpunkten HealthLake VPC

Bevor Sie einen VPC Schnittstellenendpunkt für einrichten HealthLake, sollten Sie die Eigenschaften und Einschränkungen des Schnittstellenendpunkts im VPCAmazon-Benutzerhandbuch lesen.

HealthLake unterstützt das Aufrufen all seiner API Aktionen von Ihrem ausVPC.

Erstellen eines VPC Schnittstellenendpunkts für HealthLake;

Sie können einen VPC Endpunkt für den HealthLake; -Service entweder mit der VPC Amazon-Konsole oder mit der AWS Command Line Interface (AWS CLI) erstellen. Weitere Informationen finden Sie unter Erstellen eines Schnittstellenendpunkts im VPCAmazon-Benutzerhandbuch.

Erstellen Sie einen VPC Endpunkt für HealthLake; unter Verwendung des folgenden Servicenamens:

- com.amazonaws. *region*. Gesundheitssee

Wenn Sie privat DNS für den Endpunkt aktivieren, können Sie API Anfragen an die HealthLake Verwendung des DNS Standardnamens für die Region stellen. Beispiel, *healthlake.us-east-1.amazonaws.com*.

Weitere Informationen finden Sie unter Zugreifen auf einen Service über einen Schnittstellenendpunkt im VPCAmazon-Benutzerhandbuch.

Erstellen einer VPC Endpunktrichtlinie für HealthLake

Sie können Ihrem VPC Endpunkt eine Endpunktrichtlinie hinzufügen, die den Zugriff darauf steuert HealthLake. Die Richtlinie gibt die folgenden Informationen an:

- Prinzipal, der die Aktionen ausführen kann.
- Aktionen, die ausgeführt werden können
- Die Ressourcen, für die Aktionen ausgeführt werden können.

Weitere Informationen finden Sie unter Steuern des Zugriffs auf Dienste mit VPC Endpunkten im VPCAmazon-Benutzerhandbuch.

Beispiel: VPC Endpunktrichtlinie für Aktionen HealthLake

Das Folgende ist ein Beispiel für eine Endpunktrichtlinie für HealthLake. Wenn diese Richtlinie an einen Endpunkt angehängt ist, gewährt sie allen Prinzipalen auf allen Ressourcen Zugriff auf die HealthLake `CreateFHIRDatastore` Aktion.

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

Ressourcen taggen in AWS HealthLake

Sie können AWS-Ressourcen Metadaten in Form von Tags zuweisen. Jedes Tag ist ein Label, das aus einem benutzerdefinierten Schlüssel und Wert besteht. Mit Tags können Sie Ressourcen verwalten, identifizieren, organisieren, suchen und filtern.

In diesem Thema werden häufig verwendete Tagging-Kategorien und -Strategien beschrieben, mit denen Sie eine konsistente und effektive Tagging-Strategie implementieren können. In den folgenden Abschnitten werden Grundkenntnisse in den Bereichen AWS Ressourcen, Tagging, detaillierte Abrechnung sowie AWS Identity and Access Management (IAM) vorausgesetzt.

Jedes Tag besteht aus zwei Teilen:

- Ein Tag-Schlüssel (zum CostCenter Beispiel Environment oder Project). Bei Tag-Schlüsseln wird zwischen Groß- und Kleinschreibung unterschieden.
- Ein Tag-Wert (zum Beispiel 111122223333 oder Production). Wie bei Tag-Schlüsseln wird auch bei Tag-Werten zwischen Groß- und Kleinschreibung unterschieden.

Sie können Tags verwenden, um Ressourcen nach Zweck, Eigentümer, Umgebung oder anderen Kriterien zu kategorisieren. Weitere Informationen finden Sie unter Tagging-Strategien in AWS.

Sie können Tags für jede Ressource einzeln über die Servicekonsole, den Service oder den hinzufügen, ändern oder entfernen. API AWS CLI

Um das Tagging zu aktivieren, stellen Sie sicher, dass Sie autorisiert TagResources sind. Sie können die Autorisierung vornehmen, TagResources indem Sie eine IAM Richtlinie wie im folgenden Beispiel anhängen.

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

```
            "Resource": "*"
        }
    ]
}
```

Wichtiger Hinweis

AWS HealthLake schützt Kundendaten gemäß den Richtlinien des Modells der AWS gemeinsamen Verantwortung. Das bedeutet, dass alle Kundendaten sowohl bei der Übertragung als auch bei der Speicherung verschlüsselt werden. Allerdings sind nicht alle vom Kunden eingegebenen Namen für Datenspeicher oder auftragsbasierte Vorgänge verschlüsselt. Sie sollten niemals persönlich identifizierbare Informationen oder geschützte Gesundheitsinformationen enthalten. Weitere Informationen finden Sie im Kapitel AWS HealthLake Sicherheit.

Bewährte Methoden

Befolgen Sie beim Erstellen einer Tagging-Strategie für AWS-Ressourcen die folgenden bewährten Methoden:

- Speichern Sie keine persönlich identifizierbaren Informationen (PII), persönlichen Gesundheitsinformationen (PHI) oder andere vertrauliche Informationen in Tags.
- Verwenden Sie für Tags ein standardisiertes Format, bei dem die Groß-/Kleinschreibung beachtet wird, und wenden Sie es konsistent für alle Ressourcentypen an.
- Verwenden Sie Tag-Richtlinien, die mehrere Zwecke unterstützen, wie die Verwaltung der Ressourcenzugriffskontrolle, Kostenverfolgung, Automatisierung und Organisation.
- Verwenden Sie automatisierte Tools, um Ressourcen-Tags zu verwalten. AWS Resource Groups und das Resource Groups Groups-Tagging API ermöglichen die programmatische Steuerung von Tags, sodass Tags und Ressourcen automatisch verwaltet, gesucht und gefiltert werden können.
- Tagging ist effektiver, wenn Sie mehr Tags verwenden.
- Tags können bearbeitet oder geändert werden, wenn sich die Benutzeranforderungen ändern. Um Zugriffskontroll-Tags zu aktualisieren, müssen Sie jedoch auch die Richtlinien aktualisieren, die auf diese Tags verweisen, um den Zugriff auf Ihre Ressourcen zu kontrollieren.

Anforderungen zum Markieren

Für Tags gelten zwei Anforderungen:

- Schlüsseln darf nicht das Präfix aws: vorangestellt werden.
- Schlüssel müssen in einem Tag-Satz eindeutig sein.
- Schlüssel müssen zwischen 1 und 128 Zeichen lang sein.
- Ein Wert muss zwischen 0 und 256 Zeichen haben.
- Werte brauchen pro Tag-Satz nicht eindeutig zu sein.
- Zulässige Zeichen für Schlüssel und Werte sind Unicode-Buchstaben, Ziffern, Leerzeichen sowie die folgenden Sonderzeichen: _ . : / = + - @.
- Bei Schlüsseln und Werten wird die Groß-/Kleinschreibung berücksichtigt.

Hinzufügen eines Tags zu einem Datenspeicher

Das Hinzufügen von Tags zu einem Datenspeicher kann Ihnen helfen, Ihre AWS Ressourcen zu identifizieren und zu organisieren und den Zugriff darauf zu verwalten. Zunächst fügen Sie einem Datenspeicher ein oder mehrere Tags (Schlüssel-Wert-Paare) hinzu. Sie können bis zu fünfzig Tags pro Benutzer verwenden. Es gibt auch Einschränkungen hinsichtlich der Zeichen, die Sie in den Schlüssel- und Wertfeldern verwenden können.

Sobald Sie über Tags verfügen, können Sie IAM Richtlinien erstellen, um den Zugriff auf den Datenspeicher auf der Grundlage dieser Tags zu verwalten. Sie können die HealthLake Konsole oder die verwenden AWS CLI , um einem Datenspeicher Tags hinzuzufügen. Das Hinzufügen von Tags zu einem Repository kann Auswirkungen auf den Zugriff auf dieses Repository haben. Bevor Sie einem Datenspeicher ein Tag hinzufügen, sollten Sie alle IAM Richtlinien überprüfen, die Tags verwenden könnten, um den Zugriff auf Ressourcen wie Datenspeicher zu steuern.

Gehen Sie wie folgt vor AWS CLI , um einem HealthLake Datenspeicher ein Tag hinzuzufügen. Informationen zum Hinzufügen eines Tags zu einem Datenspeicher, wenn Sie ihn erstellen, finden Sie unterErstellen eines Datenspeichers in AWS HealthLake.

Führen Sie im Terminal oder in der Befehlszeile den Befehl tag-resource aus und geben Sie den Amazon-Ressourcennamen (ARN) des Datenspeichers an, dem Sie Tags hinzufügen möchten, sowie den Schlüssel und Wert des Tags, den Sie hinzufügen möchten. Sie können einem Datenspeicher mehr als ein Tag hinzufügen. Es gibt auch Einschränkungen in Bezug auf die Zeichen, die Sie in den Schlüssel- und Wertfeldern verwenden können. Eine Liste finden Sie unter Wenn Sie Anforderungen zum Markieren beispielsweise einem Datenspeicher während der Erstellung Tags hinzufügen möchten, verwenden Sie den folgenden Befehl in der AWS CLI. Der Name des

Datenspeichers lautet Test_Data_Store, und die beiden hinzugefügten Tags mit Schlüsseln sind key1 und key2 mit Werten wie Wert1 bzw. Wert2 :

```
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
```

Um einem vorhandenen Datenspeicher Tags hinzuzufügen, würden Sie den folgenden Beispielbefehl ausführen:

```
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
```

Bei Erfolg gibt dieser Befehl keine Antwort zurück.

Auflisten von Tags für einen Datenspeicher

Gehen Sie wie folgt vor AWS CLI , um mit dem eine Liste der AWS Tags für einen HealthLake Datenspeicher anzuzeigen. Wenn keine Tags hinzugefügt wurden, ist die zurückgegebene Liste leer.

Führen Sie den Befehl im Terminal oder in der list-tags-for-resource Befehlszeile aus, wie im folgenden Beispiel gezeigt.

```
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"
    }
}
```

Tags aus einem Datenspeicher entfernen

Sie können ein oder mehrere Tags entfernen, die einem Datenspeicher zugeordnet sind. Das Entfernen eines Tags löscht nicht das Tag anderer AWS-Ressourcen, die mit diesem Tag verknüpft sind.

Führen Sie im Terminal oder in der Befehlszeile den Befehl untag-resource aus und geben Sie den Amazon-Ressourcennamen (ARN) des Datenspeichers an, in dem Sie Tags entfernen möchten, und den Tag-Schlüssel des Tags, das Sie entfernen möchten.

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

Bei Erfolg gibt dieser Befehl keine Antwort zurück. Führen Sie den list-tags-for-resource Befehl aus, um die mit dem Datenspeicher verknüpften Tags zu überprüfen.

Überwachung HealthLake

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Verfügbarkeit und Leistung Ihrer HealthLake anderen AWS Lösungen. AWSbietet die folgenden Überwachungstools, mit denen Sie beobachten HealthLake, melden können, wenn etwas nicht stimmt, und gegebenenfalls automatische Maßnahmen ergreifen können:

- Amazon CloudWatch überwacht Ihre AWS Ressourcen und die Anwendungen, auf denen Sie laufen, AWS in Echtzeit. Sie können Kennzahlen sammeln und verfolgen, benutzerdefinierte Dashboards erstellen und Alarme einrichten, die Sie benachrichtigen oder Maßnahmen ergreifen, wenn eine bestimmte Metrik einen bestimmten Schwellenwert erreicht. Sie können beispielsweise die CPU Nutzung oder andere Kennzahlen Ihrer EC2 Amazon-Instances CloudWatch verfolgen und bei Bedarf automatisch neue Instances starten. Weitere Informationen finden Sie im CloudWatch Amazon-Benutzerhandbuch.
- AWS CloudTrailerfasst API Anrufe und damit verbundene Ereignisse, die von oder im Namen Ihres AWS Kontos getätigt wurden. Der Service gibt die Protokolldateien in einen Amazon S3-Bucket aus, den Sie zuvor angegeben haben. Sie können feststellen, welche Benutzer und Konten angerufen wurden AWS, welche Quell-IP-Adresse für diese Anrufe verwendet wurde und wann sie stattfanden. Weitere Informationen finden Sie im AWS CloudTrail -Benutzerhandbuch.

Themen

- Überwachung HealthLake mit Amazon CloudWatch

Überwachung HealthLake mit Amazon CloudWatch

Sie können die HealthLake Nutzung überwachen CloudWatch, wobei Rohdaten gesammelt und zu lesbaren Kennzahlen verarbeitet werden, die nahezu in Echtzeit verfügbar sind. Diese Statistiken werden 15 Monate lang aufbewahrt, sodass Sie diese historischen Informationen nutzen und sich einen besseren Überblick über die Leistung Ihrer Webanwendung oder Ihres Dienstes verschaffen können. Sie können auch Alarme einrichten, die auf bestimmte Grenzwerte achten und Benachrichtigungen senden oder Aktivitäten auslösen, wenn diese Grenzwerte erreicht werden. Weitere Informationen finden Sie im CloudWatch Amazon-Benutzerhandbuch.

Metriken werden für alle gemeldet HealthLake APIs, einschließlich der folgenden.

- Datenspeicherverwaltung APIs —CreateFHIRDatastore, DeleteFHIRDatastore, DescribeFHIRDatastore, L istFHIRDatastores
- Importieren und Exportieren APIs tartFHIRImport —S-Job, istFHIRImport L-Jobs, escribeFHIRImport D-Job, tartFHIRExport S-Job, istFHIRExport L-Jobs, escribeFHIRExport D-Job
- HTTPRESTKunden- und Ressourcenmanagement APIs — CreateResource, DeleteResource, GetCapabilities, ReadResource, SearchAll, SearchWithGet SearchWithPost, UpdateResource.
- Markieren APIs — ListTagsForResource,, TagResource UntagResource

Die folgende Tabelle listet die Metriken und Dimensionen für HealthLake auf.

Die folgenden Kennzahlen werden gemeldet. Jede Zahl wird als Frequenzzählung für einen vom Benutzer angegebenen Datenbereich dargestellt.

Metriken

Metriken	Beschreibung
Anzahl der Anrufe	Die Anzahl der Anrufe anAPIs. Dies kann entweder für das Konto oder einen bestimmten Datenspeicher gemeldet werden. Einheiten: Anzahl Gültige Statistiken: Summe, Anzahl Dimensionen: Vorgang, Datenspeicher-ID, Datenspeichertyp
Erfolgreiche Anfragen	Die Anzahl der erfolgreichen API Anfragen. Einheiten: Anzahl Gültige Statistiken: Summe, Durchschnitt Dimensionen: Betrieb, Datenspeicher, Datenspeichertyp
Benutzerfehler	Die Anzahl der Anfragen, die aufgrund eines Benutzerfehlers fehlgeschlagen sind.

Metriken	Beschreibung
	Einheiten: Anzahl Gültige Statistiken: Summe, Durchschnitt Dimensionen: Vorgang, Datenspeicher-ID, Datenspeichertyp
Serverfehler	Die Anzahl der Anfragen, die aufgrund eines Serverfehlers fehlgeschlagen sind. Einheiten: Anzahl Gültige Statistiken: Summe, Durchschnitt Dimensionen: Vorgang, Datenspeicher-ID, Datenspeichertyp
Gedrosselte Anforderungen	Die Anzahl der Anfragen, die gedrosselt wurden. Diese Metrik ist nicht in der Anzahl der Benutzer- oder Serverfehler enthalten. Einheiten: Anzahl Gültige Statistiken: Summe, Durchschnitt Dimensionen: Vorgang, Datenspeicher-ID, Datenspeichertyp
Latency	Die Zeit in Millisekunden, die für die Verarbeit ung der Benutzeranfrage benötigt wurde. Einheit: Millisekunden Gültige Statistiken: Minimum, Maximum, Summe, Durchschnitt Abmessungen: Vorgang, Datenspeicher-ID, Datenspeichertyp

Die folgenden Dimensionen werden gemeldet.

Dimensionen

Dimensionen	Beschreibung
Operation	Welche API Operation wurde verwendet
DataStoreAusweis	Der in der API Anfrage enthaltene Datenspeicher
DataStoreType	Der Typ des Datenspeichers (derzeit wird nur FHIR R4 unterstützt)

Sie können Metriken für HealthLake mit der AWS Management Console AWS CLI, dem oder dem CloudWatch API abrufen. Sie können das CloudWatch API über eines der Amazon AWS Software Development Kits (SDKs) oder die CloudWatch API Tools verwenden. Die HealthLake Konsole zeigt Diagramme an, die auf den Rohdaten von basieren CloudWatch API.

Sie müssen über die entsprechenden CloudWatch Berechtigungen für die Überwachung HealthLake verfügen CloudWatch. Weitere Informationen finden Sie unter Authentifizierung und Zugriffskontrolle für Amazon CloudWatch im CloudWatch Amazon-Benutzerhandbuch.

HealthLake Metriken anzeigen

Um Metriken anzuzeigen (CloudWatch Konsole)

1. Melden Sie sich bei der AWS-Managementkonsole an und öffnen Sie die CloudWatch-Konsole.
2. Wählen Sie Metriken, Alle Metriken und dann AWS/HealthLake.
3. Wählen Sie die Dimension, den Namen einer Metrik und schließlich Add to graph (Dem Diagramm hinzufügen) aus.
4. Wählen Sie einen Wert für den Datumsbereich aus. Die Anzahl der Kennzahlen für den ausgewählten Zeitraum wird im Diagramm angezeigt.

Einen Alarm erstellen mit CloudWatch

Ein CloudWatch Alarm überwacht eine einzelne Metrik über einen bestimmten Zeitraum und führt eine oder mehrere Aktionen aus: das Senden einer Benachrichtigung an ein Amazon Simple

Notification Service (AmazonSNS) -Thema oder an eine Auto Scaling Scaling-Richtlinie. Die Aktion oder Aktionen basieren auf dem Wert der Metrik im Verhältnis zu einem bestimmten Schwellenwert über eine von Ihnen angegebene Anzahl von Zeiträumen. CloudWatch kann Ihnen auch eine SNS Amazon-Nachricht senden, wenn sich der Zustand des Alarms ändert.

CloudWatch Alarme lösen nur dann Aktionen aus, wenn sich der Status ändert und für den von Ihnen angegebenen Zeitraum andauert.

Um Metriken anzuzeigen (Konsole) CloudWatch

1. Melden Sie sich bei der AWS-Managementkonsole an und öffnen Sie die CloudWatch-Konsole.
2. Wählen Sie Alarms und dann Create Alarm.
3. Wählen Sie AWS/HealthLakeund wählen Sie dann eine Metrik aus.
4. Wählen Sie für Time Range den zu überwachenden Zeitbereich und dann Next.
5. Geben Sie Name (Name) und Description (Beschreibung) ein.
6. Wählen Sie für Whenever >= und geben Sie einen Maximalwert ein.
7. Wenn Sie eine E-Mail senden CloudWatch möchten, wenn der Alarmstatus erreicht ist, wählen Sie im Bereich Aktionen für Wann immer dieser Alarm ist die Option Status ist ALARM. Wählen Sie unter Benachrichtigung senden an eine Mailingliste aus oder wählen Sie Neue Liste und erstellen Sie eine neue Mailingliste.
8. Nutzen Sie die Alarmvorschau im Bereich Alarm Preview. Wenn Sie mit dem Alarm zufrieden sind, wählen Sie Create Alarm.

SMARTIntegrieren FHIR mit AWS HealthLake

Ein FHIR aktivierter HealthLake Datenspeicher für austauschbare medizinische Anwendungen und wiederverwendbare Technologien (SMART) ermöglicht FHIR kompatiblen Anwendungen den Zugriff SMART auf Daten, die in einem HealthLake Datenspeicher gespeichert sind. HealthLake Der Zugriff auf Daten erfolgt durch Authentifizierung und Autorisierung von Anfragen mithilfe eines Autorisierungsservers eines Drittanbieters und durch Einrichtung zusätzlicher Ressourcen in. AWS

Um SMART on FHIR mit Ihrem HealthLake Datenspeicher zu verwenden, müssen Sie in Ihrer reateFHIRDatastoreAPIC-Anfrage Folgendes angeben.

- Setzen Sie den AuthorizationStrategyWert gleich auf`SMART_ON_FHIR_V1`.
- Stellen Sie den IdpLambdaArn Wert auf den ARN Wert ein, den AWS Lambda Sie erstellt haben, um die Token-Dekodierung mit Ihrem Autorisierungsserver zu verwalten.
- Definieren Sie die auf Ihrem Autorisierungsserver angegebenen Metadatenelemente. Diese Metadatenelemente werden im Discovery-Dokument zurückgegeben. Weitere Informationen hierzu finden Sie unter Abrufen des SMART Discovery-Dokuments eines HealthLake Datenspeichers, der FHIR aktiviert ist.
- Optional: Aktivieren Sie diese Option, FineGrainedAuthorizationEnabledwenn Sie auf Ihrem Autorisierungsserver eine detaillierte Autorisierung eingerichtet haben.

Sie können mit AWS Command Line Interface (AWS CLI) oder SMART über einen der AWS unterstützten SDKs Datenspeicher einen FHIR aktivierten Datenspeicher einrichten. Das Erstellen eines SMART HealthLake Datenspeichers, der FHIR aktiviert ist, wird über die HealthLake Konsole nicht unterstützt. Weitere Informationen hierzu finden Sie unter Erstellen Sie einen Datenspeicher, SMART der FHIR aktiviert ist.

Um diese Parameter in der Anforderung vorzuschreiben, müssen Sie Ressourcen in anderen AWS Diensten (AWS Secrets Manager und AWS Lambda) einrichten, neue IAM Dienstrollen erstellen und einen autorisierten Server einrichten, SMART der nicht FHIR konform ist. Im Abschnitt Einrichtung der erforderlichen Ressourcen für die Implementierung eines SMART FHIR On-kompatiblen Datenspeichers erfahren Sie mehr über die Einrichtung der erforderlichen Ressourcen und erhalten einen allgemeinen Überblick darüber, wie eine SMART FHIR On-Anwendung damit interagiert. HealthLake

Das bedeutet, dass AWS Identity and Access Management Sie die Benutzeranmeldedaten nicht SMART über einen FHIR kompatiblen Autorisierungsserver verwalten.

HealthLake unterstützt SMART auf FHIR 1.0. Weitere Informationen zu diesem Framework finden Sie im SMARTApplication Launch Framework Implementation Guide, Version 1.0.

Um Anfragen für Datenspeicher mithilfe von SMART on zu autorisieren und zu authentifizierenFHIR, HealthLake unterstützt es die Verwendung von:

- OpenID (AuthN) -Integration: Wird verwendet, um zu authentifizieren, wer (oder was) diese Person oder Client-Anwendung vorgibt zu sein.
- OAuth2.0 (AuthZ) -Integration: Wird verwendet, um zu autorisieren, welche FHIR Ressourcen in Ihrem HealthLake Datenspeicher eine authentifizierte Anfrage auch Daten lesen oder schreiben kann. Dies wird durch die Bereiche definiert, die auf Ihrem Autorisierungsserver eingerichtet sind

Inhalt

Authentifizierungsanforderungen SMART für FHIR

Um SMART auf FHIR Ressourcen in einem eigenen FHIR HealthLake Datenspeicher zuzugreifen, muss eine Client-Anwendung von einem OAuth 2.0-kompatiblen Autorisierungsserver autorisiert werden und ein OAuth Bearer-Token als Teil einer FHIR REST API Anfrage vorlegen. Um den Endpunkt des Autorisierungsservers zu ermitteln, verwenden Sie das HealthLake SMART On FHIR Discovery-Dokument über einen bekannten Uniform Resource Identifier. Weitere Informationen zu diesem Prozess finden Sie unter Abrufen des SMART Discovery-Dokuments eines HealthLake Datenspeichers, der FHIR aktiviert ist.

Wenn Sie einen SMART FHIR HealthLake On-Datenspeicher erstellen, müssen Sie den Endpunkt des Autorisierungsservers und den Token-Endpunkt im `metadata` Element der reateFHIRDatastore C-Anfrage definieren. Weitere Informationen zur Definition des `metadata` Elements finden Sie unterEinen SMART HealthLake Datenspeicher mit FHIR aktivierter Aktivierung erstellen.

Mithilfe der Endpunkte des Autorisierungsservers authentifiziert die Client-Anwendung einen Benutzer beim Autorisierungsdienst. Nach der Autorisierung und Authentifizierung wird ein JSON Web-Token (JWT) vom Autorisierungsdienst generiert und an die Client-Anwendung übergeben. Dieses Token enthält FHIR Ressourcenbereiche, die die Client-Anwendung verwenden darf, was wiederum einschränkt, auf welche Daten der Benutzer zugreifen kann. Wenn der Startbereich angegeben wurde, enthält die Antwort optional diese Details. Weitere Informationen zu den FHIR On-Scopes, die SMART von unterstützt werden HealthLake, finden Sie unterUnterstützt SMART auf FHIR OAuth Bereichen von HealthLake.

Unter Verwendung des durch die Autorisierung JWT gewährten Servers FHIR REST API ruft eine Client-Anwendung einen HealthLake Datenspeicher SMART auf, der FHIR aktiviert ist. Um das zu validieren und zu dekodierenJWT, müssen Sie eine Lambda-Funktion erstellen. HealthLake ruft diese Lambda-Funktion in Ihrem Namen auf, wenn eine FHIR REST API Anfrage eingeht. Ein Beispiel für eine Lambda-Starterfunktion finden Sie unterWird AWS Lambda für die Tokenvalidierung mit einem SMART FHIR eingeschalteten HealthLake Datenspeicher verwendet.

Elemente des Autorisierungsservers, die zum Erstellen eines HealthLake Datenspeichers mit SMART FHIR aktivierter Aktivierung erforderlich sind

In der reateFHIRDatastore C-Anfrage müssen Sie den Autorisierungsendpunkt und den Token-Endpunkt als Teil des `metadata` Elements im `IdentityProviderConfiguration` Objekt angeben. Sowohl der Autorisierungsendpunkt als auch der Token-Endpunkt sind erforderlich.
Ein Beispiel dafür, wie dies in einer reateFHIRDatastore C-Anfrage angegeben wird, finden Sie unterEinen SMART HealthLake Datenspeicher mit FHIR aktivierter Aktivierung erstellen.

