aws

개발자 안내서

AWS HealthLake

AWS HealthLake: 개발자 안내서

Table of Contents

AWS HealthLake란 무엇인가요?

AWS HealthLake 는 의료 상호 운용성FHIR(R4) 사양을 활용하여 임상 데이터 수집, 저장 및 분석에 HIPAA 적합한 서비스입니다.

> **Note**
>
> 2023년 2월 20일 이후 HealthLake 데이터 스토어는 기본적으로 통합 자연어 처리(NLP)를 사용하지 않습니다. 데이터 스토어에서이 기능을 활성화하려면 문제 해결 HealthLake의 통합 자연어 처리 기능을 켜려면 어떻게 해야 합니까? 장의 섹션을 참조하세요.

상태 데이터는 종종 불완전하고 일관되지 않습니다. 또한 임상 기록, 실험실 보고서, 보험 청구, 의료 이미지, 기록된 대화 및 시계열 데이터(예: 하트 ECG 또는 두뇌 EEG 트레이스)에 포함된 정보로 구조화되지 않은 경우가 많습니다.

의료 공급자는 HealthLake 를 사용하여 AWS 클라우드에 데이터를 저장, 변환, 쿼리 및 분석할 수 있습니다. HealthLake 통합된 의료 자연어 처리(NLP) 기능을 사용하여 다양한 소스의 비정형 임상 텍스트를 분석할 수 있습니다.는 자연어 처리 모델을 사용하여 비정형 데이터를 HealthLake 변환하고 강력한 쿼리 및 검색 기능을 제공합니다. HealthLake 를 사용하여 안전하고 규정을 준수하며 감사를 받을 수 있는 방식으로 환자 정보를 구성, 인덱싱 및 구성할 수 있습니다.

HealthLake 는 Amazon Athena 및 AWS Lake Formation과도 통합됩니다. 이 통합을 사용하여를 사용하여 데이터 스토어를 쿼리할 수 있습니다SQL.

의 이점 AWS HealthLake

를 사용하면 다음을 AWS HealthLake수행할 수 있습니다.

- 빠르고 쉽게 상태 데이터 수집 - 임상 정보, 실험실 보고서, 보험 청구 등을 포함한 온프레미스 Fast Healthcare 상호 운용성 리소스(FHIR) 파일을 Amazon Simple Storage Service(Amazon S3) 버킷으로 대량 가져올 수 있습니다. 그런 다음 다운스트림 애플리케이션 또는 워크플로에서 데이터를 사용할 수 있습니다.
- FHIR REST API 작업 사용 - FHIR REST API 작업을 사용하여 데이터 스토어에서 CRUD (Create/Read/Update/Delete) 작업을 수행할 수 있도록 HealthLake 지원합니다. FHIR 검색도 지원됩니다.

- 데이터를 감사할 수 있는 안전하고 자격 있는 방식으로 AWS 클라우드에 저장 HIPAA- 데이터를 FHIR 형식으로 저장할 수 있으므로 쉽게 쿼리할 수 있습니다.는 각 환자의 의료 기록에 대한 완전하고 시간적인 보기를 HealthLake 생성하고 이를 R4 FHIR 표준 형식으로 구성합니다.
- Athena 통합 - HealthLake의 Athena와의 통합을 통해 복잡한 필터 기준을 생성하고 저장하는 데 사용할 수 있는 강력한 SQL기반 쿼리를 생성할 수 있습니다. 그런 다음 SageMaker AI와 같은 다운스트림 애플리케이션에서이 데이터를 사용하여 기계 학습 모델을 훈련하거나 Amazon QuickSight 에서 대시보드 및 데이터 시각화를 생성할 수 있습니다.
- 특수 기계 학습(ML) 모델을 사용하여 비정형 데이터를 변환 - Amazon Comprehend Medical을 사용하여 통합된 의료 자연어 처리(NLP)를 HealthLake 제공합니다. 원시 의료 텍스트 데이터는 특수 ML 모델을 사용하여 변환됩니다. 이러한 모델은 비정형 의료 데이터에서 의미 있는 정보를 이해하고 추출하도록 훈련되었습니다. 통합 의료를 사용하면 의료 텍스트에서 엔터티(예: 의료 절차 및 약물), 엔터티 관계(예: 약물 및 용량), 엔터티 특성(예: 양성 또는 음성 테스트 결과 또는 절차 시간) 데이터를 자동으로 추출할 NLP수 있습니다. HealthLake 그런 다음 특성 징후, 증상 및 조건을 기반으로 새 리소스를 생성합니다. 이는 새 조건, 관찰 및 MedicationStatement 리소스 유형으로 추가됩니다.

HealthLake 사용 사례

다음 의료 애플리케이션에 HealthLake 를 사용할 수 있습니다.

- 인구 상태 관리 - 의료 조직이 인구 상태 추세, 결과 및 비용을 분석할 수 있도록 HealthLake 지원합니다. 이를 통해 조직은 환자 집단에 가장 적합한 개입을 식별하고 더 나은 관리 옵션을 선택할 수 있습니다.
- 진료 품질 개선 - 환자의 의학적 병력에 대한 전체 보기를 컴파일하여 HealthLake 병원, 건강 보험 회사 및 생명 과학 조직의 진료 격차를 줄이고, 진료 품질을 개선하며, 비용을 절감합니다.
- 병원 효율성 최적화 - 병원의 주요 분석 및 기계 학습 도구를 HealthLake 제공하여 효율성을 개선하고 병원 낭비를 줄입니다.

액세스 HealthLake

AWS Management Console, AWS Command Line Interface (AWS CLI) 또는를 HealthLake 통해에 액세스할 수 있습니다 AWS SDKs.

1. AWS Management Console -에 액세스하는 데 사용할 수 있는 웹 인터페이스를 제공합니다 HealthLake.

2. AWS Command Line Interface (AWS CLI) - Windows, macOS 및 Linux에서 지원되는 HealthLake 및를 포함한 광범위한 AWS 서비스에 대한 명령을 제공합니다. 설치에 대한 자세한 내용은 섹션을 AWS CLI참조하세요AWS Command Line Interface.
3. AWS SDKs –다양한 프로그래밍 언어 및 플랫폼SDKs(Java, Python, Ruby, ., NETiOS, Android 등)에 대한 라이브러리 및 샘플 코드로 구성된 (소프트웨어 개발 키트)를 AWS 제공합니다. 는 HealthLake 및에 대한 프로그래밍 방식의 액세스를 생성하는 편리한 방법을 SDKs 제공합니다 AWS. 자세한 내용은 for AWS Python을 참조SDK하세요.

HIPAA 자격 및 데이터 보안

이는 HIPAA 적격 서비스입니다. 1996년 AWS미국 건강보험 이전 및 책임에 관한 법률(HIPAA) 및 보호 대상 건강 정보를 처리, 저장 및 전송하기 위한 AWS 서비스(PHI)에 대한 자세한 내용은 HIPAA 개요를 참조하세요.

개인 식별 정보(PII)를 HealthLake 포함하는에 대한 연결은 암호화되어야 합니다. 기본적으로 HTTPS 에서 HealthLake 사용할 모든 연결은 암호화된 고객 콘텐츠를 TLS HealthLake 저장하고 AWS 공동 책임 원칙에 따라 작동합니다.

요금

HealthLake 요금에 대한 자세한 내용은 AWS HealthLake 요금 페이지를 참조하세요. 와 관련된 잠재적 비용을 더 잘 추정하려면 HealthLake 요금 계산기를 사용할 수 HealthLake있습니다.

AWS HealthLake 작동 방식

AWS HealthLake 는 의료 상호 운용성FHIR(R4) 사양을 활용하여 상태 레코드를 저장하는 데이터 스토어를 생성합니다. 를 사용하면 다음 작업을 수행할 HealthLake수 있습니다.

> **Note**
>
> 2023년 2월 20일 이후 HealthLake 데이터 스토어는 기본적으로 통합 자연어 처리(NLP)를 사용하지 않습니다. 데이터 스토어에서이 기능을 활성화하려면 문제 해결 HealthLake의 통합 자연어 처리 기능을 켜려면 어떻게 해야 합니까? 장의 섹션을 참조하세요.

- 데이터 스토어를 생성, 모니터링 및 삭제합니다.
- Amazon Simple Storage Service(Amazon S3) 버킷에서 데이터 스토어로 의료 데이터를 대량으로 가져오는 `StartFHIRImportJob` 데 사용합니다.
- 생성, 읽기, 업데이트 및 삭제(CRUD) 작업을 사용하여 데이터 스토어에 저장된 데이터를 관리합니다.
- Amazon AthenaSQL에서를 사용하여 데이터 스토어를 쿼리합니다.
- FHIR REST API 작업에서 HTTP 클라이언트를 사용하여 데이터 스토어를 검색합니다.
- Amazon Comprehend Medical API 작업을 활성화하여 자연어 처리()를 사용하여 데이터에서 의료 인사이트를 검색할 수 있습니다NLP.

데이터 스토어 생성 및 모니터링

를 사용하면 Fast Healthcare 상호 운용성 리소스(FHIR) 데이터를 저장할 HealthLake수 있는 데이터 스토어를 생성하고 모니터링할 수 있습니다.

새 데이터 스토어를 생성하려면 CreateFHIRDatastore 또는 HealthLake 콘솔을 사용할 수 있습니다. 데이터 스토어의 상태를 보려면 DescribeFHIRDatastore를 사용합니다. 여러 활성 데이터 스토어의 상태를 보려면 ListFHIRDatastores을 사용합니다. 데이터 스토어를 삭제하려면 DeleteFHIRDatastore를 사용합니다.

FHIR REST API 작업

FHIR REST API 작업을 사용하여 HealthLake 데이터 스토어에서 생성, 읽기, 업데이트, 삭제(CRUD) 작업을 수행할 수 있습니다. 가 FHIR REST API 작업을 HealthLake 지원하는 방법에 대한 자세한 내용은 섹션을 참조하세요 HealthLake 데이터 스토어와의 FHIR REST API 상호 작용 사용.

리소스 FHIR DocumentReference 확장에서 리소스 자동 생성

> **Note**
>
> HealthLake 데이터 스토어를 생성하고가 포함된 데이터를 추가`DocumentReference`하면 AWS 계정에 요금이 발생합니다. 자세한 내용은 AWS HealthLake 요금을 참조하세요.

HealthLake 는 `DocumentReference` 리소스 유형NLP에서 찾은 문서에를 제공합니다. 텍스트를 분석하려면 다음 Amazon Comprehend Medical API 작업을 HealthLake 사용합니다.

- `DetectEntitiesV2`: 임상 텍스트에 다양한 의료 개체가 있는지 검사하고 개체 범주, 위치 및 신뢰도 점수와 같은 특정 정보를 반환합니다.
- `InferICD10CM`: 임상 텍스트를 검사하여 환자 레코드에 나열된 엔터티로 의학적 상태를 감지하고 질병 제어 센터의 ICD-10-CM 지식 기반에서 정규화된 개념 식별자에 해당 엔터티를 연결합니다.
- `InferRxNorm`: 임상 텍스트를 검사하여 환자 레코드에 나열된 개체로 약물을 감지하고 National Library of Medicine에서 RxNorm 데이터베이스의 정규화된 개념 식별자에 연결합니다.

HealthLake 는 데이터 스토어에 추가될 때 `DocumentReference` 리소스 유형에서 발견된 데이터를 자동으로 분석합니다. 원본 `DocumentReference` 리소스 파일은 변경되지 않습니다. 추출된 의료 정보는 FHIR규정 준수 확장으로 자동으로 추가됩니다. 에서 NLP가 작동하는 방법에 대한 자세한 내용은 섹션을 HealthLake참조하세요에서 리소스 유형의 자연어 처리(NLP)를 기반으로 자동화된 FHIR DocumentReference 리소스 생성 사용 AWS HealthLake.

SQL기반 쿼리로 검색

> **Note**
>
> 2022년 11월 14일 이전에 생성된 데이터 스토어의 경우 검색은 FHIR REST API 작업으로 제한됩니다. HealthLake 데이터 스토어의 데이터에 SQL기반 쿼리를 사용하려면 섹션을 참조하세요Amazon AthenaSQL에서를 사용하여 AWS HealthLake 데이터 스토어 쿼리.

Amazon Athena는 서버리스 SQL기반 쿼리 서비스입니다. HealthLake 데이터 스토어는 Apache Iceberg 테이블로 Athena에 수집됩니다. 이러한 테이블은 대규모 분석 데이터 세트를 지원하도록 설계되었습니다. Athena에서 각 FHIR 리소스 유형은 테이블로 표시됩니다. Athena를 사용하면 데이터 스토어에서만 READ 요청할 수 있습니다. SQL기반 검색에 대한 자세한 내용은 섹션을 참조하세요를 사용하여 HealthLake 데이터 스토어 쿼리 SQL.

FHIR REST API 작업으로 검색

지원되는 검색 파라미터가 있는 리소스 유형을 지정하거나, 리소스 유형을 지정하지 않고 서버에 있는 리소스 ID를 사용하여 데이터 스토어에 저장된 상태 레코드를 검색할 수 있습니다. FHIR REST API 작업을 사용하여 검색하는 방법에 대한 자세한 내용은 섹션을 참조하세요 HealthLake 데이터 스토어와의 FHIR REST API 상호 작용 사용.

데이터 가져오기 작업

Amazon S3 버킷에서 파일을 대량으로 가져오는 AWS HealthLake 데 사용합니다. 콘솔 또는 StartFHIRImport작업을 사용하여 가져오기 작업을 시작합니다. 파일을 가져온 후 DescribeFHIRImport작업을 사용하여 작업 상태를 모니터링할 수 있습니다. 가져오기 작업이 완료되면 데이터를 Athena에 추가하거나 변환하거나 분석하여 다운스트림 애플리케이션에서 사용할 수 있습니다.