Erforderliche Ansprüche, um eine FHIR REST API Anfrage in einem HealthLake Datenspeicher abzuschließen, der nicht SMART FHIR aktiviert ist

Ihre AWS Lambda Funktion muss die folgenden Ansprüche enthalten, damit es sich um eine gültige FHIR REST API Anforderung für einen HealthLake Datenspeicher handelt, der nicht SMART FHIR aktiviert ist.

- `nbf`: Anspruch (nicht vorher) — Der Anspruch „nbf“ (nicht vorher) gibt den Zeitpunkt an, bis zu dem der Antrag zur Bearbeitung angenommen JWT MUST NOT wird. Für die Bearbeitung des Antrags mit dem Namen „Nbf“ ist es erforderlich, dass der aktuelle, im Antrag „Nbf“ date/time MUST be after or equal to the not-before date/time aufgeführte Betrag. Die von uns bereitgestellte Lambda-Beispielfunktion konvertiert `iat` von der Serverantwort in`nbf`.
- `exp`: (Ablaufzeit) -Anspruch — Der Anspruch „exp“ (Ablaufzeit) gibt die Ablaufzeit an oder nach der der Antrag nicht zur Bearbeitung akzeptiert werden JWT darf.
- `isAuthorized`: Ein boolescher Wert, der auf gesetzt ist. `True` Zeigt an, dass die Anfrage auf dem Autorisierungsserver autorisiert wurde.
- `aud`: (Zielgruppenanspruch) — Der Anspruch „aud“ (Zielgruppe) identifiziert die Empfänger, für die der bestimmt JWT ist. Dies muss ein SMART eingeschalteter FHIR HealthLake Datenspeicher-Endpunkt sein.

- `scope`: Dies muss mindestens ein FHIR ressourcenbezogener Bereich sein. Dieser Bereich ist auf Ihrem Autorisierungsserver definiert. Weitere Informationen zu FHIR ressourcenbezogenen Bereichen, die von akzeptiert werden HealthLake, finden Sie unterHealthLake FHIRressourcenspezifische Bereiche des Datenspeichers.

Unterstützt SMART auf FHIR OAuth Bereichen von HealthLake

HealthLake verwendet OAuth 2.0 als Autorisierungsprotokoll. Wenn Sie dieses Protokoll auf Ihrem Autorisierungsserver verwenden, können Sie definieren, auf welche FHIR Ressourcen in Ihrem HealthLake Datenspeicher eine Client-Anwendung auch Lese- und/oder Schreibzugriff haben kann.

Das SMART FHIR On-Framework definiert eine Reihe von Bereichen, die vom Autorisierungsserver angefordert werden können. Die FHIRBereichsdefinitionen im SMART FHIR On-Framework finden Sie SMART unter Bereiche im HL7FHIRRessourcenleitfaden.

Beispielsweise sollte eine Client-Anwendung, die nur darauf ausgelegt ist, dass Patienten ihre Laborergebnisse oder ihre Kontaktdaten einsehen können, nur berechtigt sein, (auf FHIR REST Anfrage) `read` Geltungsbereiche anzufordern. Um diese als Bereich zu definieren, würden Sie eine Zeichenfolge wie die folgende `patient/Observation.read` angeben. Dies würde es der Client-Anwendung ermöglichen, schreibgeschützt Zugriff auf den `Observation` Ressourcentyp für den `Patient` Ressourcentyp anzufordern.

Eigenständiger Startbereich

HealthLake unterstützt den Bereich des eigenständigen Startmodus`launch/patient`.

Im eigenständigen Startmodus fordert eine Client-Anwendung Zugriff auf die klinischen Daten des Patienten an, da der Benutzer und der Patient der Client-Anwendung nicht bekannt sind. Daher fordert die Autorisierungsanfrage der Client-Anwendung ausdrücklich die Rückgabe des Patientenbereichs an. Nach erfolgreicher Authentifizierung gibt der Autorisierungsserver ein Zugriffstoken aus, das den angeforderten Patientenstartbereich enthält. Der benötigte Patientenkontext wird zusammen mit dem Zugriffstoken in der Antwort des Autorisierungsservers bereitgestellt.

Unterstützte Bereiche für den Startmodus

Scope	Beschreibung
`launch/patient`	Ein Parameter in einer OAuth 2.0-Autorisierungsanfrage, der anfordert, dass Patientendaten in der Autorisierungsantwort zurückgegeben werden.

HealthLake FHIRressourcenspezifische Bereiche des Datenspeichers

HealthLake definiert drei Ebenen von Bereichen.

- Patientenspezifische Bereiche gewähren Zugriff auf spezifische Daten über einen einzelnen Patienten. Welcher Patient ist im Startkontext angegeben.
- Bereiche auf Benutzerebene gewähren Zugriff auf bestimmte Daten, auf die ein Benutzer zugreifen kann.
- Bereiche auf Systemebene gewähren Lese-/Schreibzugriff auf alle FHIR im Datenspeicher gefundenen Ressourcen. HealthLake

Die folgende Tabelle zeigt die Syntax für die Erstellung FHIR ressourcenbezogener Bereiche, die von unterstützt werden. HealthLake Das allgemeine Format lautet wie folgt:

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

Unterstützte Autorisierungsbereiche für HealthLake Datenspeicher

Bereichssyntax	Beispiel für einen Geltungsbereich	Ergebnis
`patient/(fhir-resource | '*'). ('read' | 'write' | '*')`	`patient/A llergyInt olerance.*`	Eine Client-An wendung hätte Lese-/ Schreibzugriff auf Allergien.
`user/(fhir-resource | '*').('re ad' | 'write' | '*')`	`user/Obse rvation.read`	Eine Client-An wendung hätte Lesezugriff auf alle

Bereichssyntax	Beispiel für einen Geltungsbereich	Ergebnis
		aufgezeichneten Beobachtungen.
`system/('read' | 'write' | *)`	`system/*.*`	Eine Client-An wendung hätte Lese-/ Schreibzugriff auf alle Daten.

Wird AWS Lambda für die Tokenvalidierung mit einem SMART FHIR eingeschalteten HealthLake Datenspeicher verwendet

Wenn Sie einen SMART HealthLake Datenspeicher erstellen, der nicht FHIR aktiviert ist, müssen Sie die AWS Lambda Funktion in der `CreateFHIRDatastore` Anfrage angeben. ARN Die Lambda-Funktion ARN wird im `IdentityProviderConfiguration` Objekt mithilfe des `IdpLambdaArn` Parameters angegeben.

Sie müssen die Lambda-Funktion erstellen, bevor Sie Ihren eigenen SMART FHIR aktivierten HealthLake Datenspeicher erstellen. Sobald Sie den Datenspeicher erstellt haben, kann das Lambda ARN nicht mehr geändert werden. Verwenden Sie den `DescribeFHIRDatastore` API Vorgang, um das Lambda zu sehen, das ARN Sie bei der Erstellung des Datenspeichers angegeben haben.

Damit eine FHIR REST Anfrage in einem nicht SMART FHIR aktivierten HealthLake Datenspeicher erfolgreich ist, muss Ihre Lambda-Funktion Folgendes tun:

- Die Lambda-Funktion muss in weniger als 1 Sekunde eine Antwort an den HealthLake Datenspeicher-Endpunkt zurückgeben.
- Dekodieren Sie das Zugriffstoken, das im Autorisierungsheader der von der REST API Client-Anwendung gesendeten Anfrage bereitgestellt wird.
- Weisen Sie eine IAM Servicerolle zu, die über ausreichende Berechtigungen verfügt, um die FHIR REST API Anfrage auszuführen.
- Die folgenden Ansprüche sind erforderlich, um eine FHIR REST API Anfrage abzuschließen. Weitere Informationen hierzu finden Sie unter Erforderliche Ansprüche.
  - `nbf`

- `exp`
- `isAuthorized`
- `aud`
- `scope`

Wenn Sie mit Lambda arbeiten, müssen Sie zusätzlich zu Ihrer Lambda-Funktion eine Ausführungsrolle und eine ressourcenbasierte Richtlinie erstellen. Die Ausführungsrolle einer Lambda-Funktion ist eine IAM Rolle, die der Funktion die Erlaubnis erteilt, auf AWS Dienste und Ressourcen zuzugreifen, die zur Laufzeit benötigt werden. Die von Ihnen bereitgestellte ressourcenbasierte Richtlinie muss es ermöglichen HealthLake , Ihre Funktion in Ihrem Namen aufzurufen.

In den Abschnitten dieses Themas werden eine Beispielanforderung von einer Client-Anwendung und eine dekodierte Antwort, die zum Erstellen einer AWS Lambda-Funktion erforderlichen Schritte und das Erstellen einer ressourcenbasierten Richtlinie beschrieben, die Folgendes voraussetzen kann. HealthLake

- Teil 1: Eine Lambda-Funktion erstellen
- Teil 2: Erstellen einer HealthLake Servicerolle, die von der AWS Lambda-Funktion verwendet wird
- Teil 3: Aktualisierung der Ausführungsrolle der Lambda-Funktion
- Teil 4: Hinzufügen einer Ressourcenrichtlinie zu Ihrer Lambda-Funktion
- Teil 5: Bereitstellung von Parallelität für Ihre Lambda-Funktion

Eine AWS Lambda-Funktion erstellen

Die in diesem Thema erstellte Lambda-Funktion wird ausgelöst, wenn Anfragen an einen HealthLake Datenspeicher HealthLake empfangen werden, SMART der FHIR aktiviert ist. Die Anfrage von der Client-Anwendung enthält einen REST API Aufruf und einen Autorisierungsheader, der ein Zugriffstoken enthält.

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

Die Lambda-Beispielfunktion in diesem Thema verwendet AWS Secrets Manager , um Anmeldeinformationen für den Autorisierungsserver zu verschleiern. Wir empfehlen dringend, Anmeldedaten für den Autorisierungsserver nicht direkt in einer Lambda-Funktion anzugeben.

Example Validierung einer FHIR REST Anfrage, die ein Autorisierungsträgertoken enthält

Die Lambda-Beispielfunktion zeigt Ihnen, wie Sie eine FHIR REST Anfrage validieren, die SMART an einen eingeschalteten FHIR HealthLake Datenspeicher gesendet wurde. step-by-stepsAnweisungen zur Implementierung dieser Lambda-Funktion finden Sie unterErstellen einer Lambda-Funktion mit dem AWS Management Console.

Wenn die FHIR REST API Anfrage keinen gültigen Datenspeicher-Endpunkt, kein Zugriffstoken und keinen gültigen REST Vorgang enthält, schlägt die Lambda-Funktion fehl. Weitere Informationen zu den erforderlichen Autorisierungsserverelementen finden Sie unterErforderliche Ansprüche.

```
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()
```

Erstellen einer Lambda-Funktion mit dem AWS Management Console

Bei diesem Verfahren wird davon ausgegangen, dass Sie die Servicerolle, die Sie bei der Bearbeitung einer FHIR REST API Anfrage in einem HealthLake Datenspeicher übernehmen möchten HealthLake , bereits erstellt FHIR haben. SMART Wenn Sie die Servicerolle nicht erstellt haben, können Sie trotzdem die Lambda-Funktion erstellen. Sie müssen die Dienstrolle ARN of hinzufügen, bevor die Lambda-Funktion funktioniert. Weitere Informationen zum Erstellen einer Servicerolle und deren Angabe in der Lambda-Funktion finden Sie unter Erstellen einer HealthLake Servicerolle zur Verwendung in der AWS Lambda-Funktion zur Dekodierung eines JWT

Um eine Lambda-Funktion ()AWS Management Console zu erstellen

1. Öffnen Sie die Seite Funktionen der Lambda-Konsole.
2. Wählen Sie Funktion erstellen aus.

3. Wählen Sie Verfassen von Grund auf aus.
4. Geben Sie unter Basisinformationen einen Funktionsnamen ein. Wählen Sie unter Runtime eine Python-basierte Runtime aus.
5. Wählen Sie für Execution role (Ausführungsrolle) die Option Create a new role with basic Lambda permissions (Neue Rolle mit grundlegenden Lambda-Berechtigungen erstellen) aus.

   Lambda erstellt eine Ausführungsrolle, die der Funktion die Berechtigung zum Hochladen von Protokollen auf Amazon CloudWatch erteilt. Die Lambda-Funktion übernimmt die Ausführungsrolle, wenn Sie Ihre Funktion aufrufen, und verwendet die Ausführungsrolle, um Anmeldeinformationen für zu erstellen. AWS SDK
6. Wählen Sie die Registerkarte Code und fügen Sie die Lambda-Beispielfunktion hinzu.

   Wenn Sie die Servicerolle für die zu verwendende Lambda-Funktion noch nicht erstellt haben, müssen Sie sie erstellen, bevor die Lambda-Beispielfunktion funktioniert. Weitere Informationen zum Erstellen einer Servicerolle für die Lambda-Funktion finden Sie unterErstellen einer HealthLake Servicerolle zur Verwendung in der AWS Lambda-Funktion zur Dekodierung eines 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()
```

Die Ausführungsrolle einer Lambda-Funktion ändern

Nachdem Sie die Lambda-Funktion erstellt haben, müssen Sie die Ausführungsrolle aktualisieren, sodass sie die erforderlichen Berechtigungen zum Aufrufen von Secrets Manager enthält. In Secrets

Manager hat jedes Secret, das Sie erstellen, eineARN. Um die geringste Berechtigung anzuwenden, sollte die Ausführungsrolle nur Zugriff auf die Ressourcen haben, die für die Ausführung der Lambda-Funktion benötigt werden.

Sie können die Ausführungsrolle einer Lambda-Funktion ändern, indem Sie in der IAM Konsole danach suchen oder in der Lambda-Konsole Konfiguration auswählen. Weitere Informationen zur Verwaltung Ihrer Ausführungsrolle für Lambda-Funktionen finden Sie unterLambda-Ausführungsrolle.

Example Lambda-Funktionsausführungsrolle, die Zugriff gewährt auf **`GetSecretValue`**

Durch das Hinzufügen der IAM Aktion `GetSecretValue` zur Ausführungsrolle wird die erforderliche Berechtigung erteilt, damit die Lambda-Beispielfunktion funktioniert.

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

Zu diesem Zeitpunkt haben Sie eine Lambda-Funktion erstellt, mit der Sie das Zugriffstoken validieren können, das als Teil der FHIR REST Anfrage SMART an Ihren eigenen FHIR aktivierten HealthLake Datenspeicher gesendet wurde.

Erstellen einer HealthLake Servicerolle zur Verwendung in der AWS Lambda-Funktion zur Dekodierung eines JWT

Persona: Administrator IAM

Ein Benutzer, der IAM Richtlinien hinzufügen oder entfernen und neue IAM Identitäten erstellen kann.

Servicerolle

Eine Servicerolle ist eine IAMRolle, die ein Dienst übernimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM

erstellen, ändern und löschen. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter Erstellen einer Rolle zum Delegieren von Berechtigungen AWS-Service an eine.

Nachdem das JSON Web Token (JWT) dekodiert wurde, benötigt Lambda die Autorisierung, um auch eine IAM Rolle zurückzugeben. ARN Diese Rolle muss über die erforderlichen Berechtigungen verfügen, um die REST API Anfrage auszuführen. Andernfalls schlägt sie aufgrund unzureichender Berechtigungen fehl.

Wenn Sie eine benutzerdefinierte Richtlinie einrichten, empfiehlt IAM es sich, die erforderlichen Mindestberechtigungen zu gewähren. Weitere Informationen finden Sie im Benutzerhandbuch unter Anwenden von Berechtigungen mit den IAM geringsten Rechten.

Das Erstellen einer HealthLake Servicerolle, die in der Autorisierungs-Lambda-Funktion festgelegt werden soll, erfordert zwei Schritte.

- Zunächst müssen Sie eine Richtlinie erstellenIAM. Die Richtlinie muss den Zugriff auf die FHIR Ressourcen spezifizieren, für die Sie Bereiche auf dem Autorisierungsserver bereitgestellt haben.
- Zweitens müssen Sie die Servicerolle erstellen. Wenn Sie die Rolle erstellen, legen Sie eine Vertrauensbeziehung fest und fügen die Richtlinie hinzu, die Sie in Schritt eins erstellt haben. Die Vertrauensstellung wird als HealthLake Dienstprinzipal bezeichnet. In diesem Schritt müssen Sie einen HealthLake Datenspeicher ARN und eine AWS Konto-ID angeben.

Eine neue IAM Richtlinie erstellen

Die Bereiche, die Sie auf Ihrem Autorisierungsserver definieren, bestimmen, auf welche FHIR Ressourcen ein authentifizierter Benutzer in einem HealthLake Datenspeicher zugreifen kann.

Die IAM Richtlinie, die Sie erstellen, kann an die von Ihnen definierten Bereiche angepasst werden.

Die folgenden Aktionen können im `Action` Element einer IAM Richtlinienerklärung definiert werden. Für jede `Action` in der Tabelle können Sie eine definieren`Resource types`. In HealthLake einem Datenspeicher befindet sich der einzige unterstützte Ressourcentyp, der im `Resource` Element einer IAM Berechtigungsrichtlinie definiert werden kann.

Einzelne FHIR Ressourcen sind keine Ressource, die Sie als Element in einer IAM Berechtigungsrichtlinie definieren können.

Aktionen, definiert von HealthLake

Aktionen	Beschreibung	Zugrif bene	Ressourcentyp (erforderlich)
CreateRes ource	Erteilt die Erlaubnis, eine Ressource zu erstellen	Schre	DatenspeicherARN: arn:aws:healthlake : ::datastore/fhir/ **your-region 111122223 333 your-datastore-id**
DeleteRes ource	Gewährt die Berechtig ung zum Löschen von Ressourcen	Schre	DatenspeicherARN: arn:aws:healthlake: **your- region** ::datastore/fhir/ **111122223333 your-datastore-id**
ReadResou rce	Gewährt die Berechtig ung zum Lesen von Ressourcen	Leser	DatenspeicherARN: arn:aws:healthlake: **your- region** ::datastore/fhir/ **111122223333 your-datastore-id**
SearchWit hGet	Erteilt die Erlaubnis , Ressourcen mit der Methode zu durchsuchen GET	Leser	DatenspeicherARN: arn:aws:healthlake : ::datastore/fhir/ **your-region 111122223 333 your-datastore-id**
SearchWit hPost	Erteilt die Erlaubnis , Ressourcen mit der Methode zu durchsuchen POST	Leser	DatenspeicherARN: arn:aws:healthlake : ::datastore/fhir/ **your-region 111122223 333 your-datastore-id**
S tartFHIRE xport JobWithPo st	Erteilt die Erlaubnis, einen FHIR Exportauftrag mit zu beginnen GET	Schre	DatenspeicherARN: arn:aws:healthlake : ::datastore/fhir/ **your-region 111122223 333 your-datastore-id**
UpdateRes ource	Gewährt die Berechtigung zum Aktualisieren von Ressourcen	Schre	DatenspeicherARN: arn:aws:healthlake: **your- region** ::datastore/fhir/ **111122223333 your-datastore-id**

Um zu beginnen, können Sie verwenden. `AmazonHealthLakeFullAccess` Diese Richtlinie würde Lese-, Schreib-, Such- und Exportvorgänge für alle FHIR Ressourcen in einem Datenspeicher zulassen. Um nur Leseberechtigungen für die Verwendung eines Datenspeichers zu gewähren. `AmazonHealthLakeReadOnlyAccess`

Weitere Informationen zum Erstellen einer benutzerdefinierten Richtlinie mithilfe von AWS Management Console AWS CLI IAMSDKs, oder finden Sie unter Erstellen von IAM Richtlinien im IAMBenutzerhandbuch.

Eine Servicerolle für HealthLake (IAMKonsole) erstellen

Gehen Sie wie folgt vor, um eine Servicerolle zu erstellen. Wenn Sie einen Dienst erstellen, müssen Sie auch eine IAM Richtlinie festlegen.

Um die Servicerolle für HealthLake (IAMKonsole) zu erstellen

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die IAM Konsole unter https://console.aws.amazon.com/iam/.
2. Wählen Sie im Navigationsbereich der IAM-Konsole Roles (Rollen) aus.
3. Wählen Sie anschließend Create role (Rolle erstellen) aus.
4. Wählen Sie auf der Seite Vertrauensentität auswählen die Option Benutzerdefinierte Vertrauensrichtlinie aus.
5. Aktualisieren Sie anschließend unter Benutzerdefinierte Vertrauensrichtlinie die Beispielrichtlinie wie folgt. **`your-account-id`**Ersetzen Sie sie durch Ihre Kontonummer und fügen Sie die Nummer ARN des Datenspeichers hinzu, den Sie für Ihre Import- oder Exportaufträge verwenden möchten.

```
{
  "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. Wählen Sie anschließend Weiter.
7. Wählen Sie auf der Seite „Berechtigungen hinzufügen“ die Richtlinie aus, die für den HealthLake Dienst gelten soll. Um Ihre Richtlinie zu finden, suchen Sie unter Berechtigungsrichtlinien danach.
8. Wählen Sie dann Richtlinie anhängen aus.
9. Geben Sie dann auf der Seite Name, Überprüfung und Erstellung unter Rollenname einen Namen ein.
10. (Optional) Fügen Sie anschließend unter Beschreibung eine kurze Beschreibung für Ihre Rolle hinzu.
11. Geben Sie möglichst einen Rollennamen oder ein Rollennamen-Suffix ein, mit dem der Zweck dieser Rolle einfach zu erkennen ist. Rollennamen müssen innerhalb Ihrer eindeutig sein AWS-Konto. Es wird hierbei nicht zwischen Groß- und Kleinschreibung unterschieden. z. B. können Sie keine Rollen erstellen, die **`PRODROLE`** bzw. **`prodrole`** heißen. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname nach der Erstellung nicht bearbeitet werden.
12. Überprüfen Sie die Rollendetails und wählen Sie dann Rolle erstellen aus.

Informationen zur Angabe der Rolle ARN in der Lambda-Beispielfunktion finden Sie unterEine AWS Lambda-Funktion erstellen.

Lambda-Ausführungsrolle

Die Ausführungsrolle einer Lambda-Funktion ist eine IAM Rolle, die der Funktion die Erlaubnis erteilt, auf AWS Dienste und Ressourcen zuzugreifen. Diese Seite enthält Informationen zum Erstellen, Anzeigen und Verwalten der Ausführungsrolle einer Lambda-Funktion.

Standardmäßig erstellt Lambda eine Ausführungsrolle mit minimalen Berechtigungen, wenn Sie eine neue Lambda-Funktion mit dem erstellen. AWS Management Console Informationen zur

Verwaltung der in der Ausführungsrolle gewährten Berechtigungen finden Sie unter Erstellen einer Ausführungsrolle in der IAM Konsole im Lambda Developer Guide.

Die in diesem Thema vorgestellte Lambda-Beispielfunktion verwendet Secrets Manager, um die Anmeldeinformationen des Autorisierungsservers zu verschleiern.

Wie bei jeder IAM Rolle, die Sie erstellen, ist es wichtig, die bewährte Methode mit den geringsten Rechten zu befolgen. Während der Entwicklungsphase können Sie manchmal Berechtigungen gewähren, die über das hinausgehen, was erforderlich ist. Bevor Sie Ihre Funktion in der Produktionsumgebung veröffentlichen, sollten Sie als bewährte Methode die Richtlinie so anpassen, dass sie nur die erforderlichen Berechtigungen enthält. Weitere Informationen finden Sie unter Anwenden der geringsten Rechte im Benutzerhandbuch. IAM

Erlaube HealthLake , deine Lambda-Funktion auszulösen

Um die Lambda-Funktion in Ihrem Namen aufrufen zu HealthLake können, müssen Sie Folgendes tun:

- Sie müssen den `IdpLambdaArn` Wert der Lambda-Funktion, die Sie in ARN der Anfrage aufrufen HealthLake möchten, auf den `CreateFHIRDatastore` Wert setzen.
- Sie benötigen eine ressourcenbasierte Richtlinie, die es Ihnen ermöglicht HealthLake , die Lambda-Funktion in Ihrem Namen aufzurufen.

Wenn HealthLake er eine FHIR REST API Anfrage für einen HealthLake Datenspeicher empfängt, SMART der FHIR aktiviert ist, benötigt er Berechtigungen, um die Lambda-Funktion aufzurufen, die bei der Erstellung des Datenspeichers in Ihrem Namen angegeben wurde. Um HealthLake Zugriff zu gewähren, verwenden Sie eine ressourcenbasierte Richtlinie. Weitere Informationen zum Erstellen einer ressourcenbasierten Richtlinie für eine Lambda-Funktion finden Sie unter Zulassen, dass ein AWS Dienst eine Lambda-Funktion aufruft im Entwicklerhandbuch.AWS Lambda

Bereitstellung von Parallelität für Ihre Lambda-Funktion

> **Important**
>
> HealthLake erfordert, dass die maximale Laufzeit für Ihre Lambda-Funktion weniger als eine Sekunde (1000 Millisekunden) beträgt.
> Wenn Ihre Lambda-Funktion das Laufzeitlimit überschreitet, erhalten Sie eine TimeOutAusnahme.

Um zu vermeiden, dass diese Ausnahme auftritt, empfehlen wir, die bereitgestellte Parallelität zu konfigurieren. Wenn Sie Provisioned Concurrency (Bereitgestellte Gleichzeitigkeit) vor einer Erhöhung der Aufrufe zuweisen, können Sie sicherstellen, dass alle Anforderungen von initialisierten Instances mit geringer Latenz verarbeitet werden. Weitere Informationen zur Konfiguration der bereitgestellten Parallelität finden Sie unter Konfiguration der bereitgestellten Parallelität im Lambda Developer Guide

Um die durchschnittliche Laufzeit Ihrer Lambda-Funktion aktuell zu sehen, verwenden Sie die Monitoring-Seite für Ihre Lambda-Funktion in der Lambda-Konsole. Standardmäßig bietet die Lambda-Konsole ein Dauer-Diagramm, das Ihnen die durchschnittliche, minimale und maximale Zeit anzeigt, die Ihr Funktionscode mit der Verarbeitung eines Ereignisses verbringt. Weitere Informationen zur Überwachung von Lambda-Funktionen finden Sie unter Überwachungsfunktionen in der Lambda-Konsole im Lambda Developer Guide.

Wenn Sie bereits Parallelität für Ihre Lambda-Funktion bereitgestellt haben und diese überwachen möchten, finden Sie weitere Informationen unter Überwachung der Parallelität im Lambda Developer Guide.

Einen SMART HealthLake Datenspeicher mit FHIR aktivierter Aktivierung erstellen

Um das SMART FHIR On-Framework mit zu verwenden HealthLake, erstellen Sie einen HealthLake Datenspeicher mit dem in Ihrer reateFHIRDatastore C-Anfrage angegebenen `IdentityProviderConfiguration` Parameter. Im `IdentityProviderConfiguration` Parameter geben Sie die folgenden Informationen an:

- Stellen Sie den AuthorizationStrategyWert gleich ein`SMART_ON_FHIR_V1`.
- Stellen Sie den IdpLambdaArnWert auf den ARN Wert ein, den AWS Lambda Sie erstellt haben, um die Token-Dekodierung mit Ihrem Autorisierungsserver zu verwalten.
- Definieren Sie die auf dem Autorisierungsserver angegebenen Metadatenelemente als JSON Block. Diese Metadatenelemente werden im Discovery-Dokument zurückgegeben.
- Optional: Aktivieren FineGrainedAuthorizationEnabled. Geben Sie `True` an, dass Sie die detaillierte Autorisierung verwenden möchten, die bereitgestellt wird von HealthLake

Sie können mit AWS Command Line Interface (AWS CLI) oder SMART über einen der AWS unterstützten SDKs Datenspeicher einen FHIR aktivierten Datenspeicher erstellen. Das Erstellen

eines SMART HealthLake Datenspeichers, der FHIR aktiviert ist, wird über die HealthLake Konsole nicht unterstützt.

Verwenden von AWS CLI zum Erstellen eines SMART HealthLake Datenspeichers mit FHIR aktiviertem

Sie können das folgende Codebeispiel verwenden, um einen SMART FHIR aktivierten HealthLake Datenspeicher mit dem zu erstellen AWS CLI. Beim Erstellen eines HealthLake Datenspeichers, SMART der FHIR aktiviert ist, müssen Sie den `identity-provider-configuration`Parameter angeben.

Im `identity-provider-configuration` Parameter können Sie optional eine differenzierte Autorisierung aktivieren, indem Sie den `FineGrainedAuthorizationEnabled` Wert gleich setzen. `True` Weitere Informationen zur feinkörnigen Autorisierung finden Sie unter. Verwendung einer detaillierten Autorisierung mit einem SMART FHIR aktivierten Datenspeicher HealthLake Das folgende Beispiel enthält ein Sonderzeichen \ zur Kennzeichnung von Zeilenumbrüchen oder als Escape-Zeichen. Dies dient der Übersichtlichkeit.

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

Bei Erfolg erhalten Sie die folgende JSON Antwort:

```
{
  "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"
}
```

Verwendung einer detaillierten Autorisierung mit einem SMART FHIR aktivierten Datenspeicher HealthLake

Bereiche allein bieten Ihnen nicht die nötige Genauigkeit darüber, auf welche Daten ein Anforderer in einem Datenspeicher zugreifen darf. Die Verwendung einer differenzierten Autorisierung ermöglicht ein höheres Maß an Spezifität bei der Gewährung des Zugriffs auf einen Datenspeicher, der nicht aktiviert ist. SMART FHIR HealthLake Um eine differenzierte Autorisierung zu verwenden, legen Sie `True` im `IdentityProviderConfiguration` Parameter Ihrer C-Anfrage den Wert „`FineGrainedAuthorizationEnabled`equal to“ fest. reateFHIRDatastore

Wenn Sie die differenzierte Autorisierung aktiviert haben, gibt Ihr Autorisierungsserver `id_token` zusammen mit dem Zugriffstoken einen `fhirUser` Bereich zurück. Dadurch können Informationen über den Benutzer von der Client-Anwendung abgerufen werden. Die Client-Anwendung sollte den `fhirUser` Anspruch als den URI einer FHIR Ressource behandeln, die den aktuellen Benutzer repräsentiert. Mögliche Werte sind `Patient`, `Practitioner` oder `RelatedPerson`. Die Antwort des Autorisierungsservers beinhaltet auch einen `user/` Bereich, der definiert, auf welche Daten der Benutzer zugreifen kann. Dabei wird die Syntax verwendet, die für Bereiche definiert ist, die sich auf FHIR ressourcenspezifische Bereiche beziehen:

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

Im Folgenden finden Sie Beispiele dafür, wie eine differenzierte Autorisierung verwendet werden kann, um Ressourcentypen im Zusammenhang mit dem Datenzugriff weiter zu spezifizieren. FHIR

- Wann bestimmt `fhirUser` eine differenzierte Autorisierung`Practitioner`, auf welche Patientengruppe der Benutzer zugreifen kann. Der Zugang `fhirUser` ist nur für Patienten erlaubt, auf die sich der Patient `fhirUser` als Allgemeinmediziner bezieht.

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

- Wann `fhirUser` ist ein `Patient` oder `RelatedPerson` und der Patient, auf den in der Anfrage verwiesen wird, unterscheidet sich von der Genehmigung`fhirUser`, zu der der angefragte Patient Zugang `fhirUser` hat. Der Zugriff ist zulässig, wenn in der angeforderten `Patient` Ressource eine Beziehung angegeben ist.

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

Abrufen des SMART Discovery-Dokuments eines HealthLake Datenspeichers, der FHIR aktiviert ist

Damit eine Client-Anwendung eine erfolgreiche FHIR REST Anfrage stellen kann, muss sie die im HealthLake Datenspeicher definierten Autorisierungsanforderungen erfassen. Für den Erfolg dieser Anfrage ist keine Autorisierung (Bearer-Token) erforderlich.

Stellen Sie dazu eine `GET` Anfrage und fügen Sie sie `/.well-known/smart-configuration` an den Endpunkt des Datenspeichers an

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

Dadurch wird das Discovery-Dokument des HealthLake Datenspeichers als JSON Blob zurückgegeben. Darin finden Sie die `authorization_endpoint` und die `token_endpoint` im HealthLake Datenspeicher definierten Spezifikationen und Funktionen.

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

URLswird benötigt, um eine Client-Anwendung erfolgreich zu starten

- Autorisierungsendpunkt: Wird URL benötigt, um eine Client-Anwendung oder einen Benutzer zu autorisieren.
- Token-Endpunkt: Der Endpunkt des Autorisierungsservers, über den die Client-Anwendung mit ihr kommuniziert.

Eine FHIR REST API Anfrage für einen SMART aktivierten HealthLake Datenspeicher stellen

Sie können FHIR REST API Anfragen für einen SMART HealthLake Datenspeicher mit FHIR aktivierter Option stellen. Das folgende Beispiel zeigt eine Anfrage von einer Client-Anwendung, die ein JWT im Autorisierungsheader enthält, und wie Lambda die Antwort dekodieren sollte. Nachdem die Anforderung der Client-Anwendung autorisiert und authentifiziert wurde, muss sie ein Bearer-Token vom Autorisierungsserver erhalten. Verwenden Sie das Bearer-Token im Autorisierungsheader, wenn Sie eine FHIR REST API Anfrage an einen Datenspeicher senden, SMART der aktiviert FHIR ist. HealthLake

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

Da im Autorisierungsheader ein Bearer-Token gefunden wurde und keine AWS IAM Identität erkannt wurde, wird die Lambda-Funktion HealthLake aufgerufen, die bei der Erstellung des SMART FHIR HealthLake On-Enabled-Datenspeichers angegeben wurde. Wenn das Token erfolgreich von Ihrer Lambda-Funktion dekodiert wurde, finden Sie hier eine Beispielantwort, die an gesendet wurde. 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
}
```

Einrichtung der Ressourcen, die für die Implementierung eines SMART FHIR Datenspeichers erforderlich sind

In diesem Thema werden die Ressourcen beschrieben, die Sie in Ihrem AWS Konto extern bereitstellen müssen HealthLake, wie Sie einen SMART HealthLake Datenspeicher mit FHIR aktivierter Aktivierung einrichten und wie eine SMART FHIR Client-Anwendung mit einem Autorisierungsserver und einem HealthLake Datenspeicher interagieren würde.

Die Schritte in diesem Workflow definieren die grundlegenden Schritte für die Bearbeitung von FHIR Anfragen und die Ressourcen, die für deren erfolgreiche Bearbeitung erforderlich sind. SMART

Bei einem SMART FHIR On-Request-Prozess arbeiten drei Anwendungen zusammen:

- Der Endbenutzer: Im Allgemeinen ein Patient oder ein Arzt, der eine Drittanbieter-App verwendet, um SMART auf FHIR Daten in einem HealthLake Datenspeicher zuzugreifen.
- Die SMART FHIR On-Applikation (als Client-Anwendung bezeichnet): Eine Anwendung, die auf Daten zugreifen möchte, die sich im HealthLake Datenspeicher befinden.
- Der Autorisierungsserver: Ein OpenID Connect-kompatibler Server, der Benutzer authentifizieren und Zugriffstoken ausstellen kann.
- Der HealthLake Datenspeicher: Ein SMART eingeschalteter FHIR HealthLake Datenspeicher, der eine Lambda-Funktion verwendet, um auf FHIR REST Anfragen zu antworten, die ein Bearer-Token bereitstellen.

Damit diese Anwendungen zusammenarbeiten können, müssen Sie die folgenden Ressourcen erstellen.

Wir empfehlen, den SMART HealthLake Datenspeicher mit FHIR aktivierter Aktivierung zu erstellen, nachdem Sie den Autorisierungsserver eingerichtet, die erforderlichen Bereiche für ihn definiert und eine AWS Lambda Funktion für die Token-Introspektion erstellt haben.

1. Einrichtung eines Autorisierungsserver-Endpunkts — Autorisierungsserver

Um das SMART FHIR On-Framework verwenden zu können, müssen Sie einen Autorisierungsserver eines Drittanbieters einrichten, der FHIR REST Anfragen an einen Datenspeicher validieren kann. Weitere Informationen zum Einrichten eines Autorisierungsserver-Endpunkts, mit dem Sie arbeiten können HealthLake, finden Sie unterAuthentifizierungsanforderungen SMART für FHIR.

2. Definieren Sie Bereiche, um zu kontrollieren, wer auf welche Daten in Ihrem HealthLake Datenspeicher auf Ihrem Autorisierungsserver zugreifen kann — Autorisierungsserver

Das SMART FHIR On-Framework verwendet OAuth Bereiche, um zu bestimmen, auf welche FHIR Ressourcen eine authentifizierte Anfrage Zugriff hat und in welchem Umfang. Die Definition von Bereichen ist eine Möglichkeit, Designs so zu gestalten, dass sie die geringsten Rechte haben. Weitere Informationen zu Bereichen, die vom FHIR On-Framework definiert und von diesem unterstützt werden, finden SMART Sie unter. HealthLake Unterstützt SMART auf FHIR OAuth Bereichen von HealthLake

3. Richten Sie eine AWS Lambda Funktion ein, die Token-Introspektion durchführen kann — Ihr Konto AWS

Eine FHIR REST Anfrage, die von der Client-Anwendung SMART an einen Datenspeicher gesendet wird, der nicht FHIR aktiviert ist, enthält ein JSON Web-Token (). JWT Weitere Informationen zum Einrichten einer Lambda-Funktion, die dekodieren und validieren kann, finden Sie unter Decoding a. JWT

4. Erstellen Sie einen HealthLake Datenspeicher, der nicht SMART FHIR aktiviert ist — Ihr Konto AWS

Um einen SMART FHIR HealthLake On-Datenspeicher zu erstellen, müssen Sie einen angeben`IdentityProviderConfiguration`. Weitere Informationen zu den erforderlichen `IdentityProviderConfiguration` Parametern in einer reateFHIRDatastore C-Anforderung finden Sie unter Einen HealthLake Datenspeicher mit SMART FHIR eingeschaltetem Speicher erstellen.

So startet eine Client-Anwendung und fordert Daten von einem HealthLake Datenspeicher SMART an, der FHIR aktiviert ist

In diesem Abschnitt wird erklärt, wie eine Client-Anwendung SMART im FHIR On-Kontext gestartet wird und erfolgreich eine FHIR REST Anfrage an einen HealthLake Datenspeicher stellen kann.

1. Die Client-Anwendung stellt eine **`GET`** Anfrage an den Known Uniform Resource Identifier

Eine SMART aktivierte Client-Anwendung muss eine `GET` Anfrage stellen, um die Autorisierungsendpunkte Ihres HealthLake Datenspeichers zu finden. Dies erfolgt über eine Known Uniform Resource Identifier (URI) -Anforderung. Weitere Informationen dazu finden Sie unter Discovery-Dokument eines SMART HealthLake Datenspeichers abrufen, der nicht FHIR aktiviert ist.

2. Zugriff und Geltungsbereiche anfordern

Die Client-Anwendung verwendet den Autorisierungsendpunkt des Autorisierungsservers, sodass sich der Benutzer anmelden kann. Dieser Prozess authentifiziert den Benutzer. Bereiche werden verwendet, um zu definieren, auf welche FHIR Ressourcen in Ihrem HealthLake Datenspeicher eine Client-Anwendung zugreifen kann. Weitere Informationen zum Definieren von Bereichen finden Sie unter. Unterstützt SMART auf FHIR OAuth Bereichen von HealthLake

3. Zugriffstoken

Nachdem der Benutzer nun authentifiziert wurde, erhält eine Client-Anwendung ein JWT Zugriffstoken vom Autorisierungsserver. Dieses Token wird bereitgestellt, wenn die Client-Anwendung eine FHIR REST Anfrage an HealthLake sendet. Weitere Informationen darüber, wie das mit einer Lambda-Funktion dekodiert JWT wird, finden Sie unter. Token-Validierung durchführen

4. Eine FHIR REST Anfrage für einen SMART FHIR HealthLake aktivierten Datenspeicher stellen

Jetzt kann die Client-Anwendung mithilfe des vom Autorisierungsserver bereitgestellten Zugriffstoken eine FHIR REST Anfrage an einen HealthLake Datenspeicher-Endpunkt senden. Ein Beispiel für eine FHIR REST Anfrage finden Sie unterEine FHIR REST API Anfrage für einen SMART aktivierten HealthLake Datenspeicher stellen.

5. Das JWT Zugriffstoken wird validiert

Verwenden Sie eine Lambda-Funktion, um das in der FHIR REST Anfrage gesendete Zugriffstoken zu validieren. Informationen zum Erstellen einer Lambda-Funktion, die eine Token-Introspektion durchführen kann, finden Sie unter. Eine AWS Lambda-Funktion erstellen

Verwenden der automatisierten Ressourcengenerierung auf der Grundlage der Verarbeitung natürlicher Sprache (NLP) des FHIR DocumentReference Ressourcentyps in AWS HealthLake

> **Note**
>
> Nach dem 20. Februar 2023 verwenden HealthLake Datenspeicher standardmäßig keine integrierte Verarbeitung natürlicher Sprache (NLP). Wenn Sie daran interessiert sind, diese Funktion in Ihrem Datenspeicher zu aktivieren, finden Sie Wie aktiviere ich die HealthLake integrierte Funktion zur Verarbeitung natürlicher Sprache? weitere Informationen im Kapitel Fehlerbehebung.
> Wenn Sie die Integration von Amazon Comprehend Medical aktiviert habenNLP, fallen beim Erstellen oder Aktualisieren von `DocumentReference` Ressourcen Gebühren auf Ihrem Konto an. AWS Weitere Informationen finden Sie unter Preise.AWS HealthLake
> Amazon Comprehend Medical ist im asiatisch-pazifischen Raum (Mumbai) nicht verfügbar. HealthLake Datenspeicher, die in der Region Asien-Pazifik (Mumbai) erstellt wurden, unterstützen keine integrierte Verarbeitung natürlicher Sprache (NLP).

HealthLake bietet Ihnen automatisch integrierte Verarbeitung natürlicher Sprache (NLP) mithilfe von Amazon Comprehend Medical für die unstrukturierte Datenverarbeitung für Daten, die `DocumentReference` im Ressourcentyp gespeichert sind. HealthLake Rufen Sie dazu Amazon Comprehend Medical und `DetectEntities-V2` `InferRxNorm` API Operations `InferICD10-CM` an. Die Ergebnisse werden automatisch als Erweiterung an die `DocumentReference` Ressource angehängt. Wenn die API Abläufe von Amazon Comprehend Medical Merkmale erkennen, die, und sind `SIGNSYMPTOM`, wird `DIAGNOSIS` automatisch ein `Linkage` Ressourcentyp generiert. Neue Zustands- und Beobachtungsressourcen werden anhand von Entitäten erstellt, die mit den Merkmalen von `SIGNSYMPTOM`, oder identifiziert wurden`DIAGNOSIS`, und sie werden mit dieser Verbindungsressource mit dem Quelldokument verknüpft.

Für Ressourcen, die durch die Integration generiert wurdenNLP, können Sie `GET` Anfragen stellen, aber die Suche nach diesen neuen Ressourcen wird nicht unterstützt.

Weitere Informationen zur Suche nach diesen Erweiterungen mithilfe HealthLake der Integration mit Athena finden Sie unterFragen Sie Ihren HealthLake Datenspeicher ab mit SQL.

Inhalt

Wie Amazon Comprehend Medical integriert ist HealthLake

HealthLake leitet mithilfe von Amazon Comprehend Medical Daten ab, die im `DocumentReference` Ressourcentyp gefunden wurden. Amazon Comprehend Medical API Medical-Operationen *`DetectEntities-V2`* und *`InferRxNorm`* erkennt Erkrankungen als Merkmale. *`InferICD10-CM`* Jede Operation bietet unterschiedliche Erkenntnisse.

> ⚠ Sprach-Support
>
> Amazon Comprehend Medical API Operations erkennt nur medizinische Entitäten in englischsprachigen Texten.

- DetectEntities-V2: Prüft den klinischen Text auf eine Vielzahl von medizinischen Entitäten und gibt spezifische Informationen zu ihnen zurück, z. B. die Kategorie der Entität, den Standort und den Konfidenzwert.
- Infer ICD1 0-CM: Erkennt Erkrankungen in einer Patientenakte als Entitäten und verknüpft diese Entitäten mit normalisierten Konzeptkennungen in der ICD -10-CM-Wissensdatenbank des National Center for Health Statistics CDC des National Center for Health Statistics mit Genehmigung der Weltgesundheitsorganisation (). WHO
- InferRxNorm: Erkennt Medikamente als Entitäten, die in einer Patientenakte aufgeführt sind, und verknüpft sie mit den normalisierten Konzeptidentifikatoren in der Datenbank der RxNorm National Library of Medicine.

Die unterstützten Merkmale für jede API Operation sind `SIGNSYMPTOM`, und. `DIAGNOSIS` Wenn Merkmale erkannt werden, werden sie als FHIR -konforme Erweiterungen zu verschiedenen Speicherorten in Ihrem HealthLake Datenspeicher hinzugefügt.

Standorte, an denen Erweiterungen hinzugefügt werden.

- `DocumentReference`: Die Ergebnisse der Amazon Comprehend Medical API Medical-Operationen werden jedem Dokument hinzugefügt`extension`, das innerhalb des DocumentReference Ressourcentyps gefunden wird. Die Ergebnisse in der Erweiterung sind in zwei Gruppen unterteilt. Sie finden sie in den Ergebnissen, die auf ihren basieren`URL`.
  - `http://healthlake.amazonaws.com/system-generated-resources/`
    - Dies sind Ressourcentypen, die von erstellt oder hinzugefügt wurden HealthLake.
  - `http://healthlake.amazonaws.com/aws-cm/`
    - Wo die Rohausgabe der Amazon Comprehend Medical API Medical-Operationen zu Ihrem HealthLake Datenspeicher hinzugefügt wird.
- `Linkage`: Dieser Ressourcentyp wird entweder hinzugefügt oder als Ergebnis der Integration erstellt. NLP Eine `GET` Anfrage zu einer bestimmten Ressource `Linkage` gibt eine Liste verknüpfter Ressourcen zurück. Suchen Sie nach dem `Linkage` hinzugefügten `"tag": [{"display": "SYSTEM_GENERATED"}]` Schlüssel-Wert-Paar HealthLake, um festzustellen, ob a hinzugefügt wurde. Weitere Informationen zu den FHIR Spezifikationen für Linkage finden Sie unter Ressourcentyp: Linkage im Dokumentationsindex. FHIR
- FHIRRessourcentypen, die als Ergebnis der Operationen von Amazon Comprehend Medical API generiert wurden.
  - `Observation`: Es wurden Ergebnisse der Amazon Comprehend Medical API DetectEntities Medical-Operationen -V2 und Infer ICD1 0-CM hinzugefügt, wenn die Merkmale oder sind. `SIGN SYMPTOM`
  - `Condition`: Es wurden Ergebnisse der Amazon Comprehend Medical API DetectEntities Medical-Operationen -V2 und Infer ICD1 0-CM hinzugefügt, wenn die Merkmale vorhanden sind. `DIAGNOSIS`
  - `MedicationStatement`: Die Ergebnisse der API Operation Amazon Comprehend Medical InferRxNorm wurden hinzugefügt.

Integration in den Betrieb FHIR REST API

Standardmäßig werden Merkmale, die von den Amazon Comprehend Medical API Medical-Vorgängen erkannt wurden, bei einer `GET` Anfrage nicht zurückgegeben.

Um die Ergebnisse der integrierten NLP Operationen für diese Ressourcentypen zu sehen, müssen Sie einen bekannten Wert angeben. `ID`

- `Linkage`
- `Observation`
- `Condition`
- `MedicationStatement`

Die Ergebnisse der integrierten NLP Operationen außerhalb des DocumentReference Ressourcentyps sind nur mit einer `GET` Anfrage verfügbar, bei der bekannt `ID` ist, dass die angegebenen Ergebnisse aus den Vorgängen von Amazon Comprehend Medical API enthalten.

Beispiele dafür, wie die API Abläufe von Amazon Comprehend Medical integriert sind HealthLake

Beispiel 1: Patientenakte, die in einen Datenspeicher aufgenommen wurde HealthLake

Hier ist ein Beispiel für eine klinische Notiz, die auf der Begegnung eines Patienten mit einem Arzt basiert.

> ⚠ Synthetische Daten
>
> Der Text in diesem Beispiel ist synthetischer Inhalt und enthält keine persönlichen Gesundheitsinformationen (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)
```

Zur Erinnerung: Diese Informationen sind in der Ressource im Base64-Format codiert. DocumentReference Wenn dieses Dokument aufgenommen wurde HealthLake und die API Vorgänge von Amazon Comprehend Medical abgeschlossen sind, können Sie mit der `GET` Anfrage für den Ressourcentyp beginnen, um die Ergebnisse zu sehen. `DocumentReference`

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

Wenn die API Operationen von Amazon Comprehend Medical erfolgreich sind, suchen Sie in den folgenden Links nach diesen Schlüssel-Wert-Paaren `extension "url": "http://healthlake.amazonaws.com/aws-cm/"`

```
{
  "url": "http://healthlake.amazonaws.com/aws-cm/status/",
  "valueString": "SUCCESS"
 },
 {
  "url": "http://healthlake.amazonaws.com/aws-cm/message/",
  "valueString": "The Amazon HealthLake integrated medical NLP operation was
 successful."
 }
```

Die folgenden Registerkarten zeigen Ihnen, wie die aufgenommenen Patientenakten je nach Ressourcentyp in Ihrem HealthLake Datenspeicher gemeldet werden.

DocumentReference

Um die Ergebnisse für einen einzelnen `DocumentReference` Ressourcentyp zu sehen, stellen Sie eine `GET` Anfrage, in `id` der die Daten einer bestimmten Ressource bereitgestellt werden.

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

Bei Erfolg erhalten Sie einen `200` HTTP Antwortcode und die folgende JSON Antwort (die aus Gründen der Übersichtlichkeit gekürzt wurde).

Hier ist der `http://healthlake.amazonaws.com/system-generated-resources/` Teil. Sie können sehen, dass ein neuer hinzugefügt `Linkage/`*`e366d29f-2c22-4c19-866e-09603937935a`* wurde. Sie können auch sehen, wo auf HealthLake Inferenzen basierende Ergebnisse zu bestimmten `Observation` `Condition` Ressourcentypen hinzugefügt wurden.

Um zu sehen, wie diese Ressourcentypen geändert wurden, wählen Sie die entsprechenden Tabs.

```
{
  "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

Um die Ergebnisse für einen einzelnen `Linkage` Ressourcentyp zu sehen, stellen Sie eine `GET` Anfrage, in `ID` der die einer bestimmten Ressource angegeben ist.

```
GET https://https://healthlake.your-region.amazonaws.com/
datastore/your-datastore-id/r4/eeb8005725ae22b35b4edbdc68cf2dfd/r4/
Linkage/e366d29f-2c22-4c19-866e-09603937935a
```

Bei Erfolg erhalten Sie einen `200` HTTP Antwortcode und die folgende gekürzte JSON Antwort.

Die Antwort enthält das `item` Element. Darin `"type": "source"` gibt das Schlüssel-Wert-Paar den spezifischen `DocumentReference` Eintrag an, mit dem das Schlüssel-Wert-Paar geändert wurde `Condition` und der unter dem `"type": "alternate"` Schlüssel-Wert-Paar `Observations` aufgeführt ist.

Sie sehen auch das *`meta`* Element und ein entsprechendes Schlüssel-Wert-Paar, was darauf hinweist*`"tag": [{"display": "SYSTEM_GENERATED"}]`*, dass diese Ressourcen von erstellt wurden. 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

Um die Ergebnisse für einen einzelnen `Observation` Ressourcentyp zu sehen, stellen Sie eine `GET` Anfrage, in `ID` der die einer bestimmten Ressource bereitgestellt wird.