데이터 내보내기 작업

HealthLake 를 사용하여 파일을 Amazon S3 버킷으로 대량으로 내보냅니다. 콘솔 또는 StartFHIRExport작업을 사용하여 내보내기 작업을 시작합니다. 파일을 내보낸 후 DescribeFHIRExport작업을 사용하여 작업 상태를 모니터링하고 속성을 볼 수 있습니다. 내보내기 작업이 완료되면 Amazon을 사용하여 데이터를 시각화 QuickSight 하거나 다른 AWS 서비스를 사용하여 데이터에 액세스할 수 있습니다.

AWS HealthLake 지원되는 FHIR 프로필 검증

HealthLake 는 기본 FHIR R4 사양을 지원합니다. R4 사양에는 FHIR 프로파일이 포함되어 있습니다. 프로필은 FHIR 리소스 유형에서 기본 리소스 유형에 대한 제약 조건 및/또는 확장을 사용하여 보다 구체적인 리소스 유형 정의를 정의하는 데 사용됩니다. 예를 들어 FHIR 프로필은 확장 및 값 세트와 같은 필수 필드를 식별할 수 있습니다. 리소스는 여러 프로파일을 지원할 수 있습니다. 모든 HealthLake 데이터 스토어는 FHIR 프로파일 사용을 지원합니다.

HealthLake 데이터 스토어에 데이터를 추가할 때는 FHIR 프로필을 추가할 필요가 없습니다. 리소스가 추가되거나 업데이트될 때 FHIR 프로파일이 지정되지 않으면 리소스는 기본 FHIR R4 스키마에 대해서만 검증됩니다.

FHIR 리소스가 준수하는 프로필은에 수집되기 전에 리소스에 포함됩니다 HealthLake.는 HealthLake 데이터 스토어에 추가될 때 지정된 FHIR 프로필을 HealthLake 검증합니다.

FHIR 프로필은 구현 안내서에 지정되어 있습니다.는 다음 구현 안내서에 정의된 FHIR 프로필을 HealthLake 검증합니다.

에서 지원하는 FHIR 프로필 HealthLake

명칭	버전	구현 안내서	기능
미국 코어	3.1.1	http://hl7.org/fhir/us/core/STU3.1.1/	기본값
미국 코어	4.0.0	https://hl7.org/fhir/us/core/STU4/index.html	지원
CARIN 파란색 버튼	1.1.0	http://hl7.org/fhir/us/carin-bb/STU1.1/	기본값
CARIN 파란색 버튼	1.0.0	https://hl7.org/fhir/us/carin-bb/STU1/	지원
Da Vinci 지급인 데이터 교환	1.0.0	https://hl7.org/fhir/us/davinci-pdex/	기본값
Da Vinci Health Record Exchange(HRex)	0.2.0	https://hl7.org/fhir/us/davinci-hrex/2020Sep/	기본값

명칭	버전	구현 안내서	기능
DaVinci PDEX 플랜 넷	1.1.0	https://hl7.org/fhir/us/davinci-pdex-plan-net/STU1.1/	기본값
DaVinci PDEX 플랜 넷	1.0.0	https://hl7.org/fhir/us/davinci-pdex-plan-net/STU1/	지원
DaVinci 지급인 데이터 교환(PDex) 미국 의약품집	1.1.0	https://hl7.org/fhir/us/davinci-drug-formulary/STU1.1/	기본값
DaVinci 지급인 데이터 교환(PDex) 미국 의약품집	1.0.1	https://hl7.org/fhir/us/davinci-drug-formulary/STU1.0.1/	지원
National Health Authority의 Ayushman Bharat Digital Mission(ABDM)	2.0	https://www.nrces.in/ndhm/fhir/r4/index.html	기본값

리소스에 지정된 FHIR 프로필 검증

검증할 FHIR 프로필의 경우 구현 가이드에 URL 지정된 프로필을 사용하여 개별 리소스의 `profile` 요소에 추가합니다.

FHIR 프로필은 데이터 스토어에 새 리소스를 추가할 때 검증됩니다. 새 리소스를 추가하려면 StartFHIRImport작업 API 작업을 사용하거나, 새 리소스를 추가하도록 POST 요청하거나, 기존 리소스를 업데이트 PUT 하도록 할 수 있습니다.

Example - 리소스에서 참조되는 FHIR 프로파일을 확인하려면

프로필URL은 `"meta" : "profile"` 키-값 페어의 `profile` 요소에 추가됩니다. 이 리소스는 명확성을 위해 잘렸습니다.

```
        {
```

```
            "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 - 기본이 아닌 지원 FHIR 프로필을 참조하는 방법

지원되는 기본이 아닌 프로필(예: CarinBB 1.0.0) - 버전('|'로 구분)URL이 있는 프로파일과 URL `meta.profile` 요소의 기본 프로파일을 추가합니다. 이 예제 리소스는 명확성을 위해 잘렸습니다.

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

사전 로드된 데이터 유형

HealthLake 는 사전 로드된 데이터 유형SYNTHEA으로만 지원합니다. Synthea는 모델 생성 환자의 병력을 모델링하는 합성 환자 생성기입니다. 사용자가 실제 환자 데이터를 사용하지 않고 모델을 테스트 HealthLake 할 수 있도록가 FHIR R4-compliant 리소스 번들을 생성할 수 있는 오픈 소스 Git 리포지토리입니다.

다음 리소스 유형은 사전 로드된 데이터 스토어에서 사용할 수 있습니다.

지원되는 Synthea 리소스 유형

AllergyIntolerance	위치
CarePlan	MedicationAdministration
CareTeam	MedicationRequest
Claim	관측치
Condition	조직
장치	환자
DiagnosticReport	실무자
상황	PractitionerRole
ExplanationofBenefit	절차
ImagingStudy	증명
예방 접종	

사용을 시작할 권한 설정 AWS HealthLake

이 장에서는 AWS Management Console 를 사용하여 데이터 스토어 사용을 시작하고 AWS HealthLake 생성하는 데 필요한 권한을 설정합니다. 데이터 스토어를 생성할 수 있는 권한을 설정하려면 데이터 레이크 관리자 및 HealthLake 관리자인 IAM 사용자 또는 역할을 생성합니다. 이 사용자를 AWS Lake Formation의 데이터 레이크 관리자로 지정합니다. 데이터 레이크 관리자는 Amazon Athena를 사용하여 데이터 스토어를 쿼리하는 데 필요한 리소스에 대한 액세스 권한을 Lake Formation에 부여합니다.

에서 데이터 스토어를 생성한 후 파일을 데이터 스토어로 가져오거나 내보낼 HealthLake수 있는 권한을 설정할 수 있습니다. 파일 가져오기 권한 설정에 대한 자세한 내용은 섹션을 참조하세요가져오기 작업에 대한 권한 설정. 파일 내보내기 권한 설정에 대한 자세한 내용은 섹션을 참조하세요내보내기 작업에 대한 권한 설정.

주제

- 에 가입 AWS 계정
- 관리자 액세스 권한이 있는 사용자 생성
- 사용할 IAM 사용자 또는 역할 구성 HealthLake (IAM 관리자)
- Lake Formation에서 Data Lake 관리자로 사용자 또는 역할 추가(IAM 관리자)

에 가입 AWS 계정

가 없는 경우 다음 단계를 AWS 계정완료하여 생성합니다.

에 가입하려면 AWS 계정

1. https://portal.aws.amazon.com/billing/가입을 엽니다.
2. 온라인 지시 사항을 따릅니다.

   등록 절차 중 전화를 받고 전화 키패드로 확인 코드를 입력하는 과정이 있습니다.

   에 가입하면 AWS 계정AWS 계정 루트 사용자가 생성됩니다. 루트 사용자에게는 계정의 모든 AWS 서비스 및 리소스에 액세스할 권한이 있습니다. 보안 모범 사례는 사용자에게 관리 액세스 권한을 할당하고, 루트 사용자만 사용하여 루트 사용자 액세스 권한이 필요한 작업을 수행하는 것입니다.

AWS 는 가입 프로세스가 완료된 후 확인 이메일을 보냅니다. 언제든지 https://aws.amazon.com/로 이동하여 내 계정을 선택하여 현재 계정 활동을 보고 계정을 관리할 수 있습니다.

관리자 액세스 권한이 있는 사용자 생성

에 가입한 후 일상적인 작업에 루트 사용자를 사용하지 않도록 관리 사용자를 AWS 계정보호 AWS IAM Identity Center, AWS 계정 루트 사용자활성화 및 생성합니다.

보안 AWS 계정 루트 사용자

1. 루트 사용자를 선택하고 AWS 계정 이메일 주소를 입력하여 계정 소유자AWS Management Console로에 로그인합니다. 다음 페이지에서 비밀번호를 입력합니다.

   루트 사용자를 사용하여 로그인하는 데 도움이 필요하면 AWS 로그인 User Guide의 루트 사용자로 로그인을 참조하십시오.
2. 루트 사용자에 대해 다중 인증(MFA)을 켭니다.

   지침은 IAM 사용 설명서의 AWS 계정 루트 사용자(콘솔)에 대한 가상 MFA 디바이스 활성화를 참조하세요.

관리자 액세스 권한이 있는 사용자 생성

1. IAM Identity Center를 활성화합니다.

   지침은 AWS IAM Identity Center 사용 설명서의 AWS IAM Identity Center설정을 참조하세요.
2. IAM Identity Center에서 사용자에게 관리 액세스 권한을 부여합니다.

   를 자격 증명 소스 IAM Identity Center 디렉터리 로 사용하는 방법에 대한 자습서는 AWS IAM Identity Center 사용 설명서의 기본값으로 사용자 액세스 구성을 IAM Identity Center 디렉터리 참조하세요.

관리 액세스 권한이 있는 사용자로 로그인

- IAM Identity Center 사용자로 로그인하려면 IAM Identity Center 사용자를 생성할 때 이메일 주소로 전송URL된 로그인을 사용합니다.

  IAM Identity Center 사용자를 사용하여 로그인하는 방법에 대한 자세한 내용은 AWS 로그인 사용 설명서의 AWS 액세스 포털에 로그인을 참조하세요.

추가 사용자에게 액세스 권한 할당

1. IAM Identity Center에서 최소 권한 적용 모범 사례를 따르는 권한 세트를 생성합니다.

   지침은AWS IAM Identity Center 사용 설명서의 Create a permission set를 참조하세요.
2. 사용자를 그룹에 할당하고, 그룹에 Single Sign-On 액세스 권한을 할당합니다.

   지침은AWS IAM Identity Center 사용 설명서의 Add groups를 참조하세요.

사용할 IAM 사용자 또는 역할 구성 HealthLake (IAM 관리자)

> 페르소나: IAM 관리자
>
> 사용자 및 역할을 생성하고 데이터 레이크 관리자를 추가할 수 있는 IAM 사용자입니다.

이 주제의 다음 단계는 IAM 관리자가 수행해야 합니다.

HealthLake 데이터 스토어를 Athena에 연결하려면 데이터 레이크 관리자 및 HealthLake 관리자인 IAM 사용자 또는 역할을 생성해야 합니다. 이 새 사용자 또는 역할은 AWS Lake Formation을 통해 데이터 스토어에 있는 리소스에 대한 액세스 권한을 부여하며 관리`AmazonHealthLakeFullAccess` AWS 형 정책이 사용자 또는 역할에 추가되었습니다.

> Important
>
> 데이터 레이크 관리자인 IAM 사용자 또는 역할은 새 데이터 레이크 관리자를 생성할 수 없습니다. 데이터 레이크 관리자를 추가하려면 `AdministratorAccess` 액세스 권한이 부여된 IAM 사용자 또는 역할을 사용해야 합니다.

관리자를 생성하려면

1. 조직의 사용자 또는 역할에 **`AmazonHealthlakeFullAccess`** IAM AWS 관리형 정책을 추가합니다.

   IAM 사용자 생성에 익숙하지 않은 경우 IAM 사용 설명서의 IAM 사용자 생성 및 정책 개요를 AWS IAM 참조하세요.
2. IAM 사용자 또는 역할에 AWS Lake Formation에 대한 액세스 권한을 부여합니다.

- 조직의 사용자 또는 역할에 다음 IAM AWS 관리형 정책을 추가합니다. **AWSLakeFormationDataAdmin**

> **Note**
>
> 이 AWSLakeFormationDataAdmin 정책은 모든 AWS Lake Formation 리소스에 대한 액세스 권한을 부여합니다. 작업을 완료하는 데 필요한 최소 권한을 항상 사용하는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 IAM 모범 사례를 참조하세요.

3. 사용자 또는 역할에 다음 인라인 정책을 추가합니다. 자세한 내용은 IAM 사용 설명서의 인라인 정책을 참조하세요.

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

`AWSLakeFormationDataAdmin` 정책에 대한 자세한 내용은 Lake Formation 개발자 안내서의 Lake Formation 페르소나 및 IAM 권한 참조를 참조하세요. AWS

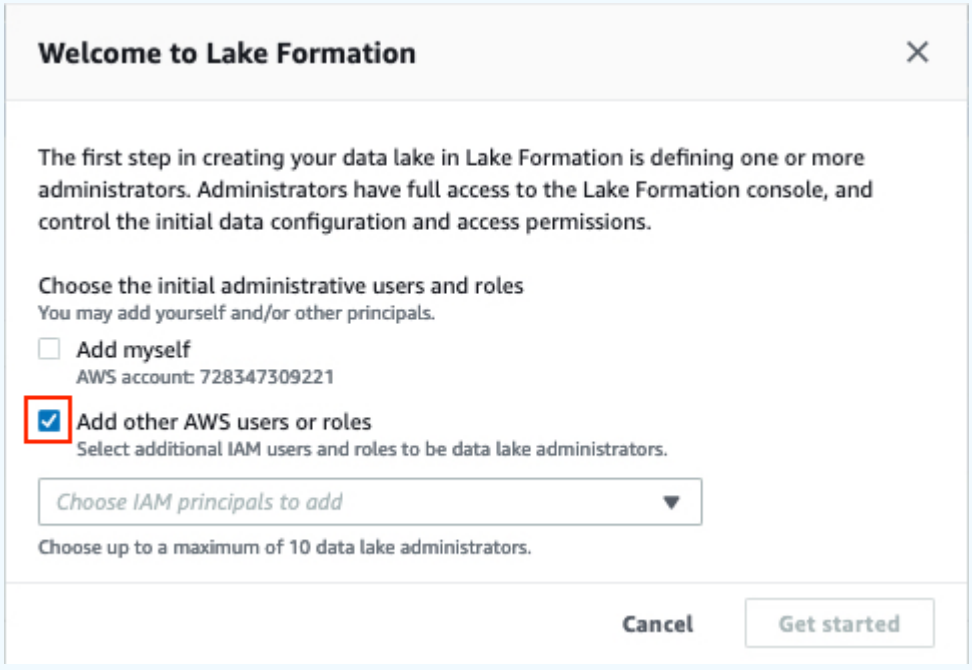
Lake Formation에서 Data Lake 관리자로 사용자 또는 역할 추가 (IAM 관리자)

다음으로 IAM 관리자는 Lake Formation에서 데이터 레이크 관리자로 1단계에서 생성한 사용자 또는 역할을 추가해야 합니다.

IAM 사용자 또는 역할을 데이터 레이크 관리자로 추가하려면

1. AWS Lake Formation 콘솔을 엽니다. https://console.aws.amazon.com/lakeformation/

> **Note**
>
> Lake Formation을 처음 방문하는 경우 Lake Formation 관리자를 정의하라는 Welcome to Lake Formation 대화 상자가 나타납니다.



2. AWS Lake Formation 데이터 레이크 관리자가 될 새 사용자 또는 역할을 할당합니다.
   - 옵션 1: Lake Formation 시작 대화 상자를 받은 경우
     1. 다른 AWS 사용자 또는 역할 추가를 선택합니다.
     2. 아래쪽 화살표(▾)를 선택합니다.

3. Lake Formation HealthLake 관리자로도 사용할 관리자를 선택합니다.
4. Get started를 선택합니다.

- 옵션 2: 탐색 창(")을 사용합니다.

1. 탐색 창(")을 선택합니다.
2. 권한에서 관리 역할 및 작업을 선택합니다.
3. 데이터 레이크 관리자 섹션에서 관리자 선택을 선택합니다.
4. 데이터 레이크 관리자 관리 대화 상자에서 아래쪽 화살표(▾)를 선택합니다.
5. 그런 다음 Lake Formation HealthLake 관리자이기도 한 관리자 사용자 또는 역할을 선택하거나 검색합니다.
6. 저장(Save)을 선택합니다.

3. Lake Formation에서 관리할 기본 보안 설정을 변경합니다. HealthLake 데이터 스토어 리소스는가 아닌 Lake Formation에서 관리해야 합니다IAM. 업데이트하려면 AWS Lake Formation 개발자 안내서의 기본 권한 모델 변경을 참조하세요.

에서 데이터 스토어 생성 AWS HealthLake

를 완료하면 데이터 스토어를 생성할 준비가 된 사용을 시작할 권한 설정 AWS HealthLake것입니다. 에서는 데이터 스토어를 AWS HealthLake사용하여 데이터를 HL7 FHIR (R4) 형식으로 저장합니다. 이 장의 주제에서는 데이터 스토어를 생성하는 방법을 설명합니다.

분석 지원 데이터 스토어를 생성하고 Athena에서 데이터 스토어에 대한 액세스 권한을 부여하려면 IAM 사용자, 그룹 또는 역할에 `AWSLakeFormationDataAdmin` 관리형 정책을 추가합니다. 이 `AWSLakeFormationDataAdmin` 정책을 사용하면 데이터 레이크 관리자를 생성하고 Athena의 데이터 스토어에 대한 액세스 권한을 부여할 수 있습니다. 권한 설정에 대한 자세한 내용은 섹션을 참조하세요사용을 시작할 권한 설정 AWS HealthLake.

HealthLake 도와 통합됩니다 AWS CloudTrail. CloudTrail 를 사용하여 사용자, 역할 또는 AWS 서비스 in HealthLake. CloudTrail Capture에서에 대한 모든 API 호출 및 콘솔 작업을 이벤트 HealthLake 로 캡처하는 서비스에서 수행한 작업의 레코드를 제공할 수 있습니다. 자세한 내용은 AWS CloudTrail을 사용하여 AWS HealthLake API 호출 로깅을 참조하십시오.

에서 지원하는 Fast Healthcare 상호 운용성 리소스(FHIR) 리소스 유형에 대한 자세한 내용은 섹션을 HealthLake참조하세요에서 지원되는 FHIR 리소스 유형 AWS HealthLake.

> **⚠ Amazon Athena 호환성**
>
> HealthLake 2022년 11월 14일 이전에 생성된 날짜 스토어는 Athena를 사용하여 SQL 쿼리를 수행할 수 없습니다. 기존 데이터 스토어에서 Athena 검색 기능을 사용하려면 먼저 데이터를 새 데이터 스토어로 마이그레이션합니다. 기존 데이터 스토어 마이그레이션에 대한 자세한 내용은 섹션을 참조하세요Amazon Athena를 사용하도록 기존 데이터 스토어 마이그레이션.

데이터 스토어를 생성한 후 API_DescribeFHIRDatastore 또는 API_ListFHIRDatastores.html API 작업을 사용하여 상태를 포함한 속성을 가져올 수 있습니다. 또는 HealthLake 콘솔의 데이터 스토어 페이지에서 데이터 스토어 상태 및 기타 세부 정보를 찾을 수 있습니다.

HealthLake 데이터 스토어의 상태는 다음과 같습니다.

- 생성 중 - 데이터 스토어가 생성 중입니다.
- 활성 - 데이터 스토어가 활성 상태입니다. 데이터를 가져오고 내보낼 수 있습니다. 데이터 스토어에 저장한 FHIR 리소스를 관리하고 검색할 수도 있습니다.

- 삭제 중 - 데이터 스토어가 삭제되고 있습니다.
- 삭제됨 - 데이터 스토어가 삭제되었습니다.

주제

데이터 스토어 생성(AWS Management Console)

> ⚠ HealthLake 콘솔 차이점
>
> HealthLake 콘솔은 FHIR 활성화된 데이터 스토어SMART에서 생성을 지원하지 않습니다. FHIR 활성화된 데이터 스토어SMART에서를 생성하려면 AWS CLI 또는 AWS 지원되는 중 하나를 사용해야 합니다SDKS. 자세한 내용은 SMART FHIR와의 통합 AWS HealthLake을 참조하십시오. 또한 콘솔은 개별 데이터 스토어의 세부 정보 페이지를 볼 HealthLake 때에서 지원하는 두 데이터 스토어 유형을 구분하지 않습니다.

HealthLake 데이터 스토어를 생성하려면

1. https://console.aws.amazon.com//healthlake/집에서 HealthLake 콘솔을 엽니다.
2. 탐색 창(™)을 엽니다.
3. 그런 다음 데이터 스토어를 선택합니다.
4. 그런 다음 데이터 스토어 생성을 선택합니다.
5. 데이터 스토어 설정 섹션에서 데이터 스토어 이름에 이름을 지정합니다.
6. (선택 사항) 데이터 스토어 설정 섹션의 샘플 데이터 사전 로드에서 Synthea 데이터를 사전 로드할 확인란을 선택합니다.
   - Synthea 데이터는 사전 로드된 샘플 데이터 세트입니다. 자세한 내용은 사전 로드된 데이터 유형 단원을 참조하십시오.
7. 데이터 스토어 암호화 섹션에서 AWS 소유 키 사용(기본값) 또는 다른 AWS KMS 키 선택(고급)을 선택합니다.
8. 태그 - 선택 사항 섹션에서 데이터 스토어에 태그를 추가할 수 있습니다.
   - 데이터 스토어 태그 지정에 대한 자세한 내용은 섹션을 참조하세요데이터 스토어에 태그 추가.

9. 그런 다음 데이터 스토어 생성을 선택합니다. 데이터 스토어의 상태는 데이터 스토어 페이지에서 확인할 수 있습니다.

데이터 스토어 생성(AWS CLI 및 AWS SDKs)

다음 코드 예제를 사용하여 HealthLake 데이터 스토어를 생성할 수 있습니다.

AWS CLI

다음 예제는 AWS CLI로 `CreateFHIRDatastore` 작업을 사용하는 방법을 보여 줍니다. 이 예제를 실행하려면 AWS CLI를 설치해야 합니다. 데이터 스토어를 생성할 때 달리 지정하지 않는 한 저장 시 암호화는 기본적으로 AWS소유 KMS 키로 설정됩니다. 의 암호화에 대한 자세한 내용은 섹션을 HealthLake 참조REST하세요에서에 REST 대한 암호화 AWS HealthLake.

다음은 Unix, Linux, macOS용 형식으로 지정된 예제입니다. Windows의 경우 각 줄 끝에 있는 백슬래시(\) Unix 연속 문자를 캐럿()으로 바꿉니다^.

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

성공하면 다음과 같은 JSON 응답을 받게 됩니다. 데이터 스토어가 데이터를 수집할 준비가 되면 상태가 로 변경됩니다`ACTIVE`. HealthLake 데이터 스토어로 데이터를 가져오는 방법에 대한 자세한 내용은 섹션을 참조하세요 HealthLake 데이터 스토어로 파일 가져오기.