```
GET https://https://healthlake.your-region.amazonaws.com/
datastore/your-datastore-id/r4/eeb8005725ae22b35b4edbdc68cf2dfd/r4/
Observation/e366d29f-2c22-4c19-866e-09603937935a
```

Die Ergebnisse der API Geschäftstätigkeit von Amazon Comprehend Medical wurden um die folgenden Elemente ergänzt: `codemeta`, und. `modifierExtension`

`code`

Ein Element vom Typ. `CodeableConcept` Weitere Informationen finden Sie CodeableConceptim FHIRDokumentationsindex.

HealthLake hängt die folgenden drei Schlüssel-Wert-Paare an.

- `"system": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/"`: Wo URL sich das auf eine bestimmte Operation von Amazon Comprehend Medical API bezieht. In diesem Fall wird auf 0CM geschlossen. ICD1
- `"code": "A52.06"`: Wo `A52.06` ist der ICD -10-CM-Code, der das Konzept identifiziert, das in der Wissensdatenbank der Centers for Disease Control zu finden ist?
- `"display": "Other syphilitic heart involvement"`: Wo `"Other syphilitic heart involvement"` ist die lange Beschreibung des ICD -10-CM-Codes in der Ontologie.

Die folgende gekürzte JSON Antwort enthält nur das Element. `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"
 }
```

Verwenden Sie das Element, um zu ermitteln, ob das Modell sicher ist, dass der zugewiesene ICD -10-CM-Code korrekt ist. `modifierExtension`

`meta`

Das `meta` Element enthält Metadaten, die angeben, ob das `code` Element Details enthält, die durch den Betrieb von Amazon Comprehend Medical API hinzugefügt wurden.

Die folgende gekürzte JSON Antwort enthält nur das Element. `meta`

```
"meta": {
  "lastUpdated": "2022-10-21T19:38:30.879Z",
  "tag": [{
   "display": "SYSTEM_GENERATED"
  }]
 }
```

`modifierExtension`

Das `modifierExtension` Element enthält weitere Informationen zum Konfidenzniveau der zugewiesenen Codes, die im `code` Element gefunden wurden. Es enthält auch Schlüssel-Wert-Paare, die einen Link zurück zum Original, das zur Generierung der Ergebnisse DocumentReference verwendet wurde, und zum zugehörigen Linkage-Ressourcentyp bereitstellen.

Für jedes hinzugefügte `coding` Element wird ein `entity-score` und ein dem `entity-Concept-Score` hinzugefügten Element angezeigt. modifierExtension Für jeden Wert im Schlüssel-Wert-Paar wird eine Punktzahl angezeigt. Denn `entity-score` dieser Wert ist das Maß an Vertrauen, das Amazon Comprehend Medical in die Genauigkeit der Erkennung hat. Denn dieser Wert ist das Maß an Vertrauen`entity-Concept-Score`, das Amazon Comprehend Medical davon hat, dass das Unternehmen korrekt mit einem ICD -10-CM-Konzept verknüpft ist.

Die folgende gekürzte JSON Antwort enthält nur das Element. `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"
   }
  }
 ]
```

Vollständige Antwort 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

Um die Ergebnisse für einen einzelnen `Condition` Ressourcentyp zu sehen, stellen Sie eine `GET` Anfrage, in `ID` der die einer bestimmten Ressource angegeben ist.

```
GET https://https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/eeb8005725ae22b35b4edbdc68cf2dfd/r4/Condition/b06d343d-ddb8-4f36-82cb-853fcd434dfd
```

Die Ergebnisse der API Geschäftstätigkeit von Amazon Comprehend Medical wurden um die folgenden Elemente ergänzt: `codemeta`, und. `modifierExtension`

`code`

Ein Element vom Typ. `CodeableConcept` Weitere Informationen finden Sie CodeableConceptim FHIRDokumentationsindex.

HealthLake hängt die folgenden drei Schlüssel-Wert-Paare an.

- `"system": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/"`: Wo URL sich das auf eine bestimmte Operation von Amazon Comprehend Medical API bezieht. In diesem Fall wird auf 0CM geschlossen. ICD1
- `"code": "I70.0"`: Wo `A52.06` ist der ICD -10-CM-Code, der das Konzept identifiziert, das in der Wissensdatenbank der Centers for Disease Control zu finden ist?
- `"display": "Atherosclerosis of aorta"`: Wo `"Other syphilitic heart involvement"` ist die lange Beschreibung des ICD -10-CM-Codes in der Ontologie.

Die folgende gekürzte JSON Antwort enthält nur das Element. `code`

```
"code": {
  "coding":
  [
    {
   "system": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/",
   "code": "I70.0",
   "display": "Atherosclerosis of aorta"
    }
  ],
  "text": "Atherosclerosis of aorta"
 }
```

Verwenden Sie das Element, um zu ermitteln, ob das Modell sicher ist, dass der zugewiesene ICD -10-CM-Code korrekt ist. `modifierExtension`

`meta`

Das `meta` Element enthält Metadaten, die angeben, ob das `code` Element Details enthält, die durch den Betrieb von Amazon Comprehend Medical API hinzugefügt wurden.

Die folgende gekürzte JSON Antwort enthält nur das Element. `meta`

```
"meta": {
  "lastUpdated": "2022-10-21T19:38:30.877Z",
  "tag": [{
   "display": "SYSTEM_GENERATED"
  }]
 }
```

`modifierExtension`

Das `modifierExtension` Element enthält weitere Informationen zum Konfidenzniveau der zugewiesenen Codes, die im `code` Element gefunden wurden. Es enthält auch Schlüssel-Wert-Paare, die einen Link zurück zum Original, das zur Generierung der Ergebnisse DocumentReference verwendet wurde, und zum zugehörigen Linkage-Ressourcentyp bereitstellen.

Für jedes hinzugefügte `coding` Element wird ein `entity-score` und ein dem `entity-Concept-Score` hinzugefügten Element angezeigt. modifierExtension Für jeden Wert im Schlüssel-Wert-Paar wird eine Punktzahl angezeigt. Denn `entity-score` dieser Wert ist das

Maß an Vertrauen, das Amazon Comprehend Medical in die Genauigkeit der Erkennung hat. Denn dieser Wert ist das Maß an Vertrauen`entity-Concept-Score`, das Amazon Comprehend Medical davon hat, dass das Unternehmen korrekt mit einem ICD -10-CM-Konzept verknüpft ist.

Die folgende gekürzte JSON Antwort enthält nur das Element. `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"
   }
  }
 ]
```

Vollständige Antwort 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"
 }
```

Beispiel 2: A**`DocumentReference`**, das den MedicationStatement Ressourcentyp enthält

Hier ist ein Beispiel für eine klinische Notiz, die auf der Begegnung eines Patienten mit einem Arzt basiert.

> ⚠ Synthetische Daten
>
> Der Text in diesem Beispiel ist synthetischer Inhalt und enthält keine persönlichen Gesundheitsinformationen (PHI).

```
Tom is not prescribed Advil
```

Die folgenden Registerkarten zeigen, wie die aufgenommenen Patientenakten je nach Ressourcentyp in Ihrem HealthLake Datenspeicher gemeldet werden.

DocumentReference

Um die Ergebnisse für einen einzelnen `DocumentReference` Ressourcentyp zu sehen, stellen Sie eine `GET` Anfrage, in `ID` der die Daten einer bestimmten Ressource bereitgestellt werden.

```
GET https://https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/eeb8005725ae22b35b4edbdc68cf2dfd/r4/DocumentReference/c549125d-a218-421f-b8bf-23614c5e796c
```

Bei Erfolg erhalten Sie einen `200` HTTP Antwortcode und die folgende gekürzte JSON Antwort.

Das Schlüssel-Wert-Paar,, gibt an`"url": "http://healthlake.amazonaws.com/system-generated-resources/"`, dass die darin enthaltenen Ressourcentypen von Amazon Comprehend Medical Operations hinzugefügt `extension` wurden. API Sie können den neuen `Linkage` Ressourcentyp und mehrere Ressourcen sehen. `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

Um die Ergebnisse für einen einzelnen `Linkage` Ressourcentyp zu sehen, stellen Sie eine `GET` Anfrage, in `ID` der die einer bestimmten Ressource bereitgestellt wird.

```
GET https://https://healthlake.your-region.amazonaws.com/
datastore/your-datastore-id/r4/eeb8005725ae22b35b4edbdc68cf2dfd/r4/
Linkage/394bb244-177b-4409-8657-26b20ed56dd7
```

Bei Erfolg erhalten Sie einen `200` HTTP Antwortcode und die folgende JSON Antwort.

Die Antwort enthält das `item` Element. Darin `"type": "source"` gibt das Schlüssel-Wert-Paar den spezifischen `DocumentReference` Eintrag an, der zur Änderung der `MedicationStatement` Ressourcentypen verwendet wird.

Sie können auch das `meta` Element und ein entsprechendes Schlüssel-Wert-Paar sehen, was darauf hinweist`"tag": [{"display": "SYSTEM_GENERATED"}]`, dass diese Ressourcen von erstellt wurden. 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

Um die Ergebnisse für einen einzelnen `MedicationStatement` Ressourcentyp zu sehen, stellen Sie eine `GET` Anfrage, in `ID` der die einer bestimmten Ressource bereitgestellt wird.

```
GET https://https://healthlake.your-region.amazonaws.com/
datastore/your-datastore-id/r4/eeb8005725ae22b35b4edbdc68cf2dfd/r4/
MedicationStatement/9a89b0d3-6681-45ca-9926-27951edce5c7
```

Der MedicationStatement Ressourcentyp ist der Ort, an dem die Ergebnisse der InferRxNorm API Operation Amazon Comprehend Medical gefunden werden. Die Ergebnisse wurden um die folgenden Elemente ergänzt: `medicationCodeableConceptmeta`, und. `modifierExtension`

medicationCodeableConcept

Ein Element vom Typ`CodeableConcept`. Weitere Informationen finden Sie CodeableConceptim FHIRDokumentationsindex.

HealthLake hängt die folgenden drei Schlüssel-Wert-Paare an.

- `"system": ""http://healthlake.amazonaws.com/aws-cm/infer-rxnorm/`: Wo URL sich das auf eine bestimmte Operation von Amazon Comprehend Medical API bezieht. In diesem Fall. InferRxNorm
- `"code": "731533"`: Wo `731533` ist eine RxNorm Konzept-ID, auch bekannt als RxCUI.

- `"display": "ibuprofen 200 MG Oral Capsule [Advil]"`: Wo `ibuprofen 200 MG Oral Capsule [Advil]` ist die Beschreibung des RxNorm Konzepts.

Die folgende gekürzte JSON Antwort enthält nur das `MedicationStatement` Element.

```
"medicationCodeableConcept": {
  "coding": [
   {
    "system": "http://healthlake.amazonaws.com/aws-cm/infer-rxnorm/",
    "code": "731533",
    "display": "ibuprofen 200 MG Oral Capsule [Advil]"
   }
  ]
 }
```

`meta`

Das `meta` Element enthält Metadaten, die angeben, ob das `code` Element Details enthält, die durch den Betrieb von Amazon Comprehend Medical API hinzugefügt wurden.

Die folgende gekürzte JSON Antwort enthält nur das Element. `meta`

```
"meta": {
  "lastUpdated": "2022-10-24T20:05:02.800Z",
  "tag": [
   {
    "display": "SYSTEM_GENERATED"
   }
  ]
 }
```

`modifierExtension`

Das `modifierExtension` Element enthält Schlüssel-Wert-Paare, die einen Link zurück zum Original, das zur Generierung der Ergebnisse DocumentReference verwendet wurde, und zum zugehörigen Linkage-Ressourcentyp bereitstellen.

```
"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"
    }
   }
]
```

Suchparameter

In der folgenden Tabelle sind die durchsuchbaren Attribute für Integrated Medical NLP aufgeführt.

Suchparameter

Suchparameter	Findet Treffer für
detectEntities-Entitäts-Kategorie	Entitätskategorie innerhalb der DetectEntities Untererweiterung innerhalb der CM-Erweiterung AWS
detectEntities-Entitätstext	Entitätstext innerhalb der DetectEntities Untererweiterung innerhalb der CM-Erweiterung AWS
detectEntities-Entitätstyp	Entitätstyp innerhalb der DetectEntities Untererweiterung innerhalb der CM-Erweiterung AWS
detectEntities-Entitäts-Score	Entity Score innerhalb der DetectEntities Untererweiterung innerhalb der CM-Erweiterung AWS
Infer-ICD10 cm-entity-text	Entitätstext innerhalb der Infer ICD1 0CM-Unter erweiterung innerhalb der CM-Erweiterung AWS
infer-icd10 cm-entity-score	Entity Score innerhalb der Infer ICD1 0CM-Unter erweiterung innerhalb der CM-Erweiterung AWS
infer-icd-10 cm-entity-concept-code	Entitätskonzeptcode innerhalb der Infer ICD1 0CM-Untererweiterung innerhalb der CM-Erweiterung AWS

Suchparameter	Findet Treffer für
infer-icd10 cm-entity-concept-description	Beschreibung des Entitätskonzepts innerhalb der Infer ICD1 0CM-Untererweiterung innerhalb der CM-Erweiterung AWS
infer-icd10 cm-entity-concept-score	Entity Concept Score innerhalb der Infer ICD1 0CM-Untererweiterung innerhalb der CM-Erweiterung AWS
infer-rxnorm-entity-score	Entity Score innerhalb der InferRxNorm Untererweiterung innerhalb der CM-Erweiterung AWS
infer-rxnorm-entity-text	Entitätstext innerhalb der InferRxNorm Untererweiterung innerhalb der CM-Erweiterung AWS
infer-rxnorm-entity-concept-Code	Entitätskonzept-Code innerhalb der InferRxNorm Untererweiterung innerhalb der CM-Erweiterung AWS
infer-rxnorm-entity-concept-Beschreibung	Beschreibung des Entitätskonzepts innerhalb der InferRxNorm Untererweiterung innerhalb der CM-Erweiterung AWS
infer-rxnorm-entity-concept-Punktzahl	Entity Concept Score innerhalb der InferRxNorm Untererweiterung innerhalb der CM-Erweiterung AWS

Um die Kriterien zu erfüllen, bei denen `EntityText` und Teil derselben Entität `EntityCategory` sind, ist eine HealthLake spezielle Suche verfügbar. In der folgenden Tabelle werden die speziellen Suchparameter beschrieben, die darin unterstützt werden HealthLake.

Suchparameter

Suchparameter	Es wurden Treffer zurückgegeben
detectEntities-entity-text-category	Wenn es in der DetectEntities Untererweiterung mindestens eine Entität gibt, die sowohl mit als auch übereinstimmt. entityText entityCategory

Suchparameter	Es wurden Treffer zurückgegeben
detectEntities-entity-type-score	Wenn es in der DetectEntities Untererweiterung mindestens eine Entität gibt, die sowohl dem als auch entspricht. entityType entityScore
detectEntities-entity-text-score	Wenn es in der DetectEntities Untererweiterung mindestens eine Entität gibt, die sowohl dem als auch entspricht. entityText entityScore
detectEntities-entity-text-type	Wenn es in der DetectEntities Untererweiterung mindestens eine Entität gibt, die sowohl dem als auch entspricht. entityText entityType
detectEntities-entity-category-score	Wenn es mindestens eine Entität gibt, die sowohl dem entityCategory als auch entspricht. entityScore
infer-icd10 -Code cm-entity-text-concept	Wenn es mindestens eine Entität in der Infer ICD10CM-Untererweiterung gibt, die der entspricht, entityText und wenn es mindestens eine conceptCode für diese Entität gibt, die dem Code entspricht.
infer-icd10 -Score cm-entity-text-concept	Wenn es mindestens eine Entität in der Infer ICD10CM-Untererweiterung gibt, die der entspricht, entityText und wenn es mindestens eine conceptScore für diese Entität gibt, die der Punktzahl entspricht.
infer-icd10 -concept-score cm-entity-concept-description	Wenn es innerhalb der Entität in der Infer ICD1 0CM-Untererweiterung mindestens ein Konzept gibt, das der Konzeptbeschreibung und dem entspricht. conceptScore
infer-rxnorm-entity-text-Konzeptcode	Wenn es in der InferRxNorm Untererweiterung mindestens eine Entität gibt, die mit der übereinstimmt, entityText und wenn es mindestens eine conceptCode für diese Entität gibt, die dem Code entspricht.

Suchparameter	Es wurden Treffer zurückgegeben
infer-rxnorm-entity-text-Konzept-Score	Wenn es in der InferRxNorm Untererweiterung mindestens eine Entität gibt, die der entspricht, entityText und wenn es conceptScore für diese Entität mindestens eine gibt, die der Punktzahl entspricht.
infer-rxnorm-entity-concept-description-concept-score	Wenn es innerhalb der Entität in der InferRxNorm Untererweiterung mindestens ein Konzept gibt, das der Konzeptbeschreibung und der entspricht. conceptScore

Sicherheit in AWS HealthLake

Cloud-Sicherheit AWS hat höchste Priorität. Als AWS Kunde profitieren Sie von einer Rechenzentrums- und Netzwerkarchitektur, die darauf ausgelegt sind, die Anforderungen der sicherheitssensibelsten Unternehmen zu erfüllen.

Sicherheit ist eine gemeinsame Verantwortung von Ihnen AWS und Ihnen. Das Modell der geteilten Verantwortung beschreibt dies als Sicherheit der Cloud selbst und Sicherheit in der Cloud:

- Die Sicherheit der Cloud AWS ist für den Schutz der Infrastruktur verantwortlich, auf der AWS Dienste in der AWS Cloud ausgeführt werden. AWS bietet Ihnen auch Dienste, die Sie sicher nutzen können. Externe Prüfer testen und verifizieren regelmäßig die Wirksamkeit unserer Sicherheitsmaßnahmen im Rahmen der AWS . Weitere Informationen zu den Compliance-Programmen, die für gelten HealthLake, finden Sie unter AWSServices im Umfang nach Compliance-Programmen AWS .
- Sicherheit in der Cloud — Ihre Verantwortung richtet sich nach dem AWS Dienst, den Sie nutzen. Sie sind auch für andere Faktoren verantwortlich, etwa für die Vertraulichkeit Ihrer Daten, für die Anforderungen Ihres Unternehmens und für die geltenden Gesetze und Vorschriften.

Diese Dokumentation hilft Ihnen zu verstehen, wie Sie das Modell der gemeinsamen Verantwortung bei der Nutzung anwenden können HealthLake. In den folgenden Themen erfahren Sie, wie Sie die Konfiguration vornehmen HealthLake , um Ihre Sicherheits- und Compliance-Ziele zu erreichen. Sie erfahren auch, wie Sie andere AWS Dienste nutzen können, die Sie bei der Überwachung und Sicherung Ihrer HealthLake Ressourcen unterstützen.

Themen

Datenschutz in AWS HealthLake

Das Modell der AWS gemeinsamen Verantwortung und geteilter Verantwortung gilt für den Datenschutz in AWS HealthLake. Wie in diesem Modell beschrieben, AWS ist verantwortlich für den Schutz der globalen Infrastruktur, auf der alle Systeme laufen AWS Cloud. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Sie sind auch für die Sicherheitskonfiguration und die Verwaltungsaufgaben für die von Ihnen verwendeten AWS-Services verantwortlich. Weitere Informationen zum Datenschutz finden Sie im Abschnitt Datenschutz FAQ. Informationen zum Datenschutz in Europa finden Sie im AWS Shared Responsibility Model und im GDPR Blogbeitrag auf dem AWS Security Blog.

Aus Datenschutzgründen empfehlen wir, dass Sie Ihre AWS-Konto Anmeldeinformationen schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) einrichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem empfehlen wir, die Daten mit folgenden Methoden schützen:

- Verwenden Sie für jedes Konto eine Multi-Faktor-Authentifizierung (MFA).
- Verwenden SieSSL/TLS, um mit AWS Ressourcen zu kommunizieren. Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Einrichtung API und Protokollierung von Benutzeraktivitäten mit AWS CloudTrail. Informationen zur Verwendung von CloudTrail Pfaden zur Erfassung von AWS Aktivitäten finden Sie unter Arbeiten mit CloudTrail Pfaden im AWS CloudTrail Benutzerhandbuch.
- Verwenden Sie AWS Verschlüsselungslösungen zusammen mit allen darin enthaltenen Standardsicherheitskontrollen AWS-Services.
- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu schützen.
- Wenn Sie FIPS 140-3 validierte kryptografische Module für den Zugriff AWS über eine Befehlszeilenschnittstelle oder eine benötigenAPI, verwenden Sie einen Endpunkt. FIPS Weitere Informationen zu den verfügbaren FIPS Endpunkten finden Sie unter Federal Information Processing Standard () 140-3. FIPS

Wir empfehlen dringend, in Freitextfeldern, z. B. im Feld Name, keine vertraulichen oder sensiblen Informationen wie die E-Mail-Adressen Ihrer Kunden einzugeben. Dies gilt auch, wenn Sie mit der Konsole arbeiten HealthLake oder sie anderweitig AWS-Services verwenden,API, AWS CLI oder. AWS SDKs Alle Daten, die Sie in Tags oder Freitextfelder eingeben, die für Namen verwendet werden, können für Abrechnungs- oder Diagnoseprotokolle verwendet werden. Wenn

Sie einem externen Server eine URL zur Verfügung stellen, empfehlen wir dringend, dass Sie keine Anmeldeinformationen in den angebenURL, um Ihre Anfrage an diesen Server zu überprüfen.

Verschlüsselung bei REST für AWS HealthLake

HealthLake bietet standardmäßig Verschlüsselung zum Schutz vertraulicher Kundendaten im Speicher mithilfe eines diensteigenen AWS Key Management Service (AWSKMS) -Schlüssels. Kundenverwaltete KMS Schlüssel werden ebenfalls unterstützt und sind sowohl für den Import als auch für den Export von Dateien aus einem Datenspeicher erforderlich. Weitere Informationen zu Customer-managed KMS Key finden Sie unter Amazon Key Management Service. Kunden können bei der Erstellung eines AWS KMS Datenspeichers zwischen einem eigenen KMS Schlüssel oder einem vom Kunden verwalteten Schlüssel wählen. Die Verschlüsselungskonfiguration kann nicht geändert werden, nachdem ein Datenspeicher erstellt wurde. Wenn ein Datenspeicher einen AWS eigenen KMS Schlüssel verwendet, wird dieser als bezeichnet, AWS_OWNED_KMS_KEY und der spezifische Schlüssel, der für die Verschlüsselung verwendet wird, wird im Ruhezustand nicht angezeigt.

AWSeigener Schlüssel KMS

HealthLake verwendet diese Schlüssel standardmäßig, um potenziell vertrauliche Informationen wie personenbezogene Daten oder Daten mit privaten Gesundheitsinformationen (PHI) im Ruhezustand automatisch zu verschlüsseln. AWSKMSEigene Schlüssel werden nicht in Ihrem Konto gespeichert. Sie sind Teil einer Sammlung von KMS Schlüsseln, die AWS Eigentümer sind und von denen sie verwaltet werden, sodass sie in mehreren AWS Konten verwendet werden können. AWSDienste können AWS eigene KMS Schlüssel verwenden, um Ihre Daten zu schützen. Sie können AWS eigene KMS Schlüssel nicht einsehen, verwalten, verwenden oder deren Verwendung überprüfen. Sie müssen jedoch keine Arbeit verrichten oder Programme ändern, um die Schlüssel zu schützen, mit denen Ihre Daten verschlüsselt werden.

Wenn Sie AWS eigene KMS Schlüssel verwenden, wird Ihnen weder eine monatliche Gebühr noch eine Nutzungsgebühr berechnet, und sie werden nicht auf die AWS KMS Kontingente für Ihr Konto angerechnet. Weitere Informationen finden Sie unter AWSEigene Schlüssel.

Vom Kunden verwaltete KMS Schlüssel

HealthLake unterstützt die Verwendung eines symmetrischen, vom Kunden verwalteten KMS Schlüssels, den Sie selbst erstellen, besitzen und verwalten, um eine zweite Verschlüsselungsebene

zur bestehenden AWS Verschlüsselung hinzuzufügen. Da Sie die volle Kontrolle über diese Verschlüsselungsebene haben, können Sie beispielsweise folgende Aufgaben ausführen:

- Einrichtung und Aufrechterhaltung wichtiger Richtlinien, IAM Richtlinien und Zuschüsse
- Kryptographisches Material mit rotierendem Schlüssel
- Aktivieren und Deaktivieren wichtiger Richtlinien
- Hinzufügen von Tags
- Erstellen von Schlüsselaliasen
- Schlüssel für das Löschen von Schlüsseln planen

Sie können es auch verwenden CloudTrail , um die Anfragen nachzuverfolgen, die in Ihrem Namen HealthLake AWS KMS an gesendet werden. Es AWS KMS fallen zusätzliche Gebühren an. Weitere Informationen finden Sie unter Kundeneigene Schlüssel.

Einen kundenverwalteten Schlüssel erstellen

Sie können einen symmetrischen, vom Kunden verwalteten Schlüssel mithilfe der AWS Management Console oder der erstellen. AWS KMS APIs

Folgen Sie den Schritten zur Erstellung eines symmetrischen, vom Kunden verwalteten Schlüssels im AWS Key Management Service Developer Guide.

Schlüsselrichtlinien steuern den Zugriff auf den vom Kunden verwalteten Schlüssel. Jeder vom Kunden verwaltete Schlüssel muss über genau eine Schlüsselrichtlinie verfügen, die aussagt, wer den Schlüssel wie verwenden kann. Wenn Sie Ihren vom Kunden verwalteten Schlüssel erstellen, können Sie eine Schlüsselrichtlinie angeben. Weitere Informationen finden Sie unter Verwaltung des Zugriffs auf vom Kunden verwaltete Schlüssel im AWS Key Management Service Developer Guide.

Um Ihren vom Kunden verwalteten Schlüssel mit Ihren HealthLake Ressourcen verwenden zu können, müssen kms: CreateGrant -Operationen in der Schlüsselrichtlinie zugelassen sein. Dadurch wird einem vom Kunden verwalteten Schlüssel ein Zuschuss hinzugefügt, der den Zugriff auf einen bestimmten KMS Schlüssel steuert, wodurch ein Benutzer Zugriff auf die für kms:grant-Operationen erforderlichen Funktionen erhält. HealthLake Weitere Informationen finden Sie unter Zuschüsse verwenden.

Damit Sie Ihren vom Kunden verwalteten KMS Schlüssel mit Ihren HealthLake Ressourcen verwenden können, müssen die folgenden API Vorgänge in der Schlüsselrichtlinie zulässig sein:

- kms: CreateGrant fügt einem bestimmten vom Kunden verwalteten KMS Schlüssel Zuschüsse hinzu, wodurch der Zugriff auf Zuschussoperationen ermöglicht wird.
- kms: DescribeKey stellt die vom Kunden verwalteten Schlüsselinformationen bereit, die zur Validierung des Schlüssels erforderlich sind. Dies ist für alle Operationen erforderlich.
- kms: GenerateDataKey bietet Zugriff auf die Verschlüsselung von Ressourcen im Ruhezustand für alle Schreibvorgänge.
- kms:Decrypt bietet Zugriff auf Lese- oder Suchvorgänge für verschlüsselte Ressourcen.

Im Folgenden finden Sie ein Beispiel für eine Richtlinienanweisung, die es einem Benutzer ermöglicht, einen Datenspeicher zu erstellen und mit diesem zu interagieren, der mit diesem AWS HealthLake Schlüssel verschlüsselt ist:

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

Erforderliche IAM Berechtigungen für die Verwendung eines vom Kunden verwalteten KMS Schlüssels

Beim Erstellen eines Datenspeichers mit aktivierter AWS KMS Verschlüsselung mithilfe eines vom Kunden verwalteten KMS Schlüssels sind Berechtigungen sowohl für die Schlüsselrichtlinie als auch für die IAM Richtlinie für den Benutzer oder die Rolle erforderlich, die den HealthLake Datenspeicher erstellt.

Sie können den ViaService Bedingungsschlüssel kms: verwenden, um die Verwendung des KMS Schlüssels auf Anfragen zu beschränken, die von stammen HealthLake.

Weitere Informationen zu wichtigen Richtlinien finden Sie unter Aktivieren von IAM Richtlinien im AWS Key Management Service Developer Guide.

Der IAM Benutzer, die IAM Rolle oder das AWS Konto, die Ihre Repositorys erstellt, muss über die Berechtigungen kms:CreateGrant, kms: GenerateDataKey und sowie kms: DescribeKey über die erforderlichen HealthLake Berechtigungen verfügen.

Wie HealthLake verwendet man Zuschüsse in AWS KMS

HealthLake erfordert einen Zuschuss, um Ihren vom Kunden verwalteten KMS Schlüssel verwenden zu können. Wenn Sie einen Datenspeicher erstellen, der mit einem vom Kunden verwalteten KMS Schlüssel verschlüsselt ist, HealthLake erstellt in Ihrem Namen einen Zuschuss, indem Sie eine CreateGrantAnfrage an senden AWSKMS. Grants in AWS KMS werden verwendet, um HealthLake Zugriff auf einen KMS Schlüssel in einem Kundenkonto zu gewähren.

Die Zuschüsse, die HealthLake in Ihrem Namen gewährt werden, sollten nicht zurückgezogen oder zurückgezogen werden. Wenn Sie die Erteilung, die die HealthLake Erlaubnis zur Verwendung der AWS KMS Schlüssel in Ihrem Konto gewährt, widerrufen oder zurückziehen, HealthLake können Sie nicht auf diese Daten zugreifen, neue FHIR Ressourcen verschlüsseln, die in den Datenspeicher übertragen werden, oder sie entschlüsseln, wenn sie abgerufen werden. Wenn Sie einen Zuschuss für widerrufen oder zurückziehen HealthLake, wird die Änderung sofort wirksam. Um die Zugriffsrechte zu entziehen, sollten Sie den Datenspeicher löschen, anstatt die Gewährung zu widerrufen. Wenn ein Datenspeicher gelöscht wird, werden die Zuschüsse HealthLake in Ihrem Namen zurückgezogen.

Überwachen Sie Ihre Verschlüsselungsschlüssel für HealthLake

Sie können CloudTrail damit die Anfragen verfolgen, die in Ihrem Namen HealthLake AWS KMS an gesendet werden, wenn Sie einen vom Kunden verwalteten KMS Schlüssel verwenden. In den

Protokolleinträgen im CloudTrail Protokoll wird healthlake.amazonaws.com im userAgent Feld angezeigt, sodass die Anfragen von eindeutig unterschieden werden können. HealthLake

Die folgenden Beispiele sind CloudTrail Ereignisse für CreateGrant,, GenerateDataKey Decrypt und zur Überwachung von AWS KMS Vorgängen, die aufgerufen werden, DescribeKey um auf Daten zuzugreifen, die mit Ihrem vom HealthLake Kunden verwalteten Schlüssel verschlüsselt wurden.

Im Folgenden wird gezeigt, wie Sie CreateGrant den HealthLake Zugriff auf einen vom Kunden bereitgestellten KMS Schlüssel HealthLake ermöglichen und mit diesem KMS Schlüssel alle gespeicherten Kundendaten verschlüsseln können.

Benutzer müssen keine eigenen Zuschüsse erstellen. HealthLake erstellt in Ihrem Namen einen Zuschuss, indem Sie eine CreateGrant Anfrage an senden AWSKMS. Zuschüsse in AWS KMS werden verwendet, um HealthLake Zugriff auf einen AWS KMS Schlüssel in einem Kundenkonto zu gewähren.