```
{
    "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/"
}
```

모든 데이터 스토어의 목록을 보려면 `ListFHIRDataStore` 작업을 사용할 수 있습니다. HealthLake 콘솔에서 활성 데이터 스토어 목록을 볼 수도 있습니다.

Python (boto3)

다음 예제에서는 `create_fhir_datastore` 작업을 사용하여 HealthLake 데이터 스토어를 생성하는 방법을 보여줍니다. 데이터 스토어 저장 시 암호화를 생성하면 달리 지정되지 않는 한 기본적으로 AWS소유 AWS KMS 키가 설정됩니다. 의 암호화에 대한 자세한 내용은 섹션을 HealthLake 참조REST하세요에서에 REST 대한 암호화 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')
```

데이터 스토어는 네 가지 상태 중 하나를 가질 수 있습니다. 상태에 관계없이 HealthLake 데이터 스토어 목록을 보는 `list_fhir_datastores` 데 사용합니다. 이 예제에서는 데이터 스토어의 상태를 기반으로 필터링하는 방법을 보여줍니다.

```
import boto3

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

자세한 내용은 Boto3 설명서`list_fhir_datastore`의 섹션을 참조하세요.

HealthLake 데이터 스토어로 파일 가져오기

를 완료한 후 Amazon Simple Storage Service(Amazon S3) 버킷에서 데이터 스토어로 파일을 가져올 에서 데이터 스토어 생성 AWS HealthLake수 있습니다. 파일을 가져오려면 HealthLake 콘솔 또는 작업을 사용하여 가져오기 `StartFHIRImportJob` API 작업을 시작합니다.

가져오기 작업을 생성할 때 Amazon S3의 입력 데이터 위치, 출력 로그 파일의 Amazon S3 버킷 위치, 버킷에 대한 HealthLake 액세스 권한을 부여하는 IAM 역할 및 고객 소유 또는 AWS 소유 AWS Key Management Service 키를 지정합니다.는이 키를 HealthLake 사용하여 소스 위치에서 데이터를 암호화하고가 HealthLake 가져올 수 있도록 복호화하는 데 사용됩니다. 가져오기 작업에 대한 권한 설정에 대한 자세한 내용은 섹션을 참조하세요가져오기 작업에 대한 권한 설정. AWS KMS 키 생성 및 사용에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 키 생성을 참조하세요.

HealthLake 는 줄 바꿈으로 구분된JSON(`.ndjson`) 형식의 입력 파일을 수락합니다. 여기서 각 줄은 유효한 FHIR 리소스로 구성됩니다. `DescribeFHIRImportJob` 및 API 작업을 사용하여 진행 중인 가져오기 작업을 설명하고 나열`ListFHIRImportJobs`할 수 있습니다.

각 가져오기 작업에 대해 `manifest.json` 파일을 HealthLake 생성합니다. 이 로그는 가져오기 작업의 성공과 실패를 모두 설명합니다.는 가져오기 작업을 생성할 때 지정한 Amazon S3 버킷으로 파일을 HealthLake 출력합니다. 자세한 내용은 매니페스트 JSON 파일 단원을 참조하십시오.

가져오기 또는 내보내기 작업을 대기열에 넣을 수 있습니다. 이러한 비동기 가져오기 또는 내보내기 작업은 FIFO (선입선출) 방식으로 처리됩니다. 가져오기 또는 내보내기 작업이 진행되는 동안 FHIR 리소스를 생성, 읽기, 업데이트 또는 삭제할 수 있습니다.

데이터 스토어를 미리 로드된 데이터로 채우거나 데이터를 가져온 후 Amazon AthenaSQL에서를 사용하여 데이터 스토어 쿼리를 시작할 수 있습니다. 자세한 내용은 Amazon AthenaSQL에서를 사용하여 AWS HealthLake 데이터 스토어 쿼리 단원을 참조하십시오.

주제

- 가져오기 작업에 대한 권한 설정
- 에서 가져오기 작업 시작 HealthLake
- 매니페스트 JSON 파일
- 예:를 사용하여 가져오기 작업 시작 및 모니터링 AWS CLI

가져오기 작업에 대한 권한 설정

파일을 데이터 스토어로 가져오기 전에 Amazon S3의 입력 및 출력 버킷에 액세스할 수 있는 HealthLake 권한을 부여해야 합니다. HealthLake 액세스 권한을 부여하려면에 대한 IAM 서비스 역할을 생성하고 HealthLake, 역할에 신뢰 정책을 추가하여 HealthLake 수임 역할 권한을 부여하고, 역할에 권한 정책을 연결하여 Amazon S3 버킷에 대한 액세스 권한을 부여합니다.

가져오기 작업을 생성할 때에 대해이 역할의 Amazon 리소스 이름(ARN)을 지정합니다DataAccessRoleArn. IAM 역할 및 신뢰 정책에 대한 자세한 내용은 IAM 역할을 참조하세요.

권한을 설정한 후에는 가져오기 작업을 사용하여 파일을 데이터 스토어로 가져올 준비가 되었습니다. 자세한 내용은 에서 가져오기 작업 시작 HealthLake 단원을 참조하십시오.

가져오기 권한을 설정하려면

1. 아직 생성하지 않은 경우 출력 로그 파일에 대한 대상 Amazon S3 버킷을 생성합니다. Amazon S3 버킷은 서비스와 동일한 AWS 리전에 있어야 하며 모든 옵션에 대해 퍼블릭 액세스 차단을 활성화해야 합니다. 자세한 내용은 Amazon S3 퍼블릭 액세스 차단 사용을 참조하세요. Amazon 소유 또는 고객 소유 KMS 키도 암호화에 사용해야 합니다. KMS 키 사용에 대한 자세한 내용은 Amazon Key Management Service를 참조하세요.
2. 에 대한 데이터 액세스 서비스 역할을 생성하고 HealthLake 서비스에 다음 신뢰 정책을 사용하여 이를 수임할 수 있는 권한을 HealthLake 부여합니다.는 이를 HealthLake 사용하여 출력 Amazon S3 버킷을 씁니다.