```
         {
"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"
}
```


Die folgenden Beispiele zeigen, wie Sie sicherstellen können GenerateDataKey , dass der Benutzer vor dem Speichern über die erforderlichen Berechtigungen zum Verschlüsseln von Daten verfügt.

```
         {
    "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"
}
```

Das folgende Beispiel zeigt, wie der Vorgang Decrypt HealthLake aufgerufen wird, um den gespeicherten verschlüsselten Datenschlüssel für den Zugriff auf die verschlüsselten Daten zu verwenden.

```
         {
    "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"
}
```

Das folgende Beispiel zeigt, wie der DescribeKey Vorgang HealthLake verwendet wird, um zu überprüfen, ob sich der AWS KMS Schlüssel im Besitz des AWS KMS Kunden in einem verwendbaren Zustand befindet, und um einem Benutzer bei der Fehlerbehebung zu helfen, falls er nicht funktioniert.

```
            {
    "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"
}
```


Weitere Informationen

Die folgenden Ressourcen bieten weitere Informationen zur Verschlüsselung von Daten im Ruhezustand.

Weitere Informationen zu den grundlegenden Konzepten des AWS Key Management Service finden Sie in der AWS KMS Dokumentation.

Weitere Informationen zu bewährten Sicherheitsmethoden finden Sie in der AWS KMS Dokumentation.

Verschlüsselung bei der Übertragung für AWS HealthLake

AWS HealthLake verwendet TLS 1.2, um Daten bei der Übertragung über den öffentlichen Endpunkt und über Backend-Dienste zu verschlüsseln.

Identitäts- und Zugriffsmanagement für AWS HealthLake

AWS Identity and Access Management (IAM) hilft einem Administrator AWS-Service , den Zugriff auf AWS Ressourcen sicher zu kontrollieren. IAMAdministratoren kontrollieren, wer authentifiziert (angemeldet) und autorisiert werden kann (über Berechtigungen verfügt), um HealthLake Ressourcen zu verwenden. IAMist eine AWS-Service , die Sie ohne zusätzliche Kosten verwenden können.

Themen

Zielgruppe

Wie Sie AWS Identity and Access Management (IAM) verwenden, hängt von der Arbeit ab, in der Sie arbeiten HealthLake.

Dienstbenutzer — Wenn Sie den HealthLake Dienst für Ihre Arbeit verwenden, stellt Ihnen Ihr Administrator die erforderlichen Anmeldeinformationen und Berechtigungen zur Verfügung. Wenn Sie für Ihre Arbeit mehr HealthLake Funktionen verwenden, benötigen Sie möglicherweise

zusätzliche Berechtigungen. Wenn Sie die Funktionsweise der Zugriffskontrolle nachvollziehen, wissen Sie bereits, welche Berechtigungen Sie von Ihrem Administrator anfordern müssen. Unter Problembehandlung bei AWS HealthLake Identität und Zugriff finden Sie nützliche Informationen für den Fall, dass Sie keinen Zugriff auf eine Feature in HealthLake haben.

Serviceadministrator — Wenn Sie in Ihrem Unternehmen für die HealthLake Ressourcen verantwortlich sind, haben Sie wahrscheinlich vollen Zugriff auf HealthLake. Es ist Ihre Aufgabe, zu bestimmen, auf welche HealthLake Funktionen und Ressourcen Ihre Servicebenutzer zugreifen sollen. Sie müssen anschließend bei Ihrem IAM-Administrator entsprechende Änderungen für die Berechtigungen Ihrer Service-Benutzer anfordern. Lesen Sie die Informationen auf dieser Seite, um die Grundkonzepte von IAM zu verstehen. Weitere Informationen darüber, wie Ihr Unternehmen IAM mit nutzen kann HealthLake, finden Sie unterWie AWS HealthLake funktioniert mit IAM.

IAMAdministrator — Wenn Sie ein IAM Administrator sind, möchten Sie vielleicht mehr darüber erfahren, wie Sie Richtlinien schreiben können, um den Zugriff darauf zu verwalten HealthLake. Beispiele für HealthLake identitätsbasierte Richtlinien, die Sie in verwenden könnenIAM, finden Sie unter. Beispiele für identitätsbasierte Richtlinien für AWS HealthLake

Authentifizierung mit Identitäten

Authentifizierung ist die Art und Weise, wie Sie sich AWS mit Ihren Identitätsdaten anmelden. Sie müssen als IAM Benutzer authentifiziert (angemeldet AWS) sein oder eine IAM Rolle übernehmen. Root-Benutzer des AWS-Kontos

Sie können sich AWS als föderierte Identität anmelden, indem Sie Anmeldeinformationen verwenden, die über eine Identitätsquelle bereitgestellt wurden. AWS IAM Identity Center (IAMIdentity Center-) Nutzer, die Single-Sign-On-Authentifizierung Ihres Unternehmens und Ihre Google- oder Facebook-Anmeldeinformationen sind Beispiele für föderierte Identitäten. Wenn Sie sich als Verbundidentität anmelden, hat der Administrator vorher mithilfe von IAM-Rollen einen Identitätsverbund eingerichtet. Wenn Sie über den Verbund darauf zugreifen AWS , übernehmen Sie indirekt eine Rolle.

Je nachdem, welcher Benutzertyp Sie sind, können Sie sich beim AWS Management Console oder beim AWS Zugangsportal anmelden. Weitere Informationen zur Anmeldung finden Sie AWS unter So melden Sie sich bei Ihrem an AWS-Konto im AWS-Anmeldung Benutzerhandbuch.

Wenn Sie AWS programmgesteuert darauf zugreifen, AWS stellt es ein Software Development Kit (SDK) und eine Befehlszeilenschnittstelle (CLI) bereit, mit der Sie Ihre Anfragen mithilfe Ihrer Anmeldeinformationen kryptografisch signieren können. Wenn Sie keine AWS Tools verwenden, müssen Sie Anfragen selbst signieren. Weitere Informationen zur Verwendung der empfohlenen

Methode, um Anfragen selbst zu signieren, finden Sie im IAMBenutzerhandbuch unter AWS Signature Version 4 für API Anfragen.

Unabhängig von der verwendeten Authentifizierungsmethode müssen Sie möglicherweise zusätzliche Sicherheitsinformationen bereitstellen. AWS Empfiehlt beispielsweise, die Multi-Faktor-Authentifizierung (MFA) zu verwenden, um die Sicherheit Ihres Kontos zu erhöhen. Weitere Informationen finden Sie unter Multi-Faktor-Authentifizierung im AWS IAM Identity Center Benutzerhandbuch und AWS Multi-Faktor-Authentifizierung IAM im IAM Benutzerhandbuch.

AWS-Konto Root-Benutzer

Wenn Sie einen erstellen AWS-Konto, beginnen Sie mit einer Anmeldeidentität, die vollständigen Zugriff auf alle AWS-Services Ressourcen im Konto hat. Diese Identität wird als AWS-Konto Root-Benutzer bezeichnet. Sie können darauf zugreifen, indem Sie sich mit der E-Mail-Adresse und dem Passwort anmelden, mit denen Sie das Konto erstellt haben. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Schützen Sie Ihre Root-Benutzer-Anmeldeinformationen. Verwenden Sie diese nur, um die Aufgaben auszuführen, die nur der Root-Benutzer ausführen kann. Eine vollständige Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie im Benutzerhandbuch unter Aufgaben, für die Root-Benutzeranmeldedaten erforderlich sind. IAM

Verbundidentität

Als bewährte Methode sollten menschliche Benutzer, einschließlich Benutzer, die Administratorzugriff benötigen, für den Zugriff AWS-Services mithilfe temporärer Anmeldeinformationen den Verbund mit einem Identitätsanbieter verwenden.

Eine föderierte Identität ist ein Benutzer aus Ihrem Unternehmensbenutzerverzeichnis, einem Web-Identitätsanbieter AWS Directory Service, dem Identity Center-Verzeichnis oder einem beliebigen Benutzer, der mithilfe AWS-Services von Anmeldeinformationen zugreift, die über eine Identitätsquelle bereitgestellt wurden. Wenn föderierte Identitäten darauf zugreifen AWS-Konten, übernehmen sie Rollen, und die Rollen stellen temporäre Anmeldeinformationen bereit.

Für die zentrale Zugriffsverwaltung empfehlen wir Ihnen, AWS IAM Identity Center zu verwenden. Sie können Benutzer und Gruppen in IAM Identity Center erstellen, oder Sie können eine Verbindung zu einer Gruppe von Benutzern und Gruppen in Ihrer eigenen Identitätsquelle herstellen und diese synchronisieren, um sie in all Ihren AWS-Konten Anwendungen zu verwenden. Informationen zu IAM Identity Center finden Sie unter Was ist IAM Identity Center? im AWS IAM Identity Center Benutzerhandbuch.

IAM-Benutzer und -Gruppen

Ein IAMBenutzer ist eine Identität innerhalb Ihres Unternehmens AWS-Konto , die über spezifische Berechtigungen für eine einzelne Person oder Anwendung verfügt. Wir empfehlen, sich nach Möglichkeit auf temporäre Anmeldeinformationen zu verlassen, anstatt IAM Benutzer mit langfristigen Anmeldeinformationen wie Passwörtern und Zugriffsschlüsseln zu erstellen. Wenn Sie jedoch spezielle Anwendungsfälle haben, für die langfristige Anmeldeinformationen von IAM Benutzern erforderlich sind, empfehlen wir, die Zugriffsschlüssel abwechselnd zu verwenden. Weitere Informationen finden Sie im Benutzerhandbuch unter Regelmäßiges Rotieren von Zugriffsschlüsseln für Anwendungsfälle, für die IAM langfristige Anmeldeinformationen erforderlich sind.

Ein IAM-Gruppe ist eine Identität, die eine Sammlung von IAM.Benutzern angibt. Sie können sich nicht als Gruppe anmelden. Mithilfe von Gruppen können Sie Berechtigungen für mehrere Benutzer gleichzeitig angeben. Gruppen vereinfachen die Verwaltung von Berechtigungen, wenn es zahlreiche Benutzer gibt. Sie könnten beispielsweise eine Gruppe benennen IAMAdminsund dieser Gruppe Berechtigungen zur Verwaltung von IAM Ressourcen erteilen.

Benutzer unterscheiden sich von Rollen. Ein Benutzer ist einer einzigen Person oder Anwendung eindeutig zugeordnet. Eine Rolle kann von allen Personen angenommen werden, die sie benötigen. Benutzer besitzen dauerhafte Anmeldeinformationen. Rollen stellen temporäre Anmeldeinformationen bereit. Weitere Informationen finden Sie im Benutzerhandbuch unter Anwendungsfälle für IAM IAM Benutzer.

IAM-Rollen

Eine IAMRolle ist eine Identität innerhalb von Ihnen AWS-Konto , die über bestimmte Berechtigungen verfügt. Sie ist einem IAM-Benutzer vergleichbar, jedoch nicht mit einer bestimmten Person verknüpft. Um vorübergehend eine IAM Rolle in der zu übernehmen AWS Management Console, können Sie von einem Benutzer zu einer IAM Rolle (Konsole) wechseln. Sie können eine Rolle übernehmen, indem Sie eine AWS CLI AWS API OR-Operation aufrufen oder eine benutzerdefinierte Operation verwendenURL. Weitere Informationen zu Methoden zur Verwendung von Rollen finden Sie unter Methoden zur Übernahme einer Rolle im IAMBenutzerhandbuch.

IAM-Rollen mit temporären Anmeldeinformationen sind in folgenden Situationen hilfreich:

- Verbundbenutzerzugriff – Um einer Verbundidentität Berechtigungen zuzuweisen, erstellen Sie eine Rolle und definieren Berechtigungen für die Rolle. Wird eine Verbundidentität authentifiziert, so wird die Identität der Rolle zugeordnet und erhält die von der Rolle definierten Berechtigungen. Informationen zu Rollen für den Verbund finden Sie im IAMBenutzerhandbuch unter Erstellen einer

Rolle für einen externen Identitätsanbieter (Federation). Wenn Sie IAM Identity Center verwenden, konfigurieren Sie einen Berechtigungssatz. Um zu kontrollieren, worauf Ihre Identitäten nach der Authentifizierung zugreifen können, korreliert IAM Identity Center den Berechtigungssatz mit einer Rolle in. IAM Informationen zu Berechtigungssätzen finden Sie unter Berechtigungssätze im AWS IAM Identity Center -Benutzerhandbuch.

- Temporäre IAM Benutzerberechtigungen — Ein IAM Benutzer oder eine Rolle kann eine IAM Rolle übernehmen, um vorübergehend verschiedene Berechtigungen für eine bestimmte Aufgabe zu übernehmen.
- Kontoübergreifender Zugriff: Sie können eine IAM-Rolle verwenden, um jemandem (einem vertrauenswürdigen Prinzipal) in einem anderen Konto den Zugriff auf Ressourcen in Ihrem Konto zu ermöglichen. Rollen stellen die primäre Möglichkeit dar, um kontoübergreifendem Zugriff zu gewähren. Bei einigen können Sie AWS-Services jedoch eine Richtlinie direkt an eine Ressource anhängen (anstatt eine Rolle als Proxy zu verwenden). Informationen zum Unterschied zwischen Rollen und ressourcenbasierten Richtlinien für den kontenübergreifenden Zugriff finden Sie IAMim Benutzerhandbuch unter Kontoübergreifender Ressourcenzugriff. IAM
- Serviceübergreifender Zugriff — Einige AWS-Services verwenden Funktionen in anderen. AWS-Services Wenn Sie beispielsweise einen Service aufrufen, ist es üblich, dass dieser Service Anwendungen in Amazon ausführt EC2 oder Objekte in Amazon S3 speichert. Ein Dienst kann dies mit den Berechtigungen des aufrufenden Prinzipals mit einer Servicerolle oder mit einer serviceverknüpften Rolle tun.
  - Zugriffssitzungen weiterleiten (FAS) — Wenn Sie einen IAM Benutzer oder eine Rolle verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. Wenn Sie einige Dienste verwenden, führen Sie möglicherweise eine Aktion aus, die dann eine weitere Aktion in einem anderen Dienst auslöst. FASverwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, kombiniert mit der Anforderung, Anfragen AWS-Service an nachgeschaltete Dienste zu stellen. FASAnfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS-Services oder Ressourcen erforderlich sind. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien beim Stellen von FAS Anfragen finden Sie unter Zugriffssitzungen weiterleiten.
  - Servicerolle — Eine Servicerolle ist eine IAMRolle, die ein Dienst übernimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter Erstellen einer Rolle zum Delegieren von Berechtigungen AWS-Service an eine.

- Dienstbezogene Rolle — Eine dienstverknüpfte Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Servicebezogene Rollen erscheinen in Ihrem Dienst AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für serviceverknüpfte Rollen anzeigen, aber nicht bearbeiten.
- Auf Amazon ausgeführte Anwendungen EC2 — Sie können eine IAM Rolle verwenden, um temporäre Anmeldeinformationen für Anwendungen zu verwalten, die auf einer EC2 Instance ausgeführt werden und AWS API Anfragen stellen AWS CLI . Das ist empfehlenswerter, als Zugriffsschlüssel innerhalb der EC2-Instance zu speichern. Um einer EC2 Instance eine AWS Rolle zuzuweisen und sie all ihren Anwendungen zur Verfügung zu stellen, erstellen Sie ein Instance-Profil, das an die Instance angehängt ist. Ein Instance-Profil enthält die Rolle und ermöglicht, dass Programme, die in der EC2-Instance ausgeführt werden, temporäre Anmeldeinformationen erhalten. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter Verwenden einer IAM Rolle, um Berechtigungen für Anwendungen zu erteilen, die auf EC2 Amazon-Instances ausgeführt werden.

Verwalten des Zugriffs mit Richtlinien

Sie kontrollieren den Zugriff, AWS indem Sie Richtlinien erstellen und diese mit AWS Identitäten oder Ressourcen verknüpfen. Eine Richtlinie ist ein Objekt, AWS das, wenn es einer Identität oder Ressource zugeordnet ist, deren Berechtigungen definiert. AWS wertet diese Richtlinien aus, wenn ein Prinzipal (Benutzer, Root-Benutzer oder Rollensitzung) eine Anfrage stellt. Die Berechtigungen in den Richtlinien legen fest, ob eine Anforderung zugelassen oder abgelehnt wird. Die meisten Richtlinien werden in AWS Form von JSON Dokumenten gespeichert. Weitere Informationen zur Struktur und zum Inhalt von JSON Richtliniendokumenten finden Sie im IAMBenutzerhandbuch unter Überblick über JSON Richtlinien.

Administratoren können mithilfe von AWS JSON Richtlinien festlegen, wer Zugriff auf was hat. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Um Benutzern die Erlaubnis zu erteilen, Aktionen mit den Ressourcen durchzuführen, die sie benötigen, kann ein IAM Administrator IAM Richtlinien erstellen. Der Administrator kann dann die IAM Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen übernehmen.

IAM-Richtlinien definieren Berechtigungen für eine Aktion unabhängig von der Methode, die Sie zur Ausführung der Aktion verwenden. Angenommen, es gibt eine Richtlinie, die Berechtigungen für die `iam:GetRole`-Aktion erteilt. Ein Benutzer mit dieser Richtlinie kann Rolleninformationen aus dem AWS Management Console AWS CLI, dem oder dem abrufen AWS API.

Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind Dokumente mit JSON Berechtigungsrichtlinien, die Sie an eine Identität anhängen können, z. B. an einen IAM Benutzer, eine Benutzergruppe oder eine Rolle. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen einer identitätsbasierten Richtlinie finden Sie im Benutzerhandbuch unter Definieren benutzerdefinierter IAM Berechtigungen mit vom Kunden verwalteten Richtlinien. IAM

Identitätsbasierte Richtlinien können weiter als Inline-Richtlinien oder verwaltete Richtlinien kategorisiert werden. Inline-Richtlinien sind direkt in einen einzelnen Benutzer, eine einzelne Gruppe oder eine einzelne Rolle eingebettet. Verwaltete Richtlinien sind eigenständige Richtlinien, die Sie mehreren Benutzern, Gruppen und Rollen in Ihrem System zuordnen können. AWS-Konto Zu den verwalteten Richtlinien gehören AWS verwaltete Richtlinien und vom Kunden verwaltete Richtlinien. Informationen zur Auswahl zwischen einer verwalteten Richtlinie oder einer Inline-Richtlinie finden Sie im IAMBenutzerhandbuch unter Wählen Sie zwischen verwalteten Richtlinien und Inline-Richtlinien.

Ressourcenbasierte Richtlinien

Ressourcenbasierte Richtlinien sind JSON Richtliniendokumente, die Sie an eine Ressource anhängen. Beispiele für ressourcenbasierte Richtlinien sind IAM Rollenvertrauensrichtlinien und Amazon S3 S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie einen Prinzipal angeben. Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services

Ressourcenbasierte Richtlinien sind Richtlinien innerhalb dieses Diensts. Sie können AWS verwaltete Richtlinien nicht IAM in einer ressourcenbasierten Richtlinie verwenden.

Zugriffskontrolllisten () ACLs

Zugriffskontrolllisten (ACLs) steuern, welche Principals (Kontomitglieder, Benutzer oder Rollen) über Zugriffsberechtigungen für eine Ressource verfügen. ACLsähneln ressourcenbasierten Richtlinien, verwenden jedoch nicht das JSON Richtliniendokumentformat.

Amazon S3 und AWS WAF Amazon VPC sind Beispiele für Dienste, die Unterstützung bietenACLs. Weitere Informationen finden Sie unter Übersicht über ACLs die Zugriffskontrollliste (ACL) im Amazon Simple Storage Service Developer Guide.

Weitere Richtlinientypen

AWS unterstützt zusätzliche, weniger verbreitete Richtlinientypen. Diese Richtlinientypen können die maximalen Berechtigungen festlegen, die Ihnen von den häufiger verwendeten Richtlinientypen erteilt werden können.

- Berechtigungsgrenzen — Eine Berechtigungsgrenze ist eine erweiterte Funktion, mit der Sie die maximalen Berechtigungen festlegen, die eine identitätsbasierte Richtlinie einer IAM Entität (IAMBenutzer oder Rolle) gewähren kann. Sie können eine Berechtigungsgrenze für eine Entität festlegen. Die daraus resultierenden Berechtigungen sind der Schnittpunkt der identitätsbasierten Richtlinien einer Entität und ihrer Berechtigungsgrenzen. Ressourcenbasierte Richtlinien, die den Benutzer oder die Rolle im Feld `Principal` angeben, werden nicht durch Berechtigungsgrenzen eingeschränkt. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen zu Berechtigungsgrenzen finden Sie im IAMBenutzerhandbuch unter Berechtigungsgrenzen für IAM Entitäten.
- Dienststeuerungsrichtlinien (SCPs) — SCPs sind JSON Richtlinien, die die maximalen Berechtigungen für eine Organisation oder Organisationseinheit (OU) in festlegen AWS Organizations. AWS Organizations ist ein Dienst zur Gruppierung und zentralen Verwaltung mehrerer AWS-Konten Unternehmenseigentümer. Wenn Sie alle Funktionen in einer Organisation aktivieren, können Sie Richtlinien zur Servicesteuerung (SCPs) auf einige oder alle Ihre Konten anwenden. Das SCP schränkt die Berechtigungen für Entitäten in Mitgliedskonten ein, einschließlich der einzelnen Root-Benutzer des AWS-Kontos. Weitere Informationen zu Organizations und SCPs finden Sie unter Richtlinien zur Servicesteuerung im AWS Organizations Benutzerhandbuch.
- Richtlinien zur Ressourcenkontrolle (RCPs) — RCPs sind JSON Richtlinien, mit denen Sie die maximal verfügbaren Berechtigungen für Ressourcen in Ihren Konten festlegen können, ohne die IAM Richtlinien aktualisieren zu müssen, die jeder Ressource zugeordnet sind, deren Eigentümer Sie sind. Sie RCP schränken die Berechtigungen für Ressourcen in Mitgliedskonten ein und

können sich auf die effektiven Berechtigungen für Identitäten auswirken, einschließlich der Root-Benutzer des AWS-Kontos, unabhängig davon, ob sie zu Ihrer Organisation gehören. Weitere Informationen zu OrganizationsRCPs, einschließlich einer Liste AWS-Services dieser Support-LeistungenRCPs, finden Sie unter Resource Control Policies (RCPs) im AWS Organizations Benutzerhandbuch.

- Sitzungsrichtlinien – Sitzungsrichtlinien sind erweiterte Richtlinien, die Sie als Parameter übergeben, wenn Sie eine temporäre Sitzung für eine Rolle oder einen verbundenen Benutzer programmgesteuert erstellen. Die resultierenden Sitzungsberechtigungen sind eine Schnittmenge der auf der Identität des Benutzers oder der Rolle basierenden Richtlinien und der Sitzungsrichtlinien. Berechtigungen können auch aus einer ressourcenbasierten Richtlinie stammen. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen finden Sie unter Sitzungsrichtlinien im Benutzerhandbuch zu IAM.

Mehrere Richtlinientypen

Wenn mehrere auf eine Anforderung mehrere Richtlinientypen angewendet werden können, sind die entsprechenden Berechtigungen komplizierter. Informationen darüber, wie AWS bestimmt wird, ob eine Anfrage zulässig ist, wenn mehrere Richtlinientypen betroffen sind, finden Sie im IAMBenutzerhandbuch unter Bewertungslogik für Richtlinien.

Wie AWS HealthLake funktioniert mit IAM

Informieren Sie sich vor der Verwendung IAM zur Verwaltung des Zugriffs auf HealthLake, welche IAM Funktionen zur Verwendung verfügbar sind HealthLake.

IAMFunktionen, die Sie zusammen verwenden können AWS HealthLake

IAM-Feature	HealthLake Unterstützung
Identitätsbasierte Richtlinien	Ja
Ressourcenbasierte Richtlinien	Nein
Richtlinienaktionen	Ja
Richtlinienressourcen	Ja
Bedingungsschlüssel für die Richtlinie	Ja

IAM-Feature	HealthLake Unterstützung
ACLs	Nein
ABAC(Tags in Richtlinien)	Ja
Temporäre Anmeldeinformationen	Ja
Prinzipalberechtigungen	Ja
Servicerollen	Ja
Service-verknüpfte Rollen	Nein

Einen allgemeinen Überblick darüber, wie HealthLake und wie andere AWS Dienste mit den meisten IAM Funktionen funktionieren, finden Sie IAM im IAMBenutzerhandbuch unter AWS Dienste, die mit funktionieren.

Identitätsbasierte Richtlinien für AWS HealthLake

Unterstützt Richtlinien auf Identitätsbasis: Ja

Identitätsbasierte Richtlinien sind Dokumente mit JSON Berechtigungsrichtlinien, die Sie an eine Identität anhängen können, z. B. an einen IAM Benutzer, eine Benutzergruppe oder eine Rolle. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen einer identitätsbasierten Richtlinie finden Sie im Benutzerhandbuch unter Definieren benutzerdefinierter IAM Berechtigungen mit vom Kunden verwalteten Richtlinien. IAM

Mit identitätsbasierten IAM-Richtlinien können Sie angeben, welche Aktionen und Ressourcen zugelassen oder abgelehnt werden. Darüber hinaus können Sie die Bedingungen festlegen, unter denen Aktionen zugelassen oder abgelehnt werden. Sie können den Prinzipal nicht in einer identitätsbasierten Richtlinie angeben, da er für den Benutzer oder die Rolle gilt, dem er zugeordnet ist. Weitere Informationen zu allen Elementen, die Sie in einer JSON Richtlinie verwenden können, finden Sie in der Referenz zu IAM JSON Richtlinienelementen im IAMBenutzerhandbuch.

Beispiele für identitätsbasierte Richtlinien für AWS HealthLake

Beispiele für HealthLake identitätsbasierte Richtlinien finden Sie unter. Beispiele für identitätsbasierte Richtlinien für AWS HealthLake

Ressourcenbasierte Richtlinien finden Sie in AWS HealthLake

Unterstützt ressourcenbasierte Richtlinien: Nein

Ressourcenbasierte Richtlinien sind JSON Richtliniendokumente, die Sie an eine Ressource anhängen. Beispiele für ressourcenbasierte Richtlinien sind IAM Rollenvertrauensrichtlinien und Amazon S3 S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie einen Prinzipal angeben. Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services

Um kontoübergreifenden Zugriff zu ermöglichen, können Sie ein gesamtes Konto oder IAM-Entitäten in einem anderen Konto als Prinzipal in einer ressourcenbasierten Richtlinie angeben. Durch das Hinzufügen eines kontoübergreifenden Auftraggebers zu einer ressourcenbasierten Richtlinie ist nur die halbe Vertrauensbeziehung eingerichtet. Wenn sich der Prinzipal und die Ressource unterscheiden AWS-Konten, muss ein IAM Administrator des vertrauenswürdigen Kontos auch der Prinzipalentität (Benutzer oder Rolle) die Berechtigung zum Zugriff auf die Ressource erteilen. Sie erteilen Berechtigungen, indem Sie der juristischen Stelle eine identitätsbasierte Richtlinie anfügen. Wenn jedoch eine ressourcenbasierte Richtlinie Zugriff auf einen Prinzipal in demselben Konto gewährt, ist keine zusätzliche identitätsbasierte Richtlinie erforderlich. Weitere Informationen finden Sie IAMim IAMBenutzerhandbuch unter Kontenübergreifender Ressourcenzugriff.

Politische Maßnahmen für AWS HealthLake

Unterstützt Richtlinienaktionen: Ja

Administratoren können mithilfe von AWS JSON Richtlinien angeben, wer Zugriff auf was hat. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das `Action` Element einer JSON Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Richtlinienaktionen haben normalerweise denselben Namen wie der zugehörige AWS API Vorgang. Es gibt einige Ausnahmen, z. B. Aktionen, für die nur eine Genehmigung erforderlich ist und für die es keinen entsprechenden Vorgang gibt.

API Es gibt auch einige Operationen, die mehrere Aktionen in einer Richtlinie erfordern. Diese zusätzlichen Aktionen werden als abhängige Aktionen bezeichnet.

Schließen Sie Aktionen in eine Richtlinie ein, um Berechtigungen zur Durchführung der zugeordneten Operation zu erteilen.

Eine Liste der HealthLake Aktionen finden Sie unter Aktionen definiert von AWS HealthLake in der Serviceautorisierungsreferenz.

Bei Richtlinienaktionen wird vor der Aktion das folgende Präfix HealthLake verwendet:

```
healthlake
```

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie jede Aktion durch ein Komma.

```
"Action": [
      "healthlake:action1",
      "healthlake:action2"
         ]
```

Beispiele für HealthLake identitätsbasierte Richtlinien finden Sie unter. Beispiele für identitätsbasierte Richtlinien für AWS HealthLake

Politische Ressourcen für AWS HealthLake

Unterstützt Richtlinienressourcen: Ja

Administratoren können mithilfe von AWS JSON Richtlinien festlegen, wer Zugriff auf was hat. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das `Resource` JSON Richtlinienelement gibt das Objekt oder die Objekte an, für die die Aktion gilt. Anweisungen müssen entweder ein – `Resource`oder ein `NotResource`-Element enthalten. Es hat sich bewährt, eine Ressource mit ihrem Amazon-Ressourcennamen (ARN) anzugeben. Sie können dies für Aktionen tun, die einen bestimmten Ressourcentyp unterstützen, der als Berechtigungen auf Ressourcenebene bezeichnet wird.

Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, z. B. Auflistungsoperationen, einen Platzhalter (*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

```
"Resource": "*"
```

Eine Liste der HealthLake Ressourcentypen und ihrer ARNs Eigenschaften finden Sie unter Ressourcen definiert von AWS HealthLake in der Service Authorization Reference. Informationen zu den Aktionen, mit denen Sie die einzelnen Ressourcen spezifizieren können, finden Sie unter Aktionen definiert von AWS HealthLake. ARN

Beispiele für HealthLake identitätsbasierte Richtlinien finden Sie unter. Beispiele für identitätsbasierte Richtlinien für AWS HealthLake

Bedingungsschlüssel für Richtlinien für AWS HealthLake

Unterstützt servicespezifische Richtlinienbedingungsschlüssel: Ja

Administratoren können mithilfe von AWS JSON Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Condition` (oder `Condition` block) ermöglicht Ihnen die Angabe der Bedingungen, unter denen eine Anweisung wirksam ist. Das Element `Condition` ist optional. Sie können bedingte Ausdrücke erstellen, die Bedingungsoperatoren verwenden, z. B. ist gleich oder kleiner als, damit die Bedingung in der Richtlinie mit Werten in der Anforderung übereinstimmt.

Wenn Sie mehrere `Condition`-Elemente in einer Anweisung oder mehrere Schlüssel in einem einzelnen `Condition`-Element angeben, wertet AWS diese mittels einer logischen AND-Operation aus. Wenn Sie mehrere Werte für einen einzelnen Bedingungsschlüssel angeben, AWS wertet die Bedingung mithilfe einer logischen OR Operation aus. Alle Bedingungen müssen erfüllt werden, bevor die Berechtigungen der Anweisung gewährt werden.

Sie können auch Platzhaltervariablen verwenden, wenn Sie Bedingungen angeben. Beispielsweise können Sie einem IAM-Benutzer die Berechtigung für den Zugriff auf eine Ressource nur dann gewähren, wenn sie mit dessen IAM-Benutzernamen gekennzeichnet ist. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter IAMRichtlinienelemente: Variablen und Tags.

AWS unterstützt globale Bedingungsschlüssel und dienstspezifische Bedingungsschlüssel. Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter Kontext-Schlüssel für AWS globale Bedingungen im IAMBenutzerhandbuch.

Eine Liste der HealthLake Bedingungsschlüssel finden Sie unter Bedingungsschlüssel für AWS HealthLake in der Service Authorization Reference. Informationen zu den Aktionen und Ressourcen, mit denen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter Aktionen definiert von AWS HealthLake.

Beispiele für HealthLake identitätsbasierte Richtlinien finden Sie unter. Beispiele für identitätsbasierte Richtlinien für AWS HealthLake

Zugriffskontrolllisten () ACLs in AWS HealthLake

UnterstütztACLs: Nein

Zugriffskontrolllisten (ACLs) steuern, welche Principals (Kontomitglieder, Benutzer oder Rollen) über Zugriffsberechtigungen für eine Ressource verfügen. ACLsähneln ressourcenbasierten Richtlinien, verwenden jedoch nicht das JSON Richtliniendokumentformat.

Attributbasierte Zugriffskontrolle () mit ABAC AWS HealthLake

Unterstützt ABAC (Tags in Richtlinien): Ja

Die attributbasierte Zugriffskontrolle (ABAC) ist eine Autorisierungsstrategie, die Berechtigungen auf der Grundlage von Attributen definiert. In werden AWS diese Attribute als Tags bezeichnet. Sie können Tags an IAM Entitäten (Benutzer oder Rollen) und an viele AWS Ressourcen anhängen. Das Markieren von Entitäten und Ressourcen ist der erste Schritt vonABAC. Anschließend entwerfen Sie ABAC Richtlinien, die Operationen zulassen, wenn das Tag des Prinzipals mit dem Tag auf der Ressource übereinstimmt, auf die er zugreifen möchte.

ABACist hilfreich in Umgebungen, die schnell wachsen, und hilft in Situationen, in denen die Richtlinienverwaltung umständlich wird.

Um den Zugriff auf der Grundlage von Tags zu steuern, geben Sie im Bedingungselement einer Richtlinie Tag-Informationen an, indem Sie die Schlüssel `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, oder Bedingung `aws:TagKeys` verwenden.

Wenn ein Service alle drei Bedingungsschlüssel für jeden Ressourcentyp unterstützt, lautet der Wert für den Service Ja. Wenn ein Service alle drei Bedingungsschlüssel für nur einige Ressourcentypen unterstützt, lautet der Wert Teilweise.

Weitere Informationen dazu finden Sie ABAC unter Definieren von Berechtigungen mit ABAC Autorisierung im IAMBenutzerhandbuch. Ein Tutorial mit Schritten zur Einrichtung finden Sie im ABAC Benutzerhandbuch unter Verwenden der attributebasierten Zugriffskontrolle (ABAC). IAM

Verwenden temporärer Anmeldeinformationen mit AWS HealthLake

Unterstützt temporäre Anmeldeinformationen: Ja

Einige funktionieren AWS-Services nicht, wenn Sie sich mit temporären Anmeldeinformationen anmelden. Weitere Informationen, einschließlich Informationen darüber, AWS-Services wie Sie mit temporären Anmeldeinformationen arbeiten können AWS-Services , finden Sie IAM im IAMBenutzerhandbuch unter Diese Informationen.

Sie verwenden temporäre Anmeldeinformationen, wenn Sie sich mit einer anderen AWS Management Console Methode als einem Benutzernamen und einem Kennwort anmelden. Wenn Sie beispielsweise AWS über den Single Sign-On-Link (SSO) Ihres Unternehmens darauf zugreifen, werden bei diesem Vorgang automatisch temporäre Anmeldeinformationen erstellt. Sie erstellen auch automatisch temporäre Anmeldeinformationen, wenn Sie sich als Benutzer bei der Konsole anmelden und dann die Rollen wechseln. Weitere Informationen zum Rollenwechsel finden Sie im Benutzerhandbuch unter Von einem Benutzer zu einer IAM Rolle (Konsole) wechseln. IAM

Mit dem AWS CLI oder können Sie manuell temporäre Anmeldeinformationen erstellen AWS API. Sie können diese temporären Anmeldeinformationen dann für den Zugriff verwenden AWS. AWS empfiehlt, temporäre Anmeldeinformationen dynamisch zu generieren, anstatt langfristige Zugriffsschlüssel zu verwenden. Weitere Informationen finden Sie unter Temporäre Sicherheitsanmeldeinformationen unter IAM.

Serviceübergreifende Prinzipalberechtigungen für AWS HealthLake

Unterstützt Forward-Access-Sitzungen (FAS): Ja

Wenn Sie einen IAM Benutzer oder eine Rolle verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. Wenn Sie einige Dienste verwenden, führen Sie möglicherweise eine Aktion aus, die dann eine weitere Aktion in einem anderen Dienst auslöst. FASverwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, kombiniert mit der Anforderung, Anfragen AWS-Service an nachgeschaltete Dienste zu stellen. FASAnfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS-Services oder Ressourcen erforderlich sind. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien beim Stellen von FAS Anfragen finden Sie unter Zugriffssitzungen weiterleiten.

Servicerollen für AWS HealthLake

Unterstützt Servicerollen: Ja

Eine Servicerolle ist eine IAMRolle, die ein Dienst übernimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter Erstellen einer Rolle zum Delegieren von Berechtigungen AWS-Service an eine.

Informationen zu Servicerollen und den Inline-Richtlinien, die für den vollen Zugriff auf erforderlich sind AWS HealthLake, finden Sie unterBerechtigungen einrichten, um mit der Verwendung zu beginnen AWS HealthLake.

> **Warning**
>
> Das Ändern der Berechtigungen für eine Servicerolle kann zu HealthLake Funktionseinschränkungen führen. Bearbeiten Sie Servicerollen nur, HealthLake wenn Sie dazu eine Anleitung erhalten.

Dienstbezogene Rollen für AWS HealthLake

Unterstützt serviceverknüpfte Rollen: Ja

Eine dienstbezogene Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Dienstbezogene Rollen werden in Ihrem Dienst angezeigt AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für serviceverknüpfte Rollen anzeigen, aber nicht bearbeiten.

Einzelheiten zum Erstellen oder Verwalten von dienstbezogenen Rollen finden Sie unter AWS Dienste, die mit funktionieren. IAM Suchen Sie in der Tabelle nach einem Service mit einem `Yes` in der Spalte Service-linked role (Serviceverknüpfte Rolle). Wählen Sie den Link Yes (Ja) aus, um die Dokumentation für die serviceverknüpfte Rolle für diesen Service anzuzeigen.

Beispiele für identitätsbasierte Richtlinien für AWS HealthLake

Benutzer und Rollen haben standardmäßig nicht die Berechtigung, HealthLake-Ressourcen zu erstellen oder zu ändern. Sie können auch keine Aufgaben mithilfe von AWS Management Console, AWS Command Line Interface (AWS CLI) oder ausführen. AWS API Um Benutzern die Berechtigung zu erteilen, Aktionen mit den Ressourcen durchzuführen, die sie benötigen, kann ein

IAM Administrator IAM Richtlinien erstellen. Der Administrator kann dann die IAM Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen übernehmen.

Informationen zum Erstellen einer IAM identitätsbasierten Richtlinie mithilfe dieser Beispieldokumente zu JSON Richtlinien finden Sie im IAMBenutzerhandbuch unter IAM Richtlinien erstellen (Konsole).

Einzelheiten zu Aktionen und Ressourcentypen, die von definiert wurden HealthLake, einschließlich des Formats von ARNs für jeden der Ressourcentypen, finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für AWS HealthLake in der Service Authorization Reference.

Themen

- Bewährte Methoden für Richtlinien
- Verwenden der AWS HealthLake-Konsole
- Zugreifen auf einen AWS HealthLake Datenspeicher in Amazon Athena
- Benutzern die Berechtigung zur Anzeige eigener Berechtigungen erteilen

Bewährte Methoden für Richtlinien

Identitätsbasierte Richtlinien legen fest, ob jemand HealthLake Ressourcen in Ihrem Konto erstellen, darauf zugreifen oder sie löschen kann. Dies kann zusätzliche Kosten für Ihr verursachen AWS-Konto. Befolgen Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Anleitungen und Empfehlungen:

- Beginnen Sie mit AWS verwalteten Richtlinien und wechseln Sie zu Berechtigungen mit den geringsten Rechten — Verwenden Sie die AWS verwalteten Richtlinien, die Berechtigungen für viele gängige Anwendungsfälle gewähren, um Ihren Benutzern und Workloads zunächst Berechtigungen zu gewähren. Sie sind in Ihrem verfügbar. AWS-Konto Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom AWS Kunden verwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen finden Sie AWS im IAMBenutzerhandbuch unter AWS Verwaltete Richtlinien oder Verwaltete Richtlinien für Jobfunktionen.
- Berechtigungen mit den geringsten Rechten anwenden — Wenn Sie Berechtigungen mit IAM Richtlinien festlegen, gewähren Sie nur die Berechtigungen, die für die Ausführung einer Aufgabe erforderlich sind. Sie tun dies, indem Sie die Aktionen definieren, die für bestimmte Ressourcen unter bestimmten Bedingungen durchgeführt werden können, auch bekannt als die geringsten Berechtigungen. Weitere Informationen zur Verwendung IAM zum Anwenden von Berechtigungen finden Sie IAMim Benutzerhandbuch unter Richtlinien und Berechtigungen. IAM

- Verwenden Sie Bedingungen in IAM Richtlinien, um den Zugriff weiter einzuschränken — Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und Ressourcen einzuschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben, um anzugeben, dass alle Anfragen mit gesendet werden müssenSSL. Sie können auch Bedingungen verwenden, um Zugriff auf Serviceaktionen zu gewähren, wenn diese über einen bestimmten Zweck verwendet werden AWS-Service, z. AWS CloudFormation B. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter IAMJSONRichtlinienelemente: Bedingung.
- Verwenden Sie IAM Access Analyzer, um Ihre IAM Richtlinien zu validieren, um sichere und funktionale Berechtigungen zu gewährleisten. IAM Access Analyzer validiert neue und bestehende Richtlinien, sodass die Richtlinien der IAM Richtliniensprache (JSON) und den IAM bewährten Methoden entsprechen. IAM Access Analyzer bietet mehr als 100 Richtlinienprüfungen und umsetzbare Empfehlungen, um Sie bei der Erstellung sicherer und funktionaler Richtlinien zu unterstützen. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter Überprüfen von Richtlinien mit IAM Access Analyzer.
- Multi-Faktor-Authentifizierung erforderlich (MFA) — Wenn Sie ein Szenario haben, in dem IAM Benutzer oder ein Root-Benutzer erforderlich sind AWS-Konto, aktivieren Sie die Option MFA für zusätzliche Sicherheit. Wenn Sie festlegen möchten, MFA wann API Operationen aufgerufen werden, fügen Sie MFA Bedingungen zu Ihren Richtlinien hinzu. Weitere Informationen finden Sie unter Sicherer API Zugriff mit MFA im IAMBenutzerhandbuch.

Weitere Informationen zu bewährten Methoden finden Sie unter Bewährte Sicherheitsmethoden IAM im IAM Benutzerhandbuch. IAM

Verwenden der AWS HealthLake-Konsole

Um auf die AWS HealthLake Konsole zugreifen zu können, benötigen Sie ein Mindestmaß an Berechtigungen. Diese Berechtigungen müssen es Ihnen ermöglichen, Details zu den HealthLake Ressourcen in Ihrem aufzulisten und anzuzeigen AWS-Konto. Wenn Sie eine identitätsbasierte Richtlinie erstellen, die strenger ist als die mindestens erforderlichen Berechtigungen, funktioniert die Konsole nicht wie vorgesehen für Entitäten (Benutzer oder Rollen) mit dieser Richtlinie.

Sie müssen Benutzern, die nur Anrufe an AWS CLI oder am tätigen, keine Mindestberechtigungen für die Konsole gewähren AWS API. Erlauben Sie stattdessen nur den Zugriff auf die Aktionen, die dem API Vorgang entsprechen, den sie ausführen möchten.

Um vollen Zugriff auf zu erhalten HealthLake, fügen Sie einem IAM Benutzer oder einer Rolle die folgenden Richtlinien hinzu: `AmazonHealthLakeFullAccess`

und`AWSLakeFormationDataAdmin`. Sie müssen auch die HealthLake Inline-Richtlinie anhängen, bei der es sich um eine Servicerolle handelt. Eine Servicerolle ist eine IAMRolle, die ein Dienst übernimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter Erstellen einer Rolle zum Delegieren von Berechtigungen AWS-Service an eine. Informationen zur Inline-Richtlinie, mit der die erforderliche Servicerolle erstellt wird, finden Sie unterBerechtigungen einrichten, um mit der Verwendung zu beginnen AWS HealthLake. Sie müssen auch die AWS Lake Formation Konsole verwenden oder Ihrem HealthLake Administrator CLI die Position eines AWS Lake Formation Data Lake-Administrators zuweisen. Weitere Informationen finden Sie unter Berechtigungen einrichten, um mit der Verwendung zu beginnen AWS HealthLake.

Zugreifen auf einen AWS HealthLake Datenspeicher in Amazon Athena

Wenn Sie Benutzern und Rollen Zugriff auf die HealthLake Datenspeicher in gewähren möchten Amazon Athena, fügen Sie der Rolle oder dem Benutzer die folgenden IAM Richtlinien hinzu: `AmazonAthenaFullAccess` und`AmazonS3FullAccess`. `Select`und `Describe` Berechtigungen sind auch für Tabellen erforderlich, die von verwaltet werden AWS Lake Formation. AWS Lake Formation Tabellenberechtigungen werden von einem AWS Lake Formation Administrator in der AWS Lake Formation Konsole oder über die erteiltCLI. Weitere Informationen finden Sie unter Berechtigungen einrichten, um mit der Verwendung zu beginnen AWS HealthLake

Benutzern die Berechtigung zur Anzeige eigener Berechtigungen erteilen

In diesem Beispiel wird gezeigt, wie Sie eine Richtlinie erstellen, die IAM-Benutzern die Berechtigung zum Anzeigen der Inline-Richtlinien und verwalteten Richtlinien gewährt, die ihrer Benutzeridentität angefügt sind. Diese Richtlinie umfasst Berechtigungen zum Ausführen dieser Aktion auf der Konsole oder programmgesteuert mithilfe von oder. 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 verwaltete Richtlinien für AWS HealthLake

Eine AWS verwaltete Richtlinie ist eine eigenständige Richtlinie, die von erstellt und verwaltet wird AWS. AWS Verwaltete Richtlinien dienen dazu, Berechtigungen für viele gängige Anwendungsfälle bereitzustellen, sodass Sie damit beginnen können, Benutzern, Gruppen und Rollen Berechtigungen zuzuweisen.

Beachten Sie, dass AWS verwaltete Richtlinien für Ihre speziellen Anwendungsfälle möglicherweise keine Berechtigungen mit den geringsten Rechten gewähren, da sie allen AWS Kunden zur Verfügung stehen. Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie kundenverwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind.

Sie können die in AWS verwalteten Richtlinien definierten Berechtigungen nicht ändern. Wenn die in einer AWS verwalteten Richtlinie definierten Berechtigungen AWS aktualisiert werden, wirkt sich das Update auf alle Prinzidentitäten (Benutzer, Gruppen und Rollen) aus, denen die Richtlinie zugeordnet ist. AWS aktualisiert eine AWS verwaltete Richtlinie höchstwahrscheinlich, wenn eine neue Richtlinie eingeführt AWS-Service wird oder neue API Operationen für bestehende Dienste verfügbar werden.

Weitere Informationen finden Sie unter AWS verwaltete Richtlinien im IAM-Benutzerhandbuch.

AWS verwaltete Richtlinie: AmazonHealthLakeFullAccess

Die `AmazonHealthLakeFullAccess` Richtlinie bietet vollen Zugriff auf HealthLake. Wenn diese Richtlinie ihrem Benutzer oder ihrer Rolle zugewiesen ist, können Benutzer HealthLake damit auf Daten zugreifen, sie abfragen, importieren und exportieren HealthLake. Um viele allgemeine Aktionen in ausführen zu können HealthLake, müssen Sie dem Benutzer oder der Rolle zusätzliche Richtlinien hinzufügen. Weitere Informationen finden Sie unter HealthLake Operationen Berechtigungen einrichten, um mit der Verwendung zu beginnen AWS HealthLake und Berechtigungen.

Sie können die `AmazonHealthLakeFullAccess`-Richtlinie an Ihre IAM-Identitäten anfügen.

Diese Richtlinie gewährt Benutzern und Rollen *administrative and contributor* Berechtigungen, die es Benutzern und Rollen ermöglichen, Abfragen, Suchen HealthLake, Importieren und Exportieren durchzuführen, und ermöglicht es auch, Aktionen im Namen der Benutzer und Rollen auszuführen, die über diese Berechtigungen verfügen. HealthLake

Details zu Berechtigungen

Diese Richtlinie beinhaltet die folgende Aussage.