```
{
    "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. Amazon S3 버킷에 액세스할 수 있도록 허용하는 권한 정책을 데이터 액세스 역할에 추가합니다. `amzn-s3-demo-bucket`를 버킷 이름으로 바꿉니다.

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

에서 가져오기 작업 시작 HealthLake

데이터 스토어를 생성하고 가져오기 작업(가져오기 작업에 대한 권한 설정)에 대한 권한을 설정한 후 가져오기 작업을 사용하여 파일 가져오기를 시작할 수 있습니다. AWS HealthLake 콘솔 또는 가져오기 API, start-fhir-import-job API를 사용하여 AWS HealthLake 가져오기 작업을 시작할 수 있습니다.

주제

- API 작업을 사용하여 파일 가져오기
- 가져오기 작업 시작(콘솔)

API 작업을 사용하여 파일 가져오기

사전 조건

작업을 사용할 AWS HealthLake API 때는 먼저 AWS Identity and Access Management (IAM) 정책을 생성하고 IAM 역할에 연결해야 합니다. IAM 역할 및 신뢰 정책에 대한 자세한 내용은 IAM 정책 및 권한을 참조하세요. 또한 고객은 암호화에 KMS 키를 사용해야 합니다. KMS 키 사용에 대한 자세한 내용은 Amazon Key Management Service를 참조하세요.

파일(API)을 가져오려면 다음 단계를 사용합니다.

1. Amazon S3 버킷에 데이터를 업로드합니다.
2. `start-fhir-import-job API` API 작업을 사용합니다. 작업을 시작할 때 입력 파일이 포함된 Amazon S3 버킷의 이름, 암호화에 사용할 KMS 키 및 출력 데이터 구성을 지정합니다.
3. FHIR 가져오기 작업에 대해 자세히 알아보려면 describe-fhir-import-job 작업을 사용하여 작업의 ID, ARN, 이름, 시작 시간, 종료 시간 및 현재 상태를 가져옵니다. list-fhir-import-job를 사용하여 모든 가져오기 작업과 해당 상태를 표시합니다.

가져오기 작업 시작(콘솔)

콘솔을 사용하여 파일을 가져오려면 Amazon S3 버킷에 데이터를 업로드합니다.

파일을 가져오려면 다음 단계를 따릅니다.

1. Amazon S3 버킷에 데이터를 업로드합니다.
2. https://console.aws.amazon.com//healthlake/집에서 HealthLake 콘솔을 엽니다.
3. 데이터 스토어의 데이터 스토어 세부 정보 페이지로 이동하여 가져오기를 선택합니다.

4. Amazon S3 버킷을 지정하고 사용할 IAM 역할과 KMS 키를 생성하거나 식별합니다.
5. 데이터 가져오기를 선택합니다.

매니페스트 JSON 파일

각 가져오기 작업에 대해 `manifest.json` 파일을 HealthLake 생성합니다. 가져오기 작업을 생성할 때 지정한 Amazon S3 버킷으로 파일을 HealthLake 출력합니다.

`manifest.json` 파일은 가져오기 작업의 성공과 실패를 모두 설명합니다. 로그 파일은 SUCCESS 및 라는 두 개의 폴더로 구성됩니다FAILURE. 출력 파일에는 민감한 정보가 포함될 수 있으므로 가져오기 작업을 생성할 때 출력 Amazon S3 버킷과 암호화용 AWS KMS 키를 모두 제공해야 합니다.

다음은 출력 `manifest.json` 파일의 예입니다. 실패한 가져오기 작업 문제를 해결하는 첫 단계로이 파일을 사용하는 것이 좋습니다. 각 파일에 대한 세부 정보와 가져오기 작업이 실패한 원인을 제공합니다.

```
  {
  "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
}
```

예:를 사용하여 가져오기 작업 시작 및 모니터링 AWS CLI

다음 예제에서는를 사용하여 가져오기 작업을 AWS Command Line Interface 시작하고 모니터링하는 방법을 보여줍니다. start-fhir-import-job API도 사용할 수 있습니다.

```
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
```

가져오기 작업이 시작되면 다음과 같은 확인 메시지가 표시됩니다.

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

가져오기 작업의 상태를 모니터링하거나 구성 속성을 알아보려면 다음 예제와 같이 describe-fhir-import-job API 또는 AWS CLI 명령을 사용합니다.

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

```
    --region us-east-1
```

응답으로 다음 정보를 받게 됩니다.

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

모든 가져오기 작업 목록을 보려면 다음 예제와 같이 list-fhir-import-jobs API 또는 AWS CLI 명령을 사용합니다. 하나 이상의 필터를 추가하여 결과를 제한할 수 있습니다.

```
    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)
```

응답으로 다음 정보를 받게 됩니다.

```
{
    "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
```

HealthLake 데이터 스토어에서 파일 내보내기

데이터 스토어를 생성하고 데이터를 가져온 후(또는 미리 로드된 샘플 데이터를 사용하는 경우) Amazon S3 버킷으로 데이터를 내보낼 수 있습니다. HealthLake 데이터 스토어에서 데이터를 내보내려면 다음 작업을 사용합니다.

- 및를 사용하여 `StartFHIRExportJob` API 작업을 사용하여 AWS SDKs 내보내기 요청을 수행합니다 HealthLake.
  - 이 작업은 시스템 전체 내보내기 요청만 지원합니다.
- 를 사용하여 `export` 구문을 사용하여 HealthLake FHIR 내보내기 요청을 합니다RESTAPI.
  - 이 작업은 시스템 전체, 환자 및 그룹 내보내기 요청 생성을 지원합니다. 파라미터를 적용하여 내보내기 요청의 데이터를 추가로 필터링할 수도 있습니다.

> **Important**
>
> HealthLake SDK `StartFHIRExportJob` API 작업을 사용한 내보내기 요청과 `StartFHIRExportJobWithPost` API 작업을 사용한 FHIR REST API 내보내기 요청에는 별도의 IAM 작업이 있습니다. 를 사용하여 SDK 내보내`StartFHIRExportJob`고 를 사용하여 FHIR REST API 내보내는 각 IAM 작업은 허용/거부 권한을 별도로 처리할 `StartFHIRExportJobWithPost`수 있습니다. SDK 및 FHIR REST API 내보내기를 모두 제한하려면 각 IAM 작업에 대한 권한을 거부해야 합니다.

두 작업 모두 Amazon S3(S3) 버킷으로 파일 내보내기만 지원합니다. 데이터 스토어의 HealthLake 모든 파일은 줄 바꿈으로 구분된JSON(`.ndjson`) 파일로 내보내지며, 각 줄은 유효한 FHIR 리소스로 구성됩니다.

두 작업 모두 서비스 역할이 필요합니다. 여기서는 서비스 보안 주체로 정의 HealthLake 되어야 하며 파일을 내보낼 Amazon Simple Storage Service(S3) 버킷을 정의해야 합니다. 자세한 내용은 내보내기 작업에 대한 권한 설정을 참조하십시오.

가져오기 또는 내보내기 작업을 대기열에 넣을 수 있습니다. 이러한 비동기 가져오기 또는 내보내기 작업은 FIFO (선입선출) 방식으로 처리됩니다. 가져오기 또는 내보내기 작업이 진행되는 동안 FHIR 리소스를 생성, 읽기, 업데이트 또는 삭제할 수 있습니다.

HealthLake 데이터 스토어에서 파일을 내보내려면 다음 섹션을 참조하세요.

내보내기 작업에 대한 권한 설정

데이터 스토어에서 파일을 내보내기 전에 Amazon S3의 출력 버킷에 액세스할 수 있는 HealthLake 권한을 부여해야 합니다. HealthLake 액세스 권한을 부여하려면에 대한 IAM 서비스 역할을 생성하고 HealthLake, 역할에 신뢰 정책을 추가하여 HealthLake 수임 역할 권한을 부여하고, Amazon S3 버킷에 대한 액세스 권한을 부여하는 권한 정책을 역할에 연결합니다.

HealthLake 에서에 대한 역할을 이미 생성한 경우 가져오기 작업에 대한 권한 설정역할을 재사용하고 이 주제에 나열된 내보내기 Amazon S3 버킷에 대한 추가 권한을 부여할 수 있습니다. IAM 역할 및 신뢰 정책에 대한 자세한 내용은 IAM 정책 및 권한을 참조하세요.

> **Important**
>
> HealthLake SDK `StartFHIRExportJob` API 작업을 사용한 내보내기 요청과 `StartFHIRExportJobWithPost` API 작업을 사용한 FHIR REST API 내보내기 요청에는 별도의 IAM 작업이 있습니다. 를 사용하여 SDK 내보내`StartFHIRExportJob`고 를 사용하여 FHIR REST API 내보내는 각 IAM 작업은 허용/거부 권한을 별도로 처리할 `StartFHIRExportJobWithPost`수 있습니다. SDK 및 FHIR REST API 내보내기를 모두 제한하려면 각 IAM 작업에 대한 권한을 거부해야 합니다. 사용자에게에 대한 전체 액세스 권한을 부여하면 IAM 사용자 권한을 변경할 필요가 HealthLake없습니다.

권한을 설정하는 사용자 또는 역할에는 역할을 생성하고, 정책을 생성하고, 정책을 역할에 연결할 수 있는 권한이 있어야 합니다. 다음 IAM 정책은 이러한 권한을 부여합니다.

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

내보내기 권한을 설정하려면

1. 아직 생성하지 않은 경우 데이터 스토어에서 내보낼 데이터에 대한 대상 Amazon S3 버킷을 생성합니다. Amazon S3 버킷은 서비스와 동일한 AWS 리전에 있어야 하며 모든 옵션에 대해 퍼블릭 액세스 차단을 활성화해야 합니다. 자세한 내용은 Amazon S3 퍼블릭 액세스 차단 사용을 참조하세요. Amazon 소유 또는 고객 소유 KMS 키도 암호화에 사용해야 합니다. KMS 키 사용에 대한 자세한 내용은 Amazon Key Management Service를 참조하세요.
2. 아직 생성하지 않은 경우에 대한 데이터 액세스 서비스 역할을 생성하고 HealthLake 서비스에 다음 신뢰 정책으로 수임할 수 있는 권한을 HealthLake 부여합니다.는 이를 HealthLake 사용하여 출력 Amazon S3 버킷을 씁니다. 에서 이미 생성한 경우 가져오기 작업에 대한 권한 설정재사용하여 다음 단계에서 Amazon S3 버킷에 대한 권한을 부여할 수 있습니다.

```
{
    "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. 데이터 액세스 역할에 출력 Amazon S3 버킷에 액세스할 수 있는 권한 정책을 추가합니다. amzn-s3-demo-bucket를 버킷 이름으로 바꿉니다.

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

HealthLake 콘솔 또는를 사용하여 데이터 스토어에서 파일 내보내기 AWS SDKs

를 완료한 후 데이터 스토어에서 Amazon Simple Storage Service(Amazon S3) 버킷으로 파일을 내보낼 내보내기 작업에 대한 권한 설정수 있습니다. 데이터 스토어에서 파일을 내보내려면에서 내보내기 작업을 시작합니다 HealthLake. 내보내기 작업은 데이터 스토어에서 파일을 줄 바꿈으로 구분된 JSON(`.ndjson`) 형식으로 내보냅니다. 여기서 각 줄은 유효한 FHIR 리소스로 구성됩니다. 내보내기 작업을 시작할 때 암호화를 위한 AWS KMS 키를 지정해야 합니다. KMS 키 생성에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 키 생성을 참조하세요.

다음 주제에서는 AWS HealthLake 콘솔을 사용하여 내보내기 작업을 시작하고 start-fhir-export-job API 작업을 사용하여 AWS SDKs를 시작하는 방법을 다룹니다.

주제

- 데이터 스토어에서 파일 내보내기(콘솔)
- 데이터 스토어에서 파일 내보내기(AWS SDKs)

데이터 스토어에서 파일 내보내기(콘솔)

파일을 내보내려면(콘솔) 다음 단계를 사용합니다.

1. 와 동일한 리전에 출력 S3 버킷을 생성합니다 HealthLake.
2. 새 내보내기 작업을 시작하려면 출력 Amazon S3 버킷을 식별하고 사용하려는 IAM 역할을 생성하거나 식별합니다. IAM 역할 및 신뢰 정책에 대한 자세한 내용은 IAM 역할을 참조하세요. KMS 키 암호화도 사용합니다. KMS 키 사용에 대한 자세한 내용은 Amazon Key Management Service를 참조하세요.
3. 내보내기 작업의 상태를 보려면 `ListFHIRExportJobs` API 작업을 사용합니다.

데이터 스토어에서 파일 내보내기(AWS SDKs)

를 사용하여 데이터 스토어에서 파일을 내보내려면 start-fhir-export-job 작업을 AWS SDKs사용합니다. 다음 코드는 SDK for Python(Boto3)을 사용하여 내보내기 작업을 시작하는 방법을 보여줍니다.