```
{
 "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 verwaltete Richtlinie: AmazonHealthLakeReadOnlyAccess

`AmazonHealthLakeReadOnlyAccess`Die Richtlinie gewährt Lesezugriff und Berechtigungen für HealthLake und verwandte Ressourcen in anderen AWS Diensten. Wenden Sie diese Richtlinie auf Benutzer an, denen Sie die Möglichkeit geben möchten, HealthLake Datenspeicher abzufragen und anzuzeigen, aber nicht die Möglichkeit, diese zu erstellen oder zu ändern.

Sie können die `AmazonHealthLakeReadOnlyAccess`-Richtlinie an Ihre IAM-Identitäten anfügen.

Diese Richtlinie gewährt *`read-only`* Berechtigungen, mit denen Benutzer und Rollen Abfragen durchführen können HealthLake.

Details zu Berechtigungen

Diese Richtlinie beinhaltet die folgende Aussage.

```
{
    "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 Operationen und Genehmigungen

In der folgenden Tabelle sind typische Operationen HealthLake und die zu ihrer Ausführung erforderlichen Berechtigungen aufgeführt.

HealthLake Operationen	Erforderliche Berechtigungen
Erstellen Sie einen Datenspeicher in HealthLake	`AmazonHealthLakeFullAccess` `AmazonLakeFormationDataAdmin` , Inline-Richtlinien und AWS Lake Formation Administratorberechtigungen, verwaltet von AWS Lake Formation
Löschen Sie einen Datenspeicher in HealthLake	`AmazonHealthLakeFullAccess` `AmazonLakeFormationDataAdmin` , Inline-Richtlinien und AWS Lake Formation Administratorberechtigungen, verwaltet von AWS Lake Formation
Einen Datenspeicher auflisten, durchsuchen oder abfragen in HealthLake	`AmazonHealthLakeReadOnlyAccess`
Einen Datenspeicher abfragen mit Amazon Athena	`AmazonAthenaFullAccess` `AmazonS3FullAccess` , AWS Lake Formation `Select` und `Describe` Berechtigungen für Tabellen, die verwaltet werden von AWS Lake Formation
Daten importieren von HealthLake	Siehe Berechtigungen für Importjobs einrichten.
Daten exportieren von HealthLake	Siehe Exportieren von Dateien aus Ihrem Datenspeicher (AWS SDKs).

HealthLake Aktualisierungen AWS verwalteter Richtlinien

Hier finden Sie Informationen zu Aktualisierungen AWS verwalteter Richtlinien HealthLake ab dem Zeitpunkt, zu dem dieser Dienst mit der Erfassung dieser Änderungen begann. Abonnieren Sie den RSS Feed auf der Seite HealthLake Dokumentenverlauf, um automatische Benachrichtigungen über Änderungen an dieser Seite zu erhalten.

Änderung	Beschreibung	Datum
AmazonHealthLakeFullAccess	`AmazonHealthLakeFullAccess` Richtlinie erforderlich, um vollen Zugriff auf zu gewähren HealthLake.	14. November 2022
AmazonHealthLakeReadOnlyAccess	`AmazonHealthLakeReadOnlyAccess` Für den schreibgeschützten Zugriff auf ist eine Richtlinie erforderlich. HealthLake	14. November 2022
HealthLake hat begonnen, Änderungen zu verfolgen	HealthLake hat begonnen, Änderungen für die AWS verwalteten Richtlinien zu verfolgen.	14. November 2022

Problembehandlung bei AWS HealthLake Identität und Zugriff

Verwenden Sie die folgenden Informationen, um häufig auftretende Probleme zu diagnostizieren und zu beheben, die bei der Arbeit mit HealthLake und auftreten könnenIAM.

Themen

- Ich bin nicht berechtigt, eine Aktion durchzuführen in AWS HealthLake
- Ich bin nicht berechtigt, iam auszuführen: PassRole
- Ich möchte Personen außerhalb meines AWS Kontos den Zugriff auf meine AWS HealthLake Ressourcen ermöglichen

Ich bin nicht berechtigt, eine Aktion durchzuführen in AWS HealthLake

Wenn Ihnen AWS Management Console mitgeteilt wird, dass Sie nicht berechtigt sind, eine Aktion durchzuführen, müssen Sie sich an Ihren Administrator wenden, um Unterstützung zu erhalten. Ihr Administrator ist die Person, die Ihnen Ihren Benutzernamen und Ihr Passwort bereitgestellt hat.

Der folgende Beispielfehler tritt auf, wenn der `mateojackson` IAM Benutzer versucht, die Konsole zu verwenden, um Details zu einer fiktiven *`my-example-widget`* Ressource anzuzeigen, aber nicht über die fiktiven `healthlake:`*`GetWidget`* Berechtigungen verfügt.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 healthlake:GetWidget on resource: my-example-widget
```

In diesem Fall bittet Mateo seinen Administrator um die Aktualisierung seiner Richtlinien, um unter Verwendung der Aktion *`my-example-widget`* auf die Ressource `healthlake:`*`GetWidget`* zugreifen zu können.

Ich bin nicht berechtigt, iam auszuführen: PassRole

Wenn Sie die Fehlermeldung erhalten, dass Sie nicht zum Durchführen der `iam:PassRole`-Aktion autorisiert sind, müssen Ihre Richtlinien aktualisiert werden, um eine Rolle an HealthLake übergeben zu können.

Einige AWS-Services ermöglichen es Ihnen, eine bestehende Rolle an diesen Dienst zu übergeben, anstatt eine neue Servicerolle oder eine dienstverknüpfte Rolle zu erstellen. Hierzu benötigen Sie Berechtigungen für die Übergabe der Rolle an den Dienst.

Der folgende Beispielfehler tritt auf, wenn ein IAM-Benutzer mit dem Namen `marymajor` versucht, die Konsole zu verwenden, um eine Aktion in HealthLake auszuführen. Die Aktion erfordert jedoch, dass der Service über Berechtigungen verfügt, die durch eine Servicerolle gewährt werden. Mary besitzt keine Berechtigungen für die Übergabe der Rolle an den Dienst.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In diesem Fall müssen die Richtlinien von Mary aktualisiert werden, um die Aktion `iam:PassRole` ausführen zu können.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich möchte Personen außerhalb meines AWS Kontos den Zugriff auf meine AWS HealthLake Ressourcen ermöglichen

Sie können eine Rolle erstellen, die Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation für den Zugriff auf Ihre Ressourcen verwenden können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. Für Dienste, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (ACLs) unterstützen, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- Informationen darüber, ob diese Funktionen HealthLake unterstützt werden, finden Sie unter. Wie AWS HealthLake funktioniert mit IAM
- Informationen dazu, wie Sie Zugriff auf Ihre Ressourcen gewähren können, AWS-Konten die Ihnen gehören, finden Sie im IAM Benutzerhandbuch unter Gewähren des Zugriffs auf einen anderen IAMBenutzer AWS-Konto , der Ihnen gehört.
- Informationen dazu, wie Sie Dritten Zugriff auf Ihre Ressourcen gewähren können AWS-Konten, finden Sie AWS-Konten im IAMBenutzerhandbuch unter Gewähren des Zugriffs für Dritte.
- Informationen dazu, wie Sie Zugriff über einen Identitätsverbund gewähren, finden Sie im Benutzerhandbuch unter Zugriff für extern authentifizierte Benutzer (Identitätsverbund). IAM
- Informationen zum Unterschied zwischen der Verwendung von Rollen und ressourcenbasierten Richtlinien für den kontenübergreifenden Zugriff finden Sie IAMim Benutzerhandbuch unter Kontoübergreifender Ressourcenzugriff. IAM

Protokollieren von AWS HealthLake API-Aufrufen mit AWS CloudTrail

AWS HealthLake ist in einen Dienst integriert AWS CloudTrail, der eine Aufzeichnung der Aktionen bereitstellt, die von einem Benutzer, einer Rolle oder einem AWS Dienst in ausgeführt wurden HealthLake. CloudTrail erfasst alle API Aufrufe HealthLake als Ereignisse. Zu den erfassten Aufrufen gehören Aufrufe von der HealthLake Konsole und Code-Aufrufe der HealthLake API Operationen. Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Bereitstellung von CloudTrail Ereignissen an einen Amazon S3 S3-Bucket aktivieren, einschließlich Ereignissen für HealthLake. Wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse trotzdem in der CloudTrail Konsole im Ereignisverlauf anzeigen. Anhand der von gesammelten Informationen können Sie die Anfrage ermitteln CloudTrail, an die die Anfrage gestellt wurde HealthLake, die IP-Adresse, von der

aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und weitere Details.

Weitere Informationen CloudTrail dazu finden Sie im AWS CloudTrail Benutzerhandbuch.

AWS HealthLake Informationen in CloudTrail

CloudTrail ist in Ihrem AWS Konto aktiviert, wenn Sie das Konto erstellen. Wenn Aktivitäten in auftreten HealthLake, wird diese Aktivität zusammen mit anderen AWS Serviceereignissen in der CloudTrail Ereignishistorie in einem Ereignis aufgezeichnet. Sie können die neusten Ereignisse in Ihr AWS -Konto herunterladen und dort suchen und anzeigen. Weitere Informationen finden Sie unter Ereignisse mit dem CloudTrail Ereignisverlauf anzeigen.

Für eine fortlaufende Aufzeichnung der Ereignisse in Ihrem AWS Konto, einschließlich der Ereignisse für HealthLake, erstellen Sie einen Trail. Ein Trail ermöglicht CloudTrail die Übermittlung von Protokolldateien an einen Amazon S3 S3-Bucket. Wenn Sie einen Trail in der Konsole anlegen, gilt dieser für alle AWS-Regionen. Der Trail protokolliert Ereignisse aus allen Regionen der AWS Partition und übermittelt die Protokolldateien an den von Ihnen angegebenen Amazon S3 S3-Bucket. Darüber hinaus können Sie andere AWS Dienste konfigurieren, um die in den CloudTrail Protokollen gesammelten Ereignisdaten weiter zu analysieren und darauf zu reagieren. Weitere Informationen finden Sie hier:

- Übersicht zum Erstellen eines Trails
- CloudTrail Unterstützte Dienste und Integrationen
- Konfiguration von SNS Amazon-Benachrichtigungen für CloudTrail
- Empfangen von CloudTrail Protokolldateien aus mehreren Regionen und Empfangen von CloudTrail Protokolldateien von mehreren Konten

Alle HealthLake Aktionen werden von protokolliert CloudTrail und sind in der HealthLake APIReferenz und in diesem Entwicklerhandbuch für Aktionen dokumentiert, die mit dem ausgeführt werden FHIR RESTAPI. Aufrufe der folgenden Aktionen generieren beispielsweise Einträge in den CloudTrail Protokolldateien:

- `DescribeFHIRImportJob`
- `DescribeFHIRExportJob`
- `StartFHIRImportJob`
- `ListFHIRImportJobs`

- `StartFHIRExportJob`
- `ListFHIRExportJobs`
- `CreateFHIRDatastore`
- `ListFHIRDatastores`
- `DeleteFHIRDatastore`
- `DescribeFHIRDatastore`
- `UpdateResource`
- `CreateResource`
- `DeleteResource`
- `ReadResource`
- `GetCapabilities`
- `SearchWithGet`
- `SearchWithPost`

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Die Identitätsinformationen unterstützen Sie bei der Ermittlung der folgenden Punkte:

- Ob die Anfrage mit Root- oder AWS Identity and Access Management (IAM) Benutzeranmeldedaten gestellt wurde.
- Gibt an, ob die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen Verbundbenutzer gesendet wurde.
- Ob die Anfrage von einem anderen AWS Dienst gestellt wurde.

Weitere Informationen finden Sie im CloudTrail userIdentityElement.

Grundlegendes zu AWS HealthLake Einträgen in Protokolldateien

Ein Trail ist eine Konfiguration, die die Übertragung von Ereignissen als Protokolldateien an einen von Ihnen angegebenen Amazon S3 S3-Bucket ermöglicht. CloudTrail Protokolldateien enthalten einen oder mehrere Protokolleinträge. Ein Ereignis stellt eine einzelne Anforderung aus einer beliebigen Quelle dar und enthält Informationen über die angeforderte Aktion, Datum und Uhrzeit der Aktion, Anforderungsparameter usw. CloudTrail Protokolldateien sind kein geordneter Stack-Trace der öffentlichen API Aufrufe, sodass sie nicht in einer bestimmten Reihenfolge angezeigt werden.

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag, der die `CreateFHIRDatastore` Aktion demonstriert.


```
{
    "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"
}
```


Überprüfung der Einhaltung der Vorschriften für AWS HealthLake

Externe Prüfer bewerten die Sicherheit und Einhaltung von Vorschriften im AWS HealthLake Rahmen mehrerer AWS Compliance-Programme. HealthLake Dazu gehörenHIPAA.

Informationen darüber, ob AWS-Service ein AWS-Services in den Geltungsbereich bestimmter Compliance-Programme fällt, finden Sie unter Umfang nach Compliance-Programm AWS-Services unter . Wählen Sie dort das Compliance-Programm aus, an dem Sie interessiert sind. Allgemeine Informationen finden Sie unter AWS Compliance-Programme AWS .

Sie können Prüfberichte von Drittanbietern unter herunterladen AWS Artifact. Weitere Informationen finden Sie unter Berichte herunterladen unter .

Ihre Verantwortung für die Einhaltung der Vorschriften bei der Nutzung AWS-Services hängt von der Vertraulichkeit Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften ab. AWS stellt die folgenden Ressourcen zur Verfügung, die Sie bei der Einhaltung der Vorschriften unterstützen:

- Compliance und Governance im Bereich Sicherheit – In diesen Anleitungen für die Lösungsimplementierung werden Überlegungen zur Architektur behandelt. Außerdem werden Schritte für die Bereitstellung von Sicherheits- und Compliance-Features beschrieben.

- Architecting for HIPAA Security and Compliance on Amazon Web Services — In diesem Whitepaper wird beschrieben, wie Unternehmen Anwendungen erstellen HIPAA können, die AWS für sie in Frage kommen.

  > **Note**
  >
  > Nicht alle sind berechtigt AWS-Services . HIPAA Weitere Informationen finden Sie in der Referenz für HIPAA qualifizierte Dienste.

- AWS Ressourcen zur AWS von Vorschriften — Diese Sammlung von Arbeitsmappen und Leitfäden kann auf Ihre Branche und Ihren Standort zutreffen.
- AWS Leitfäden zur Einhaltung von Vorschriften für Kunden — Verstehen Sie das Modell der gemeinsamen Verantwortung aus dem Blickwinkel der Einhaltung von Vorschriften. In den Leitfäden werden die bewährten Verfahren zur Sicherung zusammengefasst AWS-Services und die Leitlinien für Sicherheitskontrollen in verschiedenen Frameworks (einschließlich des National Institute of Standards and Technology (NIST), des Payment Card Industry Security Standards Council (PCI) und der International Organization for Standardization (ISO)) zusammengefasst.
- Evaluierung von Ressourcen anhand von Regeln im AWS Config Entwicklerhandbuch — Der AWS Config Service bewertet, wie gut Ihre Ressourcenkonfigurationen den internen Praktiken, Branchenrichtlinien und Vorschriften entsprechen.
- AWS Security Hub— Auf diese AWS-Service Weise erhalten Sie einen umfassenden Überblick über Ihren internen Sicherheitsstatus. AWS Security Hub verwendet Sicherheitskontrollen, um Ihre AWS -Ressourcen zu bewerten und Ihre Einhaltung von Sicherheitsstandards und bewährten Methoden zu überprüfen. Eine Liste der unterstützten Services und Kontrollen finden Sie in der Security-Hub-Steuerungsreferenz.
- Amazon GuardDuty — Dies AWS-Service erkennt potenzielle Bedrohungen für Ihre Workloads AWS-Konten, Container und Daten, indem es Ihre Umgebung auf verdächtige und böswillige Aktivitäten überwacht. GuardDuty kann Ihnen helfen, verschiedene Compliance-Anforderungen zu erfüllen PCIDSS, z. B. durch die Erfüllung der Anforderungen zur Erkennung von Eindringlingen, die in bestimmten Compliance-Frameworks vorgeschrieben sind.
- AWS Audit Manager— Auf diese AWS-Service Weise können Sie Ihre AWS Nutzung kontinuierlich überprüfen, um das Risikomanagement und die Einhaltung von Vorschriften und Industriestandards zu vereinfachen.

Resilienz in AWS HealthLake

Die AWS globale Infrastruktur basiert auf AWS Regionen und Availability Zones. AWS Regionen bieten mehrere physisch getrennte und isolierte Availability Zones, die über Netzwerke mit niedriger Latenz, hohem Durchsatz und hoher Redundanz miteinander verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Zonen ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Weitere Informationen zu AWS Regionen und Availability Zones finden Sie unter AWS Globale Infrastruktur.

Zusätzlich zur AWS globalen Infrastruktur HealthLake bietet es mehrere Funktionen, die Sie bei Ihren Anforderungen an Datenstabilität und Datensicherung unterstützen.

Sicherheit der Infrastruktur in AWS HealthLake

Als verwalteter Service AWS HealthLake ist er durch die AWS globalen Netzwerksicherheitsverfahren geschützt, die im Whitepaper Amazon Web Services: Sicherheitsprozesse im Überblick beschrieben sind.

Sie verwenden AWS veröffentlichte API Aufrufe, um HealthLake über das Netzwerk darauf zuzugreifen. Clients müssen Transport Layer Security (TLS) 1.0 oder höher unterstützen. Wir empfehlen TLS 1.2 oder höher. Kunden müssen außerdem Cipher Suites mit Perfect Forward Secrecy (PFS) wie Ephemeral Diffie-Hellman () oder Elliptic Curve Ephemeral Diffie-Hellman (DHE) unterstützen. ECDHE Die meisten modernen Systeme wie Java 7 und höher unterstützen diese Modi.

Außerdem müssen Anforderungen mit einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel signiert sein, der mit einem IAM-Prinzipal verknüpft ist. Alternativ können Sie mit AWS Security Token Service (AWS STS) temporäre Sicherheitsanmeldeinformationen erstellen, um die Anforderungen zu signieren.

Bewährte Methoden für die Sicherheit in AWS HealthLake

AWS HealthLake bietet eine Reihe von Sicherheitsfunktionen, die Sie bei der Entwicklung und Implementierung Ihrer eigenen Sicherheitsrichtlinien berücksichtigen sollten. Die folgenden bewährten Methoden sind allgemeine Richtlinien und keine vollständige Sicherheitslösung. Da diese

bewährten Methoden für Ihre Umgebung möglicherweise nicht angemessen oder ausreichend sind, sollten Sie sie als hilfreiche Überlegungen und nicht als bindend ansehen.

- Implementieren Sie den Zugriff mit den geringsten Rechten.
- Verwenden Sie nach Möglichkeit Customer-Managed-Keys (CMKs), um Ihre Daten zu verschlüsseln. Weitere Informationen finden Sie CMKs unter Amazon Key Management Service.
- Verwenden Sie Suchen mitPOST, nicht Suchen mit, GET wenn Sie nach PHI oder PII in Ihrem Datenspeicher suchen.
- Beschränken Sie den Zugriff auf sensible und wichtige Prüfungsfunktionen.
- Verwenden Sie beim Erstellen von Ressourcen durch das Update oder den Massenimport APIs nicht PHI oderPII, einschließlich der Namen von Datenspeichern und Aufträgen, in sichtbaren Feldern oder in der logischen FHIR ID (LID).
- Verwenden Sie beim Senden von Erstellungs-, Lese-, Aktualisierungs-, Lösch- oder Suchanfragen nicht PHI in der HTTP Kopfzeile.
- Aktivieren Sie AWS CloudTrail diese Option, um die AWS HealthLake Nutzung zu überwachen und sicherzustellen, dass keine unerwarteten Aktivitäten auftreten.
- Lesen Sie die Best Practices für die sichere Verwendung von Amazon S3 S3-Buckets. Weitere Informationen finden Sie unter Bewährte Sicherheitsmethoden im Amazon S3 S3-Benutzerhandbuch.

AWS HealthLake Endpunkte und Kontingente

Die folgenden Abschnitte enthalten Informationen zu AWS HealthLake Kontingenten und Endpunkten. Für anpassbare Kontingente können Sie über die Service Quotas-Konsole eine Erhöhung des Kontingents beantragen. Weitere Informationen finden Sie unter Beantragen einer Kontingenterhöhung im Service-Quotas-Benutzerhandbuch.

Service-Endpunkte

Die Tabelle zeigt die verfügbaren HealthLake Service-Endpunkte in einer bestimmten Region.

Name der Region	Region	Endpunkt	Protocol (Protokoll)	
USA Ost (Ohio)	us-east-2	healthlake.us-east-2.amazonaws.com	HTTPS	
		healthlake-fips.us-east-2.amazonaws.com	HTTPS	
USA Ost (Nord-Virginia)	us-east-1	healthlake.us-east-1.amazonaws.com	HTTPS	
		healthlake-fips.us-east-1.amazonaws.com	HTTPS	
USA West (Oregon)	us-west-2	healthlake.us-west-2.amazonaws.com	HTTPS	
		healthlake-fips.us-west-2.amazonaws.com	HTTPS	
Asien-Pazifik (Mumbai)	ap-south-1	healthlake.ap-south-1.amazonaws.com	HTTPS	
Asien-Pazifik (Sydney)	ap-southeast-2	healthlake.ap-southeast-2.amazonaws.com	HTTPS	
Europa (London)	eu-west-2	healthlake.eu-west-2.amazonaws.com	HTTPS	

Dienstkontingente für HealthLake

Im Folgenden sind die Standardkontingente für aufgeführt HealthLake.

Name	Standard	Anpas	Beschreibung
Anzahl der Zeichen in einer medizinischen Notiz	Jede unterstützte Region: 10 000	Nein	Die maximale Anzahl von Zeichen in einer einzelnen medizinischen Notiz innerhalb des DocumentReference Ressourcentyps (POST/PUTAnfragen).
Anzahl gleichzeitiger S tartFHIRImport Job-Jobs	Jede unterstützte Region: 1	Nein	Die maximale Anzahl gleichzeitiger tartFHIRImport S-Job-Jobs.
Anzahl der Jobs concurrentStart FHIRExportJob	Jede unterstützte Region: 1	Nein	Die maximale Anzahl gleichzeitiger tartFHIRExport S-Job-Jobs.
Anzahl der Datenspeicher pro Konto	Jede unterstützte Region: 10	Yes (Ja)	Die standardmäßige maximale Anzahl aktiver Datenspeicher pro Konto.
Anzahl der Dateien in einem tartFHIRImport S-Job	Jede unterstützte Region: 10 000	Nein	Die maximale Anzahl von Dateien in einem tartFHIRImport S-Job.
Anzahl der Ressourcen pro Paket	Jede unterstützte Region: 160	Nein	Die maximale Anzahl von Ressourcen, die in einer Bundle-Anfrage zulässig sind.
Rate der Bundle-Anfragen pro Konto	Jede unterstützte Region: 20	Ja	Die maximale Anzahl von POST Bundle-Anfragen,

Name	Standard	Anpas	Beschreibung
			die Sie pro Sekunde und Konto stellen können.
Rate der Bundle-Anfragen pro Datenspeicher	Jede unterstützte Region: 10	Yes (Ja)	Die maximale Anzahl von POST Bundle-Anfragen, die Sie pro Sekunde und Datenspeicher stellen können. Datenspeicher, die vor dem 21.8.2023 erstellt wurden, sind auf eine Anfrage pro Sekunde begrenzt.
Rate der Nutzung von ancelFHIRExport C-Jobanfragen DELETE pro Konto	Jede unterstützte Region: 1	Nein	Die maximale Anzahl von ancelFHIRExport C-Job-AnfragenDELETE, die Sie pro Minute und Konto stellen können.
Rate der reateFHIRDatastore C-Anfragen pro Konto	Jede unterstützte Region: 1	Nein	Die maximale Anzahl von reateFHIRDatastore C-Anfragen, die Sie pro Minute und Konto stellen können.
Rate der DELETE Anfragen pro Konto	Jede unterstützte Region: 2.000	Ja	Die maximale Anzahl von DELETE Anfragen, die Sie pro Sekunde und Konto stellen können.