```
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'])
```

FHIR 내보내기 작업의 ID, ARN, 이름, 시작 시간, 종료 시간 및 현재 상태를 가져오려면를 사용합니다describe-fhir-export-job. list-fhir-export-jobs를 사용하여 모든 내보내기 작업과 해당 상태를 나열합니다.

다음 코드는 SDK for Python(Boto3)을 사용하여 특정 내보내기 작업의 속성을 가져오는 방법을 보여줍니다.

```
import boto3

client = boto3.client('healthlake')

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

FHIR REST API 작업을 사용하여 데이터 스토어에서 HealthLake 데이터 내보내기

를 완료한 후 FHIR REST API 작업을 사용하여 데이터 스토어에서 HealthLake 데이터를 내보낼 내보내기 작업에 대한 권한 설정수 있습니다. 를 사용하여 내보내기 요청을 수행하려면 필요한 권한이 있는 IAM 사용자, 그룹 또는 역할이 FHIR REST API있어야 하며 요청의 `$export` 일부로 `POST`를 지정하고 요청 본문에 요청 파라미터를 포함해야 합니다. FHIR 사양에 따라 FHIR 서버는 `GET` 요청을 지원해야 하며 `POST` 요청을 지원할 수 있습니다. 추가 파라미터를 지원하려면 내보내기를 시작하는 데 본문이 필요하므로에서 POST 요청을 HealthLake 지원합니다.

> **Important**
>
> HealthLake 2023년 6월 1일 이전에 생성된 데이터 스토어는 시스템 전체 내보내기에 대한 FHIR REST API 기반 내보내기 작업 요청만 지원합니다.
> HealthLake 2023년 6월 1일 이전에 생성된 데이터 스토어는 데이터 스토어의 엔드포인트에서 GET 요청을 사용하여 내보내기 상태를 가져오는 것을 지원하지 않습니다.

를 사용하여 수행하는 모든 내보내기 요청은 ndjson 형식으로 반환되고 Amazon S3 버킷으로 내보내FHIRRESTAPI집니다. 각 S3 객체에는 단일 FHIR 리소스 유형만 포함됩니다.

AWS 계정 할당량에 따라 내보내기 요청을 대기열에 넣을 수 있습니다. 와 연결된 Service Quotas에 대한 자세한 내용은 섹션을 HealthLake참조하세요AWS HealthLake 엔드포인트 및 할당량.

HealthLake 는 다음과 같은 세 가지 유형의 대량 내보내기 엔드포인트 요청을 지원합니다.

유형	설명	구문
시스템 내보내기	서버에서 모든 데이터를 내보냅니다 HealthLake FHIR.	`POST https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/$export`
모든 환자	환자 리소스 유형과 연결된 리소스 유형을 포함하여 모든 환자와 관련된 모든 데이터를 내보냅니다.	`POST https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/Patient/$export`
환자 그룹	그룹 ID로 지정된 환자 그룹과 관련된 모든 데이터를 내보냅니다.	`POST https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/Group/ID/$export`

시작하기 전 준비 사항

용를 사용하여 내보내기 요청을 하려면 다음 요구 사항을 충족FHIRRESTAPI합니다 HealthLake.

- 내보내기 요청을 수행하는 데 필요한 권한이 있는 사용자, 그룹 또는 역할을 설정해야 합니다. 자세한 내용은 export 요청 승인을 참조하십시오.

- 데이터를 내보낼 Amazon S3 버킷에 대한 HealthLake 액세스 권한을 부여하는 서비스 역할을 생성했어야 합니다. 서비스 역할도를 서비스 보안 주체 HealthLake 로 지정해야 합니다. 권한 설정에 대한 자세한 내용은 섹션을 참조하세요내보내기 작업에 대한 권한 설정.

export 요청 승인

를 사용하여 성공적인 내보내기 요청을 수행하려면 IAM 또는 FHIR REST OAuth2.0을 사용하여 사용자, 그룹 또는 역할을 API승인합니다. 서비스 역할도 있어야 합니다.

를 사용하여 요청 승인 IAM

`$export` 요청을 할 때 사용자, 그룹 또는 역할에는 정책에 포함된 `StartFHIRExportJobWithPost`, `DescribeFHIRExportJobWithGet`, 및 `CancelFHIRExportJobWithDelete` IAM 작업이 있어야 합니다.

> **Important**
>
> HealthLake SDK `StartFHIRExportJob` API 작업을 사용한 내보내기 요청과 `StartFHIRExportJobWithPost` API 작업을 사용한 FHIR REST API 내보내기 요청에는 별도의 IAM 작업이 있습니다. 로 SDK 내보내기 `StartFHIRExportJob` 및 로 FHIR REST API 내보내기와 같은 각 IAM 작업은 별도로 허용/거부 권한을 처리할 `StartFHIRExportJobWithPost`수 있습니다. SDK 및 FHIR REST API 내보내기를 모두 제한하려면 각 IAM 작업에 대한 권한을 거부해야 합니다.

SMART에서를 사용하여 요청 승인FHIR(OAuth 2.0)

FHIR 활성화된 HealthLake 데이터 스토어SMART에서를 `$export` 요청할 때는 적절한 범위가 할당되어야 합니다. 지원되는 범위에 대한 자세한 내용은 섹션을 참조하세요HealthLake 데이터 스토어 FHIR 리소스별 범위.

export 요청하기

이 섹션에서는 FHIR REST를 사용하여 내보내기 요청을 할 때 수행해야 하는 필수 단계를 설명합니다 API.

AWS 계정에 실수로 요금이 부과되지 않도록 `export` 구문을 제공하지 않고 POST 요청하여 요청을 테스트하는 것이 좋습니다.

요청을 하려면 다음을 수행해야 합니다.

1. 지원되는 엔드포인트export에 URL 대한 POST 요청에를 지정합니다.
2. 필요한 헤더 파라미터를 지정합니다.
3. 필요한 파라미터를 정의하는 요청 본문을 지정합니다.

1단계: 지원되는 엔드포인트**export**에 URL 대한 **POST** 요청에를 지정합니다.

HealthLake 는 세 가지 유형의 대량 내보내기 엔드포인트 요청을 지원합니다. 대량 내보내기 요청을 하려면 지원되는 세 엔드포인트 중 하나에 대해 POST기반 요청을 해야 합니다. 다음 예제에서는 요청 export에서를 지정하는 방법을 보여줍니다URL.

- `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`

해당 POST 요청 문자열에서 지원되는 다음과 같은 검색 파라미터를 사용할 수 있습니다.

지원되는 검색 파라미터

HealthLake 는 대량 내보내기 요청에서 다음 검색 수정자를 지원합니다.

이러한 예제에는 요청을 제출하기 전에 인코딩해야 하는 특수 문자가 포함되어 있습니다.

명칭	필수?	설명	예제
`_outputFormat`	아니요	생성할 요청된 대량 데이터 파일의 형식입니다. 허용되는 값은 `application/fhir+ndjson` , `application/ndjson` , 입니다ndjson.	

명칭	필수?	설명	예제
_type	아니요	내보내기 작업에 포함할 쉼표로 구분된 FHIR 리소스 유형의 문자열입니다. 모든 리소스를 내보낼 때 비용 영향이 있을 수 _type 있으므로를 포함하는 것이 좋습니다.	&_type=MedicationStatement,Observation
_since	아니요	날짜 타임스탬프 당일 또는 이후에 수정된 리소스 유형입니다. 리소스 유형에 마지막으로 업데이트된 시간이 없는 경우 응답에 포함됩니다.	&_since=2024-05-09T00%3A00%3A00Z

2단계: 필요한 헤더 파라미터 지정

를 사용하여 내보내기 요청을 하려면 다음 두 헤더 파라미터를 지정FHIRRESTAPI해야 합니다.

- 콘텐츠 타입: application/fhir+json
- 선호 사항: respond-async

그런 다음 요청 본문에서 필수 요소를 지정해야 합니다.

3단계:가 필요한 파라미터를 정의하는 요청 본문을 지정합니다.

내보내기 요청에는 JSON 형식의 본문도 필요합니다. 본문에는 다음 파라미터가 포함될 수 있습니다.

키	필수?	설명	값
DataAccessRoleArn	예	HealthLake 서비스 역할ARN의 입니다. 사용되는 서비스 역할은	arn:aws:iam:: **444455556**

키	필수?	설명	값
		를 서비스 보안 주체 HealthLake 로 지정해야 합니다.	**666** :role/**your-healthlake-service-role**
JobName	아니요	내보내기 요청의 이름입니다.	**your-export-job-name**
S3Uri	예	OutputDataConfig 키의 일부입니다. 내보낸 데이터를 다운로드할 대상 버킷URI의 S3입니다.	s3://DOC-EXAMPLE-DESTINATION-BUCKET/ **EXPORT-JOB** /
KmsKeyId	예	OutputDataConfig 키의 일부입니다. Amazon S3 버킷을 보호하는 데 사용되는 AWS KMS 키ARN의 입니다.	arn:aws:kms: **region-of-bucket:123456789012** :key/**1234abcd-12ab-34cd-56ef-1234567890ab**

Example -를 사용하여 수행된 내보내기 요청의 본문 FHIR REST API

를 사용하여 내보내기 요청을 하려면 다음과 같이 본문을 지정FHIRRESTAPI해야 합니다.

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

요청이 성공하면 다음과 같은 응답을 받게 됩니다.

응답 헤더

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

응답 본문

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

내보내기 요청 관리

내보내기 요청이 성공하면 export를 사용하여 현재 내보내기 요청의 상태를 설명하고 현재 내보내기 요청을 취소export하여 해당 요청을 관리할 수 있습니다.

를 사용하여 내보내기 요청을 취소RESTAPI하는 경우 취소 요청을 제출한 시점까지 내보낸 데이터 부분에 대해서만 요금이 청구됩니다.

다음 주제에서는 현재 내보내기 요청의 상태를 가져오거나 취소할 수 있는 방법을 설명합니다.

내보내기 요청 취소

내보내기 요청을 취소하려면 DELETE 요청을 하고 요청에서 작업 ID를 제공합니다URL.

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

요청이 성공하면 다음을 받게 됩니다.

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

```
}
```

요청이 성공하지 못하면 다음을 수신합니다.

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

내보내기 요청 설명

내보내기 요청의 상태를 가져오려면 `export` 및를 사용하여 `GET` 요청합니다**`export-request-job-id`**.

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

JSON 응답에는 `ExportJobProperties` 객체가 포함됩니다. 여기에는 다음과 같은 키:값 페어가 포함될 수 있습니다.

명칭	필수?	설명	값
DataAccessRoleArn	아니요	HealthLake 서비스 역할ARN의 입니다. 사용되는 서비스 역할은를 서비스 보안 주체 HealthLake 로 지정해야 합니다.	`arn:aws:iam:: 444455556666 :role/your-healthlake-service-role`
SubmitTime	아니요	내보내기 작업이 제출된 날짜입니다.	`Apr 21, 2023 5:58:02`
EndTime	아니요	내보내기 작업이 완료된 시간입니다.	`Apr 21, 2023 6:00:08 PM`

명칭	필수?	설명	값
JobName	아니요	내보내기 요청의 이름입니다.	**your-export-job-name**
JobStatus	아니요		유효한 값은 다음과 같습니다. `SUBMITTED | IN_PROGRESS | COMPLETED_WITH_ERRORS | COMPLETED | FAILED`
S3Uri	예	OutputDataConfig 객체의 일부입니다. 내보낸 데이터를 다운로드할 대상 버킷URI의 Amazon S3입니다.	s3://DOC-EXAMPLE-DESTINATION-BUCKET/ **EXPORT-JOB** /
KmsKeyId	예	OutputDataConfig 객체의 일부입니다. Amazon S3 버킷을 보호하는 데 사용되는 AWS KMS 키ARN의 입니다.	arn:aws:kms: **region-of-bucket:123456789012** :key/**1234abcd-12ab-34cd-56ef-1234567890ab**

Example :를 사용하여 수행된 설명 내보내기 요청의 본문 FHIR REST API

성공하면 다음 JSON 응답을 받게 됩니다.

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

에서 데이터 스토어 삭제 HealthLake

데이터 스토어 삭제는 비동기 작업입니다. 시작되면 상태가 삭제 중으로 변경됩니다. 데이터 스토어는 날짜 스토어의 모든 FHIR 데이터와 필요한 기본 인프라도 제거될 때까지 삭제 상태를 유지합니다.

데이터 및 인프라가 제거되면 HealthLake 데이터 스토어 상태가 삭제됨으로 변경됩니다. 삭제 후 데이터 스토어에 대한 세부 정보는 `DescribeFHIRDataStore` 및 `ListFHIRDataStores` 작업을 7일 동안 사용해야만 사용할 수 있습니다. 7일이 지나면 삭제된 데이터 스토어가 결과에 표시되지 않습니다.

데이터 스토어를 성공적으로 삭제하려면 요청을 하는 사용자, 그룹 또는 역할에 해당 IAM 정책에 IAM 작업이 `glue:DeleteDatabase` 추가되어 있어야 합니다. 이 IAM 작업은 관리형 정책인의 AWS 일부로 포함되지 않습니다`AmazonHealthLakeFullAccess`.

AWS Management Console AWS SDKs또는를 사용하여 데이터 스토어를 삭제할 수 있습니다 AWS CLI.

주제

데이터 스토어 삭제(콘솔)

콘솔을 사용하여 데이터 스토어를 삭제하려면 데이터 스토어 페이지에서 데이터 스토어를 선택하고 삭제를 선택합니다.

HealthLake 데이터 스토어를 삭제하려면

1. https://console.aws.amazon.com//healthlake/집에서 HealthLake 콘솔을 엽니다.
2. 탐색 창(☰)을 엽니다.
3. 그런 다음 데이터 스토어를 선택합니다.
4. 데이터 스토어 페이지에서 삭제하려는 데이터 스토어 옆에 있는 옵션을 선택합니다.
5. 그런 다음 삭제를 선택합니다.
6. 대화 상자에서 **delete**를 입력하여 선택한 데이터 스토어를 삭제할지 확인합니다.
7. 그런 다음 삭제를 선택합니다. 그러면 데이터 스토어의 상태가 활성에서 삭제 중으로 변경됩니다.

데이터 스토어 삭제(AWS SDKs 및 AWS CLI)

아래 코드 샘플을 사용하여 HealthLake 데이터 스토어를 삭제할 수 있습니다.

AWS CLI

다음 예제에서는에서 `DeleteFHIRDatastore` 작업을 사용하는 방법을 보여줍니다 AWS CLI. 이 예제를 실행하려면 AWS CLI를 설치해야 합니다.

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

성공하면 다음과 같은 JSON 응답을 받게 됩니다.

```
{
    "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)

Python용는 AWS SDK 단일 파라미터에서를 사용하는 `describe_fhir_datastore` 메서드를 지원합니다`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)
```

성공하면 python 사전을 반환합니다.

```
{'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}}
```

한 번에 두 개 이상의 데이터 스토어에 대한 세부 정보를 반환하려면 `ListFHIRDatastore`

다음 예제와 AWS CLI 같이를 사용하여 `DeleteFHIRDataStore` 명령을 사용합니다. delete-fhir-datastore API 또는 콘솔을 사용하여 데이터 스토어를 삭제할 수도 있습니다. 데이터 스토어를 삭제하면 데이터 스토어와 기본 인프라에 포함된 모든 FHIR 리소스 버전이 제거됩니다. 삭제된 데이터 스토어와 관련된 로그는 HIPAA 지침에 따라 서비스 계정 내에 보존됩니다.

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

다음 예제 JSON 응답에서 볼 수 있듯이 상태가 “DELETING”로 변경되어 데이터 스토어와 해당 콘텐츠가 삭제 중인지 확인합니다.

```
{
```

```
  "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)"
}
```

HealthLake 데이터 스토어와의 FHIR REST API 상호 작용 사용

에서는 Fast Healthcare 상호 운용성 리소스(FHIR) REST API 상호 작용을 AWS HealthLake사용하여 데이터 스토어의 FHIR 리소스를 관리하고 검색합니다. FHIR REST API 상호 작용은 데이터 스토어의 리소스에서 생성, 읽기, 업데이트 및 삭제(CRUD) 상호 작용을 수행하는 데 사용됩니다. 는 FHIR지원되는 검색 작업의 하위 집합을 지원하므로 GET HealthLake 또는 POST HTTP 요청을 사용하여 복잡한 검색 문자열을 구성할 수도 있습니다.

적합성을 위해 FHIR 리소스 유형은 HL7 FHIR R4 StructureDefinition 리소스에 따라 검증됩니다. 활성 HealthLake 데이터 스토어의 FHIR관련 기능을 찾으려면 다음과 URL같이에 metadata가 지정된를 GET 요청합니다.

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

성공하면 HealthLake 데이터 스토어에 대한 200 HTTP 응답 코드와 기능 설명을 받게 됩니다. 자세한 내용은 HL7 FHIR R4 설명서CapabilityStatement의 섹션을 참조하세요.

다음 표에는에서 지원하는 FHIR 상호 작용이 나열되어 있습니다 AWS HealthLake.

FHIR에서 지원하는 상호 작용 AWS HealthLake

FHIR 상호 작용	설명
전체 시스템 상호 작용	
capabilities	시스템에 대한 기능 설명 가져오기
batch/transaction	단일 상호 작용에서 리소스 집합 업데이트, 생성 또는 삭제
유형 수준 상호 작용	
create	서버 할당 ID로 새 리소스 생성
search	일부 필터 기준을 기반으로 리소스 유형 검색
history	특정 리소스 유형에 대한 변경 기록 검색

FHIR 상호 작용	설명
인스턴스 수준 상호 작용	
read	리소스의 현재 상태 읽기
history	특정 리소스에 대한 변경 기록 읽기
vread	리소스의 특정 버전 상태 읽기
update	ID로 리소스 업데이트(또는 새 리소스인 경우 생성)
delete	리소스 삭제

주제

- 에서 지원되는 FHIR 리소스 유형 AWS HealthLake
- HealthLake 데이터 스토어에서 생성, 읽기, 업데이트 및 삭제(CRUD) 작업 수행
- FHIR REST API 작업을 사용하여 HealthLake 데이터 스토어 검색
- FHIR 리소스 기록 읽기
- 환자 $everything FHIR REST API 작업으로 환자 데이터 가져오기
- $export를 사용하여 데이터 스토어에서 HealthLake 데이터 내보내기

에서 지원되는 FHIR 리소스 유형 AWS HealthLake

다음 표에는에서 지원하는 FHIR R4 리소스 유형이 나열되어 있습니다 AWS HealthLake. 자세한 내용은 HL7 FHIR R4 설명서의 리소스 인덱스를 참조하세요.

FHIR에서 지원하는 R4 리소스 유형 HealthLake

계정	DetectedIssue	인보이스	실무자
ActivityDefinition	장치	라이브러리	PractitionerRole
AdverseEvent	DeviceDefinition	연결	절차
AllergyIntolerance	DeviceMetric	나열	증명

예약	DeviceUseStatement	위치	설문 조사
AppointmentResponse	DeviceRequest	치수	QuestionnaireResponse
AuditEvent-노트 참조	DiagnosticReport	MeasureReport	RelatedPerson
바이너리	DocumentManifest	미디어	RequestGroup
BodyStructure	DocumentReference	약물	ResearchStudy
번들 - 노트 참조	EffectEvidenceSynthesis	MedicationAdministration	ResearchSubject
CapabilityStatement	상황	MedicationDispense	RiskAssessment
CarePlan	엔드포인트	MedicationKnowledge	RiskEvidenceSynthesis
CareTeam	EpisodeOfCare	MedicationRequest	일정
ChargeItem	EnrollmentRequest	MedicationStatement	ServiceRequest
ChargeItemDefinition	EnrollmentResponse	MessageHeader	슬롯
Claim	ExplanationOfBenefit	MolecularSequence	표본
ClaimResponse	FamilyMemberHistory	NutritionOrder	StructureDefinition
통신	플래그	관측치	StructureMap
CommunicationRequest	목표	OperationOutcome	Substance
구성	그룹	조직	SupplyDelivery
ConceptMap	GuidanceResponse	OrganizationAffiliation	SupplyRequest
Condition	HealthcareService	파라미터	작업
동의	ImagingStudy	환자	ValueSet

계약	면역화	PaymentNotice	VisionPrescription
적용 범위	ImmunizationEvaluation	PaymentReconciliation	VerificationResult
CoverageEligibilityRequest	ImmunizationRecommendation	Person	
CoverageEligibilityResponse	InsurancePlan	PlanDefinition	

⚠ FHIR 사양 및 HealthLake

- 바이너리, 번들 OperationOutcome및 파라미터와 같은 리소스 유형을 사용하여 GET 또는 POST를 요청할 수 없습니다.
- AuditEvent - AuditEvent 리소스를 생성하거나 읽을 수 있지만 업데이트하거나 삭제할 수는 없습니다.
- 번들 - 번들 요청을 HealthLake 관리하는 방법에는 여러 가지가 있습니다. 자세한 내용은 번들을 사용하여 여러 FHIR 리소스 관리을 참조하세요.
- VerificationResult —이 리소스 유형은 2023년 12월 9일 이후에 생성된 데이터 스토어에서만 지원됩니다.

HealthLake 데이터 스토어에서 생성, 읽기, 업데이트 및 삭제(CRUD) 작업 수행

데이터 스토어를 관리하고, 데이터를 가져오고, 데이터를 내보낼 때 네이티브 AWS 작업을 사용하지만, 4가지 주요 FHIR HTTP 작업을 사용하여 HealthLake 데이터 스토어 내에서 FHIR 리소스를 생성(POST), 읽기(GET), 업데이트(PUT) 및 삭제(DELETE)합니다. 다음 주제에서는 FHIR REST API 서비스를 사용하여 HealthLake 데이터 스토어에서 생성, 읽기, 업데이트 및 삭제(CRUD) 작업을 수행하는 방법을 설명합니다. HTTP 클라이언트를 통해 전송된 요청을 인증 HealthLake API하려면 서명 버전 4 서명 프로세스를 사용해야 합니다. 자세한 내용은의 서명 버전 4 서명 프로세스를 참조하세요AWS 일반 참조.

주제

- 를 사용하여 리소스 생성 POST
- 를 사용하여 리소스 읽기 GET
- 를 사용하여 리소스 업데이트 PUT
- 를 사용하여 리소스 삭제 DELETE
- 번들을 사용하여 여러 FHIR 리소스 관리

를 사용하여 리소스 생성 **POST**

POST 요청을 사용하여 HealthLake 데이터 스토어에서 새 리소스를 생성합니다. POST 요청에는 id 요소를 제공할 필요가 없습니다. 리소스가 성공적으로 생성되면 HealthLake 서버는 201 생성된 HTTP 상태 코드를 반환합니다.

> **Note**
>
> DocumentReference 리소스 유형에 대한 POST 요청을 하면 기존 확장은 수정되지 않습니다. 대신 기존 확장이 포함된 새 확장을 데이터 스토어에 AWS HealthLake 추가합니다. HealthLake 가 DocumentReference 리소스 유형에서 자연어 처리(NLP)를 사용하여 중요한 의료 데이터를 추출하는 방법에 대한 자세한 내용은 섹션을 참조하세요에서 리소스 유형의 자연어 처리(NLP)를 기반으로 자동화된 FHIR DocumentReference 리소스 생성 사용 AWS HealthLake.

Example **POST** 요청을 사용하여 **Patient** 리소스 생성.

HealthLake 데이터 스토어 POST 요청을 생성하려면 데이터 스토어의 엔드포인트를 사용하고 JSON 요청 본문을 제공합니다. 데이터 스토어의 엔드포인트를 찾으려면 HealthLake 콘솔의 데이터 스토어를 살펴보거나 AWS HealthLake API 참조의 DescribeFHIRDatastore 작업을 사용합니다.

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 응답

환자 리소스 생성을 확인하기 위해 201 생성된 HTTP 상태 코드와 다음 JSON 응답을 받게 됩니다.

```
{
  "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"
  }
}
```

를 사용하여 리소스 읽기 **GET**

이 예제에서는 GET 요청을 사용하여 환자 FHIR 리소스를 읽는 방법을 보여줍니다.

Example **GET** 요청을 사용하여 특정 **Patient** 리소스 읽기.

HealthLake 데이터 스토어 GET 요청을 생성하려면 데이터 스토어의 엔드포인트를 사용합니다. 데이터 스토어의 엔드포인트를 찾으려면 HealthLake 콘솔의 데이터 스토어를 살펴보거나 AWS HealthLake API 참조의 DescribeFHIRDatastore 작업을 사용합니다.

또한 리소스 유형**Patient**과 유효한 식별자를 포함해야 합니다**2de04858-ba65-44c1-8af1-f2fe69a977d9**.

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

JSON 응답

성공하면 200 HTTP 상태 코드와 다음 JSON 응답을 받게 됩니다.

```
{
    "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"
}
```

를 사용하여 리소스 업데이트 **PUT**

다음 예제에서는를 PUT 사용하여 환자 FHIR 리소스 유형의 환자에 대한 세부 정보를 업데이트하는 방법을 보여줍니다. 또한 아직 생성되지 않은 리소스에 대해 PUT 요청하면 초기 버전이 생성됩니다.

리소스가 업데이트된 경우에서 `200` HTTP 상태 코드를 반환하거나 새 리소스가 생성된 경우 `201` HTTP 상태 코드를 반환하도록 요청합니다.

> **Note**
>
> `DocumentReference` 리소스 유형에 대해 PUT 요청하면 기존 확장이 수정되지 않습니다. 대신 기존 확장이 포함된 새 확장을 데이터 스토어에 AWS HealthLake 추가합니다. HealthLake 가 `DocumentReference` 리소스 유형에서 자연어 처리(NLP)를 사용하여 중요한 의료 데이터를 추출하는 방법에 대한 자세한 내용은 섹션을 참조하세요에서 리소스 유형의 자연어 처리(NLP)를 기반으로 자동화된 FHIR DocumentReference 리소스 생성 사용 AWS HealthLake.

Example **PUT** 요청을 사용하여 **`Patient`** 리소스 유형 업데이트

PUT 요청을 할 때는 데이터 스토어의 엔드포인트, 업데이트하려는 리소스 유형의 이름, 식별자 및 JSON 요청 본문이 필요합니다.

PUT를 사용하여 새 리소스를 생성하는 경우 제공된 식별자를 사용하여 새 리소스를 생성합니다.

PUT Request

유효한 PUT 요청의 예제 구조:

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

JSON Request Body

지정된 환자 리소스를 업데이트하는 데 사용되는 예제 JSON 본문입니다.

```
{
  "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 응답

변경 사항을 확인하기 위해 다음 JSON 메시지가 표시됩니다.