Name	Standard	Anpas	Beschreibung
Rate der DELETE Anfragen pro Datenspeicher	Jede unterstützte Region: 1 000	Ja	Die maximale Anzahl von DELETE Anfragen, die Sie pro Sekunde pro Datenspeicher stellen können. Datenspeicher, die vor dem 21.8.2023 erstellt wurden, sind auf 100 Anfragen pro Sekunde begrenzt.
Rate der eleteFHIRDatastore D-Anfragen pro Konto	Jede unterstützte Region: 1	Nein	Die maximale Anzahl von eleteFHIRDatastore D-Anfragen, die Sie pro Minute und Konto stellen können.
Rate von escribeFHIRDatastore D-Anfragen pro Konto	Jede unterstützte Region: 10	Nein	Die maximale Anzahl von escribeFHIRDatastore D-Anfragen, die Sie pro Sekunde und Konto stellen können.
Rate der escribeFHIRExport D-Jobanfragen pro Konto	Jede unterstützte Region: 10	Nein	Die maximale Anzahl von escribeFHIRExport D-Job-Anfragen, die Sie pro Sekunde und Konto stellen können.
Rate der Nutzung von escribeFHIRExport D-Jobanfragen GET pro Konto	Jede unterstützte Region: 10	Nein	Die maximale Anzahl von escribeFHIRExport D-Job-AnfragenGET, die Sie pro Sekunde und Konto stellen können.

Name	Standard	Anpas	Beschreibung
Rate der escribeFHIRImport D-Jobanfragen pro Konto	Jede unterstützte Region: 10	Nein	Die maximale Anzahl von escribeFHIRImport D-Job-Anfragen, die Sie pro Sekunde und Konto stellen können.
Rate der Discovery-Anfragen pro Konto	Jede unterstützte Region: 10	Nein	Die maximale Anzahl von Discovery-Anfragen, die Sie pro Minute und Konto stellen können.
Rate der GET Anfragen pro Konto	Jede unterstützte Region: 6 000	Ja	Die maximale Anzahl von GET Anfragen, die Sie pro Sekunde und Konto stellen können.
Rate der GET Anfragen pro Datenspeicher	Jede unterstützte Region: 3 000	Ja	Die maximale Anzahl von GET Anfragen, die Sie pro Sekunde pro Datenspeicher stellen können. Datenspeicher, die vor dem 21.8.2023 erstellt wurden, sind auf 100 Anfragen pro Sekunde begrenzt.
Rate der GetCapabilities Anfragen pro Konto	Jede unterstützte Region: 10	Nein	Die maximale Anzahl von GetCapabilities Anfragen, die Sie pro Sekunde und Konto stellen können.
Rate von L istFHIRDatastores Anfragen pro Konto	Jede unterstützte Region: 10	Nein	Die maximale Anzahl von L istFHIRDatastores Anfragen, die Sie pro Sekunde und Konto stellen können.

Name	Standard	Anpas	Beschreibung
Rate der L istFHIRExport Jobs-Anfragen pro Konto	Jede unterstützte Region: 10	Nein	Die maximale Anzahl von L istFHIRExport Jobs-Anfragen, die Sie pro Sekunde und Konto stellen können.
Rate der L istFHIRImport Jobs-Anfragen pro Konto	Jede unterstützte Region: 10	Nein	Die maximale Anzahl von L istFHIRImport Jobs-Anfragen, die Sie pro Sekunde und Konto stellen können.
Rate der ListTagsforResource Anfragen pro Konto	Jede unterstützte Region: 10	Nein	Die maximale Anzahl von ListTagsforResourc e Anfragen, die Sie pro Sekunde und Konto stellen können.
Rate der POST Anfragen pro Konto	Jede unterstützte Region: 2.000	Ja	Die maximale Anzahl von POST Anfragen, die Sie pro Sekunde und Konto stellen können.
Rate der POST Anfragen pro Datenspei cher	Jede unterstützte Region: 1 000	Ja	Die maximale Anzahl von POST Anfragen, die Sie pro Sekunde pro Datenspeicher stellen können. Datenspeicher, die vor dem 21.8.2023 erstellt wurden, sind auf 100 Anfragen pro Sekunde begrenzt.

Name	Standard	Anpas	Beschreibung
Rate der PUT Anfragen pro Konto	Jede unterstützte Region: 2.000	Ja	Die maximale Anzahl von PUT Anfragen, die Sie pro Sekunde und Konto stellen können.
Rate der PUT Anfragen pro Datenspeicher	Jede unterstützte Region: 1 000	Ja	Die maximale Anzahl von PUT Anfragen, die Sie pro Sekunde pro Datenspeicher stellen können. Datenspeicher, die vor dem 21.8.2023 erstellt wurden, sind auf 100 Anfragen pro Sekunde begrenzt.
Rate der tartFHIRExport S-Job-Anfragen pro Konto	Jede unterstützte Region: 1	Nein	Die maximale Anzahl von S tartFHIRExport Job-Anfragen, die Sie pro Minute und Konto stellen können.
Rate der POST pro Konto verwendeten tartFHIRExport S-Jobanfragen	Jede unterstützte Region: 1	Nein	Die maximale Anzahl von S tartFHIRExport Job-AnfragenPOST, die Sie pro Minute und Konto stellen können.
Rate der tartFHIRImport S-Job-Anfragen pro Konto	Jede unterstützte Region: 1	Nein	Die maximale Anzahl von S tartFHIRImport Job-Anfragen, die Sie pro Minute und Konto stellen können.

Name	Standard	Anpas	Beschreibung
Rate der TagResource Anfragen pro Konto	Jede unterstützte Region: 10	Nein	Die maximale Anzahl von TagResource Anfragen, die Sie pro Sekunde stellen können.
Rate der UntagResource Anfragen pro Konto	Jede unterstützte Region: 10	Nein	Die maximale Anzahl von UntagResource Anfragen, die Sie pro Sekunde und Konto stellen können.
Rate der Suchanfragen, die GET pro Konto verwendet wurden	Jede unterstützte Region: 200	Ja	Die maximale Anzahl von Suchanfragen, GET die Sie pro Sekunde und Konto stellen können.
Rate der Suchanfragen, die GET pro Datenspeicher verwendet wurden	Jede unterstützte Region: 100	Yes (Ja)	Die maximale Anzahl von Suchanfragen, GET die Sie pro Sekunde pro Datenspeicher stellen können.
Rate der Suchanfragen, die POST pro Konto verwendet wurden	Jede unterstützte Region: 200	Ja	Die maximale Anzahl von Suchanfragen, POST die Sie pro Sekunde stellen können.
Rate der Suchanfragen, die POST pro Datenspeicher verwendet wurden	Jede unterstützte Region: 100	Yes (Ja)	Die maximale Anzahl von Suchanfragen, POST die Sie pro Sekunde pro Datenspeicher stellen können.

Name	Standard	Anpas	Beschreibung
Größe der einzelnen importierten Datei	Jede unterstützte Region: 5 Gigabyte	Nein	Die maximale Größe (in GB) einer einzelnen Datei, die in einem tartFHIRImport S-Job enthalten ist.
Gesamtgröße des Importauftrags	Jede unterstützte Region: 500 Gigabyte	Nein	Die maximale Größe (in GB) aller im Importauftrag enthaltenen Dateien.

Fehlerbehebung

Die folgende Dokumentation kann Ihnen bei der Behebung von Problemen helfen, die Sie möglicherweise bei der Verwendung AWS HealthLake haben.

Themen

Warum kann ich keinen HealthLake Datenspeicher erstellen?

Am 14. November 2022 HealthLake wurden die erforderlichen IAM Berechtigungen aktualisiert, die zum Erstellen eines neuen Datenspeichers erforderlich sind. Wenn Sie die Richtlinien für den Benutzer oder die Rolle, der zugreift, nicht aktualisiert HealthLake haben, wird die folgende Fehlermeldung angezeigt.

```
AccessDeniedException: Insufficient Lake Formation permission(s): Required Database on
 Catalog
```

Informationen zu den aktualisierten IAM Richtlinienanforderungen für die Erstellung eines Datenspeichers finden Sie unter AWS Verwaltete Richtlinie: AmazonHealthLakeFullAccess. step-by-stepAnweisungen zum Hinzufügen dieser Richtlinien zu Ihrem IAM Benutzer oder Ihrer Rolle finden Sie unterBerechtigungen einrichten, um mit der Verwendung zu beginnen AWS HealthLake.

Um einen Datenspeicher zu erstellen, benötigen Sie außerdem einen symmetrischen Schlüssel, der dem Kunden oder Amazon gehört. KMS Stellen Sie sicher, dass Sie in Ihrer Richtlinie über die richtigen Berechtigungen verfügen. IAM Weitere Informationen AWS KMS dazu finden Sie AWS Key Management Serviceim AWS Key Management Service Entwicklerhandbuch.

Die Anzahl der pro Konto zulässigen Datenspeicher wurde überschritten

HealthLake hat ein Kontingent von 10 Datenspeichern pro Konto. Um zu erfahren, wie Sie eine Kontingenterhöhung beantragen können, besuchen Sie das AWS Support Center.

Wie erstelle ich eine Autorisierung für den FHIR RESTfulAPIs?

Benutzer sollten den Signaturprozess von Signature Version 4 verwenden, um HealthLake API Anfragen, die über einen HTTP Client gesendet werden, zu authentifizieren. Weitere Informationen finden Sie unter Signaturprozess für Signature Version 4.

Um eine Sigv4-Autorisierung mit dem AWS SDK für Python zu erstellen, erstellen Sie ein Skript, das dem folgenden Beispiel ähnelt.

```
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())
```

Zusätzliche Informationen zur Verwendung der Sigv4-Autorisierung mit AWS SDK Python finden Sie im Thema Boto3-Anmeldeinformationen.

Meine Daten sind nicht im FHIR R4-Format — kann ich sie trotzdem verwenden? HealthLake

Nur FHIR R4-formatierte Daten können in einen HealthLake Datenspeicher importiert werden. Eine Liste der Partner, die Produkte anbieten, mit denen Benutzer ihre Daten transformieren können, finden Sie unter AWS HealthLake Partner.

Warum erhalte ich AccessDenied Fehler, wenn ich den FHIR RESTful APIs für einen Datenspeicher verwende, der mit einem vom Kunden verwalteten KMS Schlüssel verschlüsselt ist?

Damit ein Benutzer oder eine Rolle auf einen Datenspeicher zugreifen kann, sind sowohl Berechtigungen für den vom Kunden verwalteten Schlüssel als auch für IAM Richtlinien erforderlich. Ein Benutzer muss über die erforderlichen IAM Berechtigungen für die Verwendung eines vom Kunden verwalteten Schlüssels verfügen. Wenn ein Benutzer eine Erteilung, mit der die HealthLake Erlaubnis zur Verwendung des vom Kunden verwalteten KMS Schlüssels erteilt wurde, widerrufen oder zurückgezogen hat, HealthLake wird ein AccessDenied Fehler zurückgegeben.

HealthLake muss über die erforderlichen Berechtigungen verfügen, um auf Kundendaten zuzugreifen, neue FHIR Ressourcen, die in einen Datenspeicher importiert wurden, zu verschlüsseln und die FHIR Ressourcen zu entschlüsseln, wenn sie angefordert werden.

Weitere Informationen finden Sie unter Problembehandlung beim Schlüsselzugriff.

Warum ist mein Import fehlgeschlagen?

Bei einem erfolgreichen Importauftrag wird ein Ordner mit inputFileName .ndjson-Ausgabedateien generiert. Einzelne Datensätze können jedoch möglicherweise nicht importiert werden. In diesem Fall wird ein zweiter FAILURE Ordner mit einer Liste von Datensätzen generiert, die nicht importiert werden konnten. Das Auftragsausgabespeicherort für den Zugriff auf die Manifestdatei ist JobProperties. JobOutputDataConfig.S3-Konfiguration.S3Uri.

Diese Manifestdatei enthält Details zur Jobausgabe, z. B. den Speicherort aller erfolgreichen Antworten (. successOutput successOutputS3Uri), der Ort aller fehlgeschlagenen Antworten (. failureOutput failureOutputS3Uri) und zusätzliche Job-Metriken. Der Inhalt der Manifestdatei kann programmgesteuert analysiert werden. In der folgenden Beispiel-Manifestdatei sind die Eingabe- und Ausgabe-Buckets von Amazon S3 sowie Informationen über die Anzahl der gescannten Ressourcen und darüber, wie viele erfolgreich importiert wurden, aufgeführt.

```
            {
            "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
            }
```

Um zu analysieren, warum ein Importjob fehlgeschlagen istAPI, analysieren Sie den escribeFHIRImport D-Job JobProperties. Folgendes wird empfohlen:

- Wenn der Status lautet FAILED und eine Meldung vorhanden ist, hängen die Fehler damit zusammen, dass Jobparameter wie die Größe der Eingabedaten oder die Anzahl der Eingabedateien die HealthLake Kontingente überschreiten.
- Lautet der Status des COMPLETED Importauftrags WITH _ _ERRORS, finden Sie in der Manifestdatei Manifest.json Informationen darüber, welche Dateien nicht erfolgreich importiert wurden.
- Wenn der Status des Importauftrags lautet FAILED und keine Meldung vorhanden ist, wechseln Sie zum Ausgabeort des Jobs, um auf die Manifest-Datei Manifest.json zuzugreifen.

Für jede Eingabedatei gibt es eine Fehlerausgabedatei mit dem Namen der Eingabedatei für jede Ressource, die nicht importiert werden kann. Die Antworten enthalten die Zeilennummer (lineId), die der Position der Eingabedaten entspricht, das FHIR Antwortobjekt (UpdateResourceResponse) und den Statuscode (statusCode) der Antwort.

Eine Beispielausgabedatei würde wie folgt aussehen:

```
{"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}
```

Das Beispiel zeigt, dass in den Zeilen 3, 4, 7, 9, 15 in den entsprechenden Eingabezeilen der Eingabedatei Fehler aufgetreten sind. Für jede dieser Zeilen lauten die Erklärungen wie folgt:

- In Zeile 3 wird in der Antwort erklärt, dass die resourceType Angaben in Zeile 3 der Eingabedatei nicht gültig sind.
- In Zeile 5 wird in der Antwort erklärt, dass in Zeile 5 der Eingabedatei ein FHIR Überprüfungsfehler vorliegt.
- In Zeile 7 wird in der Antwort erklärt, dass ein Validierungsproblem bei der Eingabe resourceId vorliegt.
- In Zeile 9 wird in der Antwort erklärt, dass die Eingabedatei eine gültige Ressourcen-ID enthalten muss.
- In Zeile 15 lautet die Antwort der Eingabedatei, dass die Datei kein gültiges JSON Format hat.

Wie finde ich DocumentReference Ressourcen, die nicht verarbeitet werden konnten?

Wenn eine DocumentReference Ressource nicht gültig war, HealthLake wird anstelle der integrierten medizinischen Daten eine Erweiterung angezeigt, die auf einen Validierungsfehler NLP hinweist. Um DocumentReference Ressourcen zu finden, die bei der NLP Verarbeitung zu einem Validierungsfehler geführt haben, können Kunden die Suchfunktion mit dem Suchschlüssel cm-decoration-statusund dem Suchwert VALIDATION_ verwenden HealthLakeERROR. Diese Suche listet alle DocumentReference Ressourcen auf, die zu Validierungsfehlern geführt haben, zusammen

mit einer Fehlermeldung, die die Art des Fehlers beschreibt. Die Struktur des Erweiterungsfeldes in den DocumentReference Ressourcen mit Validierungsfehlern ähnelt dem folgenden Beispiel.

```
"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/"
          }
    ]
```

Ein VALIDATION_ ERROR kann auch vorkommen, wenn bei der NLP Dekoration mehr als 10.000 verschachtelte Objekte erstellt werden. In diesem Fall muss das Dokument vor der Verarbeitung in kleinere Dokumente aufgeteilt werden.

Migration eines vorhandenen Datenspeichers zur Verwendung von Amazon Athena

Datenspeicher, die vor dem 14. November 2022 erstellt wurden, funktionieren, können aber in Athena nicht abgefragt werden. SQL Um einen bereits vorhandenen Datenspeicher mit Athena abzufragen, müssen Sie ihn zunächst in einen neuen Datenspeicher migrieren.

Um Daten in einen neuen Datenspeicher zu migrieren

1. Erstellen Sie einen neuen Datenspeicher.
2. Exportieren Sie die Daten aus dem bereits vorhandenen in einen Amazon S3 S3-Bucket.
3. Importieren Sie die Daten aus dem Amazon S3 S3-Bucket in den neuen Datenspeicher.

Das Exportieren von Daten in einen Amazon S3 S3-Bucket ist mit einem Aufpreis verbunden. Die zusätzliche Gebühr hängt von der Größe der Daten ab, die Sie exportieren.

Suchergebnisse in Athena mit anderen AWS Diensten verbinden

Es kann zu Problemen kommen, wenn Sie Ihre Suchergebnisse von Athena mit anderen AWS Diensten teilen.

Ein Problem kann auftreten, wenn Sie es `json_extract[1]` als Teil einer SQL Suchabfrage verwenden.

Um dieses Problem zu beheben, müssen Sie auf aktualisieren`CATVAR`.

Dieses Problem tritt möglicherweise auf, wenn Sie versuchen, Speicherergebnisse, eine Tabelle (statisch) oder eine Ansicht (dynamisch) zu erstellen.

Die Athena-Konsole funktioniert nach dem Import von Daten in einen neuen Datenspeicher nicht

Nachdem Sie Daten in einen neuen Datenspeicher importiert haben, können die Daten möglicherweise nicht sofort verwendet werden. Dies dient dazu, genügend Zeit für die Aufnahme der Daten in die Eisbergtabellen zu haben. Versuchen Sie es zu einem späteren Zeitpunkt erneut.

Warum erhalte ich PutDataLakeSettings beim Hinzufügen eines neuen Data Lake-Administrators die Fehlermeldung Lake Formation Permissions: lakeformation:?

Wenn Ihr IAM Benutzer oder Ihre Rolle die `AWSLakeFormationDataAdmin` AWS verwaltete Richtlinie enthält, können Sie keine neuen Data Lake-Administratoren hinzufügen. Es wird eine Fehlermeldung mit folgendem Inhalt angezeigt:

```
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
```

Die AWS verwaltete Richtlinie `AdministratorAccess` ist erforderlich, um einen IAM Benutzer oder eine Rolle als AWS Lake Formation Data Lake-Administrator hinzuzufügen. Wenn Ihr IAM Benutzer oder Ihre Rolle auch `AWSLakeFormationDataAdmin` die Aktion enthält, schlägt sie fehl. Die `AWSLakeFormationDataAdmin` AWS verwaltete Richtlinie enthält eine ausdrückliche Ablehnung der API Operation AWS Lake Formation,`PutDataLakeSetting`.

Selbst Administratoren mit uneingeschränktem AWS Zugriff auf die `AdministratorAccess` AWS verwaltete Richtlinie können durch die `AWSLakeFormationDataAdmin` Richtlinie eingeschränkt werden.

Wie aktiviere ich die HealthLake integrierte Funktion zur Verarbeitung natürlicher Sprache?

Am 20. Februar 2023 hat sich das Standardverhalten von HealthLake Datenspeichern geändert.

Aktuelle Datenspeicher: Alle aktuellen HealthLake Datenspeicher werden die Verarbeitung natürlicher Sprache (NLP) für Base64-kodierte `DocumentReference` Ressourcen nicht mehr verwenden. Das bedeutet, dass neue `DocumentReference` Ressourcen nicht anhand von NLP Text im Ressourcentyp analysiert und keine neuen Ressourcen generiert werden. `DocumentReference` Bei vorhandenen `DocumentReference` Ressourcen NLP bleiben die über generierten Daten und Ressourcen erhalten, sie werden jedoch nach dem 20. Februar 2023 nicht mehr aktualisiert.

Neue Datenspeicher: HealthLake Datenspeicher, die nach dem 20. Februar 2023 erstellt wurden, führen keine Verarbeitung natürlicher Sprache (NLP) auf Base64-codierten `DocumentReference` Ressourcen durch.

Um diese Funktion zu aktivieren, müssen Sie einen Fall mit erstellen. AWS Support Center Console Um Ihren Fall zu erstellen, melden Sie sich bei Ihrem AWS-Konto an und wählen Sie dann Fall erstellen. Weitere Informationen zur Erstellung eines Falls und zur Fallverwaltung finden Sie im Support Benutzerhandbuch unter Erstellen von Supportanfragen und Fallmanagement.

Der Status meines Datenspeichers ändert sich im Vergleich zu Creating nicht

Wenn Sie versuchen, einen neuen HealthLake Datenspeicher zu erstellen, und Ihr Datenspeicherstatus sich nicht von Creating ändert, müssen Sie Athena aktualisieren, um den AWS Glue Data Catalog verwenden zu können.

Weitere Informationen finden Sie unter Upgrade auf den AWS Glue-Datenkatalog step-by-step im Amazon Athena Athena-Benutzerhandbuch.

Nach dem erfolgreichen Upgrade von können Sie jetzt einen Datenspeicher erstellen. AWS Glue Data Catalog

Um den alten Datenspeicher zu entfernen, erstellen Sie zunächst einen Fall mit AWS Support Center Console. Um Ihren Fall zu erstellen, melden Sie sich bei Ihrem AWS-Konto an und wählen Sie dann Fall erstellen. Weitere Informationen finden Sie im Support Benutzerhandbuch unter Erstellen von Supportanfragen und Fallmanagement.

Der Erstellungsstatus meines SDK Datenspeichers gibt eine Ausnahme oder einen unbekannten Status zurück

Bitte aktualisieren Sie Ihre Version SDK auf die neueste Version, wenn bei API Aufrufen des Listendatenspeichers oder des Describe-Datenspeichers eine Ausnahme oder ein unbekannter Datenspeicherstatus zurückgegeben wird.

Bei meinem FHIR POST API Vorgang mit einem 10-MB-Dokument HealthLake wird der Fehler 413Request Entity Too Large angezeigt.

AWS HealthLake hat ein synchrones Erstellungs- und API Aktualisierungslimit von 5 MB, um erhöhte Latenzen und Timeouts zu vermeiden.

Sie können große Dokumente (bis zu 164 MB) mithilfe der Binärdatei mithilfe des Massenimports aufnehmen. ResourceType API

Dokumentenverlauf für das AWS HealthLake Entwicklerhandbuch

In der folgenden Tabelle werden die Änderungen an der Dokumentation für AWS HealthLake Versionen beschrieben.

- APIVersion: neueste
- Letzte Aktualisierung der Dokumentation: 25.10.2024

Änderung	Beschreibung	Datum
HealthLake unterstützt FHIR `history` jetzt und interagiert `vread`	HealthLake unterstützt jetzt die FHIR `history` Interaktion zum Abrufen des Verlaufs einer bestimmten Ressource und die `vread` Interaktion zum Ausführen eines versionsspezifischen Lesevorgangs einer Ressource.	25. Oktober 2024
HealthLake unterstützt jetzt neue FHIR Suchparameter, Erweiterungen und Ressourcentypen.	HealthLake unterstützt jetzt neue FHIR Suchparameter, Erweiterungen und Ressourcentypen.	09. Dezember 2023
HealthLake unterstützt jetzt das SMART FHIR On-Framework	HealthLake unterstützt jetzt das Erstellen SMART auf FHIR aktivierten HealthLake Datenspeichern.	31. Mai 2023
HealthLake unterstützt jetzt die Profilvalidierung	HealthLake unterstützt jetzt die FHIR Profilvalidierung.	31. Mai 2023
HealthLake unterstützt jetzt `export`	HealthLake unterstützt jetzt das Exportieren von Dateien	31. Mai 2023

	mithilfe der FHIR REST API Operation`export`.	
Region Asien-Pazifik (Mumbai)	AWS HealthLake ist jetzt in der Region Asien-Pazifik (Mumbai) verfügbar.	4. April 2023
Die integrierte Verarbeitung natürlicher Sprache ist ausgeschaltet	HealthLake hat die integrierte Verarbeitung natürlicher Sprache (NLP) für alle Datenspeicher am 20. Februar 2023 deaktiviert.	20. Februar 2023
HealthLake lässt sich in Amazon Athena integrieren	Sie können Athena jetzt verwenden, um Datenspeicher abzufragen, die nach dem 14. November 2022 erstellt wurden.	14. November 2022
Die Gesamtgröße der Importaufträge wurde erhöht	Die maximale Gesamtgröße aller Dateien in einer tartFHIRImport S-Job-Anfrage beträgt jetzt 500 GB.	3. Oktober 2022
Bundle-Unterstützung	HealthLake unterstützt jetzt den Bundle-Ressourcentyp für die Aufnahme mehrerer Ressourcen.	5. August 2022
Aktualisierte Kontingente für CRUD Operationen in HealthLake	HealthLake unterstützt jetzt höhere Grenzwerte für CRUD Anfragen.	14. Juli 2022
Unterstützung einbeziehen	HealthLake unterstützt jetzt Abfragen `_include` im Datenspeicher.	14. Juli 2022
AWS HealthLake ist jetzt allgemein verfügbar	HealthLake ist jetzt allgemein verfügbar.	30. Juli 2020

AWS Glossar

Die neueste AWS Terminologie finden Sie im AWS Glossar in der AWS-Glossar Referenz.

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.