```
{
 "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"
```

```
  }
}
```

조건부 업데이트

조건부 업데이트를 사용하면 논리적 ID가 아닌 일부 식별 검색 기준에 따라 기존 리소스를 업데이트할 수 있습니다. 서버가이 업데이트를 처리하면이 요청에 대한 단일 논리적 ID를 해결하기 위해 리소스 유형에 대한 표준 검색 기능을 사용하여 검색을 수행합니다.

이 작업이 수행되는 작업은 발견된 일치 항목 수에 따라 달라집니다.

- 일치 없음, 요청 본문에 제공된 ID 없음: 서버가 리소스를 생성합니다.
- 일치 항목 없음, 제공된 ID 및 리소스가 ID와 함께 아직 존재하지 않음: 서버는 상호 작용을 업데이트로 상호 작용 생성을 처리합니다.
- 일치 항목 없음, ID 제공 및 이미 존재: 서버가 `409 Conflict` 오류와 함께 업데이트를 거부합니다.
- 일치 항목 1개, 리소스 ID가 제공되지 않음 OR(리소스 ID가 제공되고 검색된 리소스와 일치함): 서버는 위와 같이 일치하는 리소스에 대해 업데이트를 수행합니다. 여기서 리소스가 업데이트된 경우 서버는를 SHALL 반환합니다`200 OK`.
- 하나의 일치, 리소스 ID가 제공되었지만 발견된 리소스와 일치하지 않음: 서버가 클라이언트 ID 사양이 가급적이면에 문제가 있음을 나타내는 `409 Conflict` 오류를 반환합니다. OperationOutcome
- 여러 일치 항목: 서버가 `412 Precondition Failed` 클라이언트의 기준이 가급적이면 OperationOutcome

Example - 이름이 peter이고 생년월일이 2000년 1월 1일이며 전화번호가 1234567890인 환자 리소스를 업데이트합니다.

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

를 사용하여 리소스 삭제 **DELETE**

HealthLake 데이터 스토어에서 리소스를 삭제하려면 `DELETE` HTTP 요청을 해야 합니다.

Example **`DELETE`** 요청을 사용하여 특정 **`Patient`** 리소스 유형 삭제

`DELETE` 요청을 생성하려면 데이터 스토어의 엔드포인트를 사용합니다. 데이터 스토어의 엔드포인트를 찾으려면 HealthLake 콘솔의 데이터 스토어를 살펴보거나 AWS HealthLake API 참조에 있는 DescribeFHIRDatastore 작업을 사용합니다.

리소스 유형과 유효한 식별자도 포함해야 합니다.

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

HTTP 응답

성공하면 리소스가 더 이상 데이터 스토어에 없음을 확인하는 204 HTTP 상태 코드를 받게 됩니다. 삭제 요청이 실패하면 DELETE 요청이 실패한 이유를 나타내는 400 시리즈 HTTP 상태 코드를 받게 됩니다.

번들을 사용하여 여러 FHIR 리소스 관리

HL7 FHIR R4 사양에서 번들은 단순히 리소스 모음입니다.는 FHIR REST API 요청에 번들 리소스 유형을 생성하고 번들 트랜잭션을 사용하여 단일 FHIR REST API 요청에서 여러 CRUD 작업을 수행할 수 있도록 HealthLake 지원합니다. 번들 트랜잭션에서는 batch FHIR REST API 요청에서와 같이 번들 유형을 지정해야 합니다.

모든 번들 요청은에 의해 기록됩니다 AWS CloudTrail. CloudTrail 와 함께를 사용하는 방법에 대한 자세한 내용은 섹션을 HealthLake참조하세요AWS CloudTrail을 사용하여 AWS HealthLake API 호출 로깅.

HL7 FHIR R4 리소스(외부)

- 전체 사양을 읽으려면 FHIR 설명서 인덱스의 리소스 유형: 번들을 참조하세요.
- 를 사용한 배치 상호 작용에 대한 자세한 내용은 FHIR 설명서 인덱스의 FHIR REST를 사용한 배치 상호 작용을 API참조하세요. FHIR REST API

아래 섹션에서는 새 번들 리소스를 생성하거나 번들 트랜잭션을 사용하여 리소스를 개별적으로 처리하기 위해 FHIR REST API 요청을 구성하는 방법을 설명합니다.

> ⚠ HealthLake 콘솔, AWS CLI및 간의 차이점 AWS SDKs
>
> HealthLake 콘솔은 요청에 번들 리소스 유형이 지정된 번들 유형 작업만 지원합니다 FHIRRESTAPIURL.

FHIR 번들을 사용하여 여러 CRUD 작업 수행

요청에 리소스 유형이 지정되지 않은 경우 URL FHIR REST API 요청은 개별 데이터 스토어 트랜잭션으로 구문 분석됩니다. JSON 본문에 제공된 각 CRUD 작업이 평가되고 특정 HTTP 상태 코드가 반환됩니다.는 번들 유형를 HealthLake 지원합니다`batch`.

단일 FHIR REST API 요청에서 여러 CRUD 작업을 수행하려면 다음을 수행합니다.

다음 목록은 번들 요청에 사용되는 FHIR REST API 요청 본문의 잘린 부분을 보여줍니다. 전체 요청 본문은 여러 CRUD 작업이 포함된 번들 요청 생성을 참조하세요.

1. POST 요청에 리소스 유형을 지정하지 마십시오.

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

2. 요청 본문에서 번들 유형을 로 지정합니다. `"type": "batch"`
3. 요청 본문에서 `resource` 키로 시작하는 각 CRUD 상호 작용에 대한 리소스별 데이터를 지정합니다.
4. 각 CRUD 작업은 다음과 같이 요청 본문`request`에서 로 지정됩니다.

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

JSON 응답에서 요청에 지정된 각 CRUD 작업에 대한 HTTP 상태 코드를 가져옵니다.

HealthLake 번들 트랜잭션 제한

- 번들의 제한 HealthLake 위치에 대한 자세한 내용은 섹션을 참조하세요AWS HealthLake 엔드포인트 및 할당량.

다음은 여러 작업을 포함하는 번들 CRUD 작업의 예입니다.

Example - 여러 CRUD 작업이 포함된 번들 요청 생성.

여러 CRUD 작업을 수행하는 FHIR REST API 요청을 하려면 데이터 스토어 엔드포인트를 사용하여 POST 요청을 하고 JSON 요청 본문을 제공해야 합니다.

HealthLake 콘솔의 데이터 스토어에서 또는 AWS HealthLake API 참조의 DescribeFHIRDatastore 작업을 사용하여 데이터 스토어의 엔드포인트를 찾을 수 있습니다.

POST Request

데이터 스토어의 엔드포인트를 사용하여 POST 요청합니다. 다음 탭인 JSON 요청 본문을 사용하여 요청 본문의 필수 요소를 확인합니다.

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

JSON Request Body

요청 본문에서 개별 CRUD 요청에 대한 다른 FHIR 리소스별 데이터와 함께 다음 키:값 페어를 제공해야 합니다. 첫 번째 예제는 필요한 요소를 강조 표시하는 잘린 JSON 요청 본문을 보여줍니다. 두 번째 예제는 전체 JSON 요청 본문을 보여줍니다.

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

다음은 새 Patient 및 Medication 리소스 유형 생성을 보여주는 전체 예제입니다.

```
{
  "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 응답

예제 번들 트랜잭션에 지정된 리소스의 생성을 확인하려면 포함된 각 CRUD 작업에 대해 201 생성된 HTTP 상태 코드를 가져옵니다. CRUD 작업이 실패하면 개별 요청이 실패한 이유를 나타내는 400 시리즈 HTTP 상태가 표시됩니다.

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

리소스를 번들 리소스 유형으로 그룹화

새 번들 리소스 유형을 생성하려면 FHIR REST API 요청에 `Bundle`를 지정하고 함께 그룹화하려는 리소스가 포함된 유효한 JSON 본문을 제공해야 합니다.

요청에 번들이 지정되면 JSON 요청 본문의 URL내용이 HealthLake 데이터 스토어에 있는 그대로 저장됩니다. 따라서 개별 리소스 유형에 대해 CRUD 작업을 수행할 수 없습니다. 이 유형의 번들에는 단일 새 리소스 ID가 할당됩니다. 리소스는 있는 그대로 저장되므로 번들 리소스 유형에 저장된 개별 리소스에 대해 `GET` 또는 `POST` 요청을 할 수 없습니다.

> **Note**
>
> HL7 FHIR R4 사양은 그룹, 구성 및 목록을 사용하여 리소스를 그룹화하는 것도 지원합니다. 이러한 리소스 유형을 생성하면 개별 리소스가 직접 포함되지 않습니다. 대신 `Reference` 요소를 사용하여 개별 리소스를 가리킵니다. 따라서 이러한 리소스 유형을 사용하면 리소스 내에 포함된 개별 리소스를 수정할 수 있습니다.

`Bundle` 리소스 유형을 생성하려면 `POST` 요청에 리소스 유형을 지정하고 포함하려는 리소스를 JSON 열거하는를 제공해야 합니다.

Example - **`POST`** 요청을 사용하여 번들 리소스 생성

`bundle` 리소스를 생성하려면 다음을 수행합니다.

1. 다음과 같이 FHIR REST API 요청의 형식을 지정합니다.

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

2. 함께 그룹화할 리소스를 지정하는 JSON 본문을 제공합니다. 이 예제에서는 두 개의 환자 리소스를 그룹화합니다.

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

FHIR REST API 작업을 사용하여 HealthLake 데이터 스토어 검색

HealthLake 는 FHIR 표준의 일부로 제공된 REST API 작업을 사용하여 데이터 스토어 검색을 지원합니다. 이 섹션에서는 여러 가지 리소스 유형에 대해 GET 및 POST 요청을 수행하는 방법의 예를 찾을 수 있습니다.

> **Note**
>
> 개인 식별 정보(PII) 또는 보호 대상 건강 정보(PHI)가 포함된 쿼리의 경우 POST 요청을 사용하는 것이 좋습니다. POST 요청에서 PII 또는 PHI는 요청 본문의 일부로 추가되며 전송 중에 암호화됩니다.

FHIR 사양은 여러 검색 파라미터 유형을 지원하지만 하위 집합만 HealthLake 지원합니다. 자세한 내용은 지원되는 검색 파라미터 유형 및 에서 지원하는 고급 검색 파라미터 HealthLake 단원을 참조하세요.

FHIR REST API 작업을 사용하여 데이터 스토어를 검색합니다.

지원되는 검색 파라미터 유형

다음 표에는에서 지원되는 검색 파라미터 유형이 나와 있습니다 HealthLake.

지원되는 검색 파라미터 유형

검색 파라미터	설명
_id	리소스 ID(전체 아님URL)
_lastUpdated	마지막으로 업데이트된 날짜입니다. 서버에는 경계 정밀도에 대한 재량이 있습니다.
_태그	리소스 태그로 검색합니다.
_profile	프로필로 태그가 지정된 모든 리소스를 검색합니다.
_보안	이 리소스에 적용된 보안 레이블을 검색합니다.

검색 파라미터	설명
_소스	리소스의 출처를 검색합니다.
_텍스트	리소스의 서술을 검색합니다.
createdAt	사용자 지정 확장 -에서 검색합니다createdAt.

> **Note**
>
> 다음 검색 파라미터는 2023년 12월 9일 이후에 생성된 데이터 스토어에 대해서만 지원됩니다. _security, _source, _text, createdAt.

다음 표는 지정된 리소스 유형에 대해 지정된 데이터 유형을 기반으로 쿼리 문자열을 수정하는 방법의 예를 보여줍니다. 명확성을 위해 예제 열의 특수 문자는 인코딩되지 않았습니다. 쿼리를 성공적으로 수행하려면 쿼리 문자열이 올바르게 인코딩되었는지 확인합니다.

파라미터 유형 검색	세부 사항	예시

숫자	지정된 리소스에서 숫자 값을 검색합니다. 중요한 수치가 관찰됩니다.
	
	유효 자릿수는 앞자리 0을 제외하고 검색 파라미터 값에 따라에 고유합니다.
	
	비교 접두사는 허용됩니다.
		`[parameter]=100`
		
		`[parameter]=1e2`
		
		`[parameter]=lt100`

날짜/DateTime	특정 날짜 또는 시간을 검색합니다. 예상 형식은 `yyyy-mm-ddThh:mm:ss[Z|(+|-)hh:mm]` 이지만 다를 수 있습니다.
	
	`date`, , `dateTime`, 및 데이터 형식을 허용합니
		`[parameter]=eq2013-01-14`
		
		`[parameter]=gt2013-01-14T10:00`
		
		`[parameter]=ne2013-01-14`

파라미터 유형 검색	세부 사항	예시
	다instantPeriodTiming. 검색에서 이러한 데이터 유형을 사용하는 방법에 대한 자세한 내용은 FHIR 설명서 인덱스의 날짜를 참조하세요. 비교 접두사는 허용됩니다.	
String	대소문자를 구분하여 문자 시퀀스를 검색합니다. HumanName 및 Address 유형을 모두 지원합니다. 자세한 내용은 FHIR 설명서 인덱스의 HumanName 데이터 형식 항목과 Address 데이터 형식 항목을 참조하세요. 고급 검색은 :text 수정자를 사용하여 지원됩니다.	[base]/Patient?given=eve [base]/Patient?given:contains=eve
토큰	종종 한 쌍의 의료 코드 값과 비교하여 문자열에 대한 일치를 검색합니다 close-to-exact. 대소문자 구분은 쿼리를 생성할 때 사용되는 코드 시스템에 연결됩니다.소비 기반 쿼리는 대소문자 구분과 관련된 문제를 줄이는 데 도움이 될 수 있습니다. 명확성을 위해 |가 인코딩되지 않았습니다.	[parameter]=[system]|[code] : 여기에서[system]는 코딩 시스템을 참조하고, 특정 시스템 내에서 발견된 코드 값을 [code] 참조합니다. [parameter]=[code] : 여기에서 입력은 코드 또는 시스템과 일치합니다. [parameter]=|[code] : 여기에서 입력은 코드와 일치하며 시스템 속성에는 식별자가 없습니다.

파라미터 유형 검색	세부 사항	예시
복합	수정자$ 및 , 작업을 사용하여 단일 리소스 유형 내에서 여러 파라미터를 검색합니다. 비교 접두사는 허용됩니다.	/Patient?language=FR,NL&language=EN Observation?component-code-value-quantity=http://loinc.org|8480-6$lt60 [base]/Group?characteristic-value=gender$mixed
수량	숫자, 시스템 및 코드를 값으로 검색합니다. 숫자는 필수이지만 시스템 및 코드는 선택 사항입니다. 수량 데이터 유형을 기반으로 합니다. 자세한 내용은 FHIR 설명서 인덱스의 수량을 참조하세요. 다음과 같이 가정된 구문을 사용합니다. [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
레퍼런스	다른 리소스에 대한 참조를 검색합니다.	[base]/Observation?subject=Patient/23 test

파라미터 유형 검색	세부 사항	예시
URI	특정 리소스를 명확하게 식별하는 문자열을 검색합니다.	`[base]/ValueSet?url=http://acme.org/fhir/ValueSet/123`
특별	통합 의료 NLP 확장을 기반으로 검색합니다.	

에서 지원하는 고급 검색 파라미터 HealthLake

HealthLake 는 다음과 같은 고급 검색 파라미터를 지원합니다.

명칭	설명	예제	기능
`_includ`	검색 요청에 추가 리소스를 반환하도록 요청하는 데 사용됩니다. 대상 리소스 인스턴스에서 참조하는 리소스를 반환합니다.	`Encounter?_include=Encounter:subject`	
`_revinc de`	검색 요청에 추가 리소스를 반환하도록 요청하는 데 사용됩니다. 기본 리소스 인스턴스를 참조하는 리소스를 반환합니다.	`Patient?_id= **patient-identifier** &_revinclude=Encounter:patient`	
`_summar`	요약을 사용하여 리소스의 하위 집합을 요청할 수 있습니다.	`Patient?_summary=text`	지원되는 요약 파라미터는 `_summary=true` , `_summary=false` , `_summary=text` , 입니다`_summary=data` .
`_elemen`	검색 결과에서 리소스의 일부로 반환할 특정 요소 세트를 요청합니다.	`Patient?_elements=`	

명칭	설명	예제	기능
		identifier,active,link	
_total	검색 파라미터와 일치하는 리소스 수를 반환합니다.	Patient?_total=accurate	_total=accurate ,를 지원합니다_total=none .
_sort	쉼표로 구분된 목록을 사용하여 반환된 검색 결과의 정렬 순서를 지정합니다. - 접두사는 쉼표로 구분된 목록의 모든 정렬 규칙에 사용할 수 있으며 내림차순을 나타냅니다.	Observation?_sort=status,-date	유형이 인 필드별 정렬을 지원합니다Number, String, Quantity, Token, URI, Reference . 정렬 기준 Date은 2023년 12월 9일 이후에 생성된 데이터 스토어에 대해서만 지원됩니다. 최대 5개의 정렬 규칙을 지원합니다.
_count	검색 번들의 페이지당 반환되는 리소스 수를 제어합니다.	Patient?_count=100	최대 페이지 크기는 100입니다.
chaining	참조된 리소스의 요소를 검색합니다. 는 연결된 검색을 참조된 리소스 내의 요소로 . 전달합니다.	DiagnosticReport?subject:Patient.name=peter	
reverse chaining (_has)	리소스를 참조하는 리소스 요소를 기반으로 리소스를 검색합니다.	Patient?_has:Observation:patient:code=1234-5	

_include

검색 쿼리_include에서를 사용하면 지정된 추가 FHIR 리소스도 반환할 수 있습니다. _include를 사용하여 앞으로 연결된 리소스를 포함합니다.

Example -를 **_include** 사용하여 환자 또는 기침 진단을 받은 환자 그룹을 찾으려면

진단 코드에서 기침을 지정하는 Condition 리소스 유형을 검색한 다음 해당 진단subject의 도 반환하도록 _include 지정하는를 사용합니다. Condition 리소스 유형에서 환자 리소스 유형 또는 그룹 리소스 유형을 subject 나타냅니다.

명확성을 위해 예제의 특수 문자는 인코딩되지 않았습니다. 쿼리를 성공적으로 수행하려면 쿼리 문자열이 올바르게 인코딩되었는지 확인합니다.

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

_revinclude

검색 쿼리_revinclude에서를 사용하면에서 추가로 지정된 FHIR 리소스도 반환할 수 있습니다. _revinclude를 사용하여 역방향으로 연결된 리소스를 포함합니다.

Example - **_revinclude**를 사용하여 특정 환자와 연결된 관련 상황 및 관찰 리소스 유형을 포함하려면

이 검색을 수행하려면 먼저 _id 검색 파라미터에 식별자를 지정Patient하여 개인을 정의해야 합니다. 그런 다음 구조 Encounter:patient 및를 사용하여 추가 FHIR 리소스를 지정합니다Observation:patient.

명확성을 위해 예제의 특수 문자는 인코딩되지 않았습니다. 쿼리를 성공적으로 수행하려면 쿼리 문자열이 올바르게 인코딩되었는지 확인합니다.

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

_summary

검색 쿼리_summary에서를 사용하면 사용자가 FHIR 리소스의 하위 집합을 요청할 수 있습니다. 다음 값 중 하나를 포함할 수 있습니다true, text, data, false. 다른 모든 값은 유효하지 않은 것으로 처리됩니다. 반환된 리소스는 meta.tag 'SUBSETTED' 에 로 표시되어 리소스가 불완전함을 나타냅니다.

- true: 리소스의 기본 정의에서 'summary(요약)'로 표시된 지원되는 모든 요소를 반환합니다.

- `text`: '텍스트', 'id', '메타' 요소만 반환하고 최상위 필수 요소만 반환합니다.
- `data`: '텍스트' 요소를 제외한 모든 부분을 반환합니다.
- `false`: 리소스(들)의 모든 부분 반환

단일 검색 요청에서는 `_include` 또는 `_revinclude` 검색 파라미터와 결합할 수 `_summary=text` 없습니다.

Example - 데이터 스토어에서 환자 리소스의 “텍스트” 요소를 가져옵니다.

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

_elements

검색 쿼리`_elements`에서를 사용하면 특정 FHIR 리소스 요소를 요청할 수 있습니다. 반환된 리소스는 meta.tag`'SUBSETTED'`에 로 표시되어 리소스가 불완전함을 나타냅니다.

`_elements` 파라미터는 리소스의 루트 수준에서 정의된 요소와 같이 쉼표로 구분된 기본 요소 이름 목록으로 구성됩니다. 나열된 요소만 반환됩니다. `_elements` 파라미터 값에 잘못된 요소가 포함된 경우 서버는 해당 요소를 무시하고 필수 요소와 유효한 요소를 반환합니다.

`_elements`는 포함된 리소스(검색 모드가 인 반환된 리소스)에는 적용되지 않습니다`include`.

단일 검색 요청에서는를 `_summary` 검색 파라미터와 결합할 수 `_elements` 없습니다.

Example – HealthLake 데이터 스토어에서 환자 리소스의 “식별자”, “활성”, “링크” 요소를 가져옵니다.

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

_total

검색 쿼리`_total`에서를 사용하면 요청된 검색 파라미터와 일치하는 리소스 수가 반환됩니다. HealthLake 는 검색 응답의에서 일치하는 리소스(검색 모드가 인 반환된 리소스match)`Bundle.total`의 총 수를 반환합니다.

`_total`는 `accurate`, `none` 파라미터 값을 지원합니다. `_total=estimate`는 지원되지 않습니다. 다른 모든 값은 유효하지 않은 것으로 처리됩니다. `_total`는 포함된 리소스(검색 모드가 인 반환된 리소스)에 적용되지 않습니다`include`.

Example - 데이터 스토어의 총 환자 리소스 수를 가져옵니다.

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

_sort

검색 쿼리_sort에서를 사용하면 결과가 특정 순서로 정렬됩니다. 결과는 쉼표로 구분된 정렬 규칙 목록을 우선 순위로 정렬됩니다. 정렬 규칙은 유효한 검색 파라미터여야 합니다. 다른 모든 값은 유효하지 않은 것으로 처리됩니다.

단일 검색 요청에서 최대 5개의 정렬 검색 파라미터를 사용할 수 있습니다. 필요에 따라 - 접두사를 사용하여 내림차순을 표시할 수 있습니다. 서버는 기본적으로 오름차순으로 정렬됩니다.

지원되는 정렬 검색 파라미터 유형은 입니다Number, String, Date, Quantity, Token, URI, Reference. 검색 파라미터가 중첩된 요소를 참조하는 경우이 검색 파라미터는 정렬에 지원되지 않습니다. 예를 들어, 리소스 유형의 '이름'에 대한 검색 환자는 HumanName 데이터 형식이 중첩된 것으로 간주되는 Patient.name 요소를 나타냅니다. 따라서 '이름'을 기준으로 환자 리소스를 정렬하는 것은 지원되지 않습니다.

Example - 데이터 스토어에서 환자 리소스를 가져오고 생년월일을 기준으로 오름차순으로 정렬합니다.

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

_count

파라미터_count는 단일 페이지에서 반환해야 하는 리소스 수에 대한 서버 지침으로 정의됩니다.

최대 페이지 크기는 100입니다. 100보다 큰 값은 유효하지 않습니다. _count=0는 지원되지 않습니다.

Example – 환자 리소스를 검색하고 검색 페이지 크기를 25로 설정합니다.

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

Chaining and Reverse Chaining(_has)

의 체인 및 리버스 체인은 상호 연결된 데이터를 더 효율적이고 컴팩트하게 얻을 수 있는 방법을 FHIR 제공하므로 여러 별도의 쿼리가 필요하지 않으며 개발자와 사용자가 데이터 검색을 더 편리하게 할 수 있습니다.

재귀 수준이 100개 이상의 결과를 반환하는 경우 HealthLake 는 4xx를 반환하여 데이터 스토어가 오버로드되어 여러 페이지 매김이 발생하지 않도록 보호합니다.

Example - 연결 - 환자 이름이 피터인 환자를 DiagnosticReport 참조하는 모든 항목을 가져옵니다.

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

Example - 역방향 연결 - 환자 리소스 가져오기 - 하나 이상의 관찰에서 환자 리소스를 참조합니다. 여기서 관찰의 코드는 1234이고 관찰은 환자 검색 파라미터의 환자 리소스를 참조합니다.

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

지원되는 검색 수정자

검색 수정자는 문자열 기반 필드에 사용됩니다. 의 모든 검색 수정자는 부울 기반 로직을 HealthLake 사용합니다. 예를 들어, 검색 결과에 포함`:contains`되도록 더 큰 문자열 필드에 작은 문자열을 포함하도록 지정할 수 있습니다.

지원되는 검색 수정자

검색 수정자	유형
:missing	다음을 제외한 모든 파라미터 `Composite`
:exact	String
: 포함	String
:not	토큰
:text	토큰

검색 수정자	유형
:identifier	레퍼런스

지원되는 검색 비교기

검색 비교기를 사용하여 검색에서 일치하는의 특성을 제어할 수 있습니다. 번호, 날짜 및 수량 필드를 검색할 때 비교기를 사용할 수 있습니다. 다음 표에는에서 지원하는 검색 비교기 및 해당 정의가 나열되어 있습니다 HealthLake.

지원되는 검색 비교기

검색 비교자	설명
eq	리소스의 파라미터 값이 제공된 값과 같습니다.
ne	리소스의 파라미터 값이 제공된 값과 같지 않습니다.
gt	리소스의 파라미터 값이 제공된 값보다 큽니다.
lt	리소스의 파라미터 값이 제공된 값보다 작습니다.
ge	리소스의 파라미터 값이 제공된 값보다 크거나 같습니다.
le	리소스의 파라미터 값이 제공된 값보다 작거나 같습니다.
sa	리소스의 파라미터 값은 제공된 값 이후에 시작됩니다.
eb	리소스의 파라미터 값은 제공된 값보다 먼저 종료됩니다.

에서 지원되지 않는 검색 파라미터 HealthLake

지원되는 검색 파라미터의 전체 목록은 FHIR 검색 파라미터 레지스트리를 참조하세요. HealthLake 테이블에 나열된 파라미터를 제외한 모든 검색 파라미터를 지원합니다.

지원되지 않는 검색 파라미터

번들 구성	위치-근처
번들 식별자	Consent-source-reference
번들 메시지	계약 환자
번들 유형	Resource-content
번들 타임스탬프	리소스 쿼리

POST 예제로 검색

POST 요청을 통해 HealthLake 데이터 스토어를 검색할 수 있습니다. URI 또는 요청 본문에서 쿼리 파라미터를 제공할 수 있지만 단일 요청에서 둘 다 사용할 수는 없습니다.

이 주제의 예제는 모범 사례를 따릅니다.

> **Note**
>
> 개인 식별 정보(PII) 또는 보호 대상 건강 정보(PHI)가 포함된 쿼리의 경우 POST 요청을 사용하는 것이 좋습니다. POST 요청에서 PII 또는 PHI는 요청 본문의 일부로 추가되며 전송 중에 암호화됩니다.

POST 요청 본문에 파라미터를 사용하여 요청할 때는를 헤더의 `Content-Type: application/x-www-form-urlencoded` 일부로 사용합니다.

이 주제에서는 다음 리소스 유형을 사용하여 POST를 검색하는 방법의 예를 제공합니다.

- Age: Age는에서 정의된 리소스 유형이 아닙니다FHIR. 대신 연령은 환자 리소스 유형의 일부로 캡처됩니다. 특정 연령 또는 연령 범위에 따라 환자 그룹을 검색하려면를 사용합니다the section called “지원되는 검색 비교기”. 자세한 내용은 FHIR 설명서 인덱스의 리소스 유형: 환자를 참조하세요.

- 조건:이 리소스 유형은 진단, 상황, 임상 조건 및 우려 수준으로 상승한 문제와 같은 임상 개념과 관련된 세부 정보를 저장합니다. 자세한 내용은 FHIR 설명서 인덱스의 리소스 유형: 조건을 참조하세요. 에서 찾을 수 있는 문서를 기반으로 새 조건을 HealthLake 생성합니다 DocumentReference. 이러한 추가 사항은 기본적으로 POST 요청 시 제외됩니다. 이를 포함하려면 검색에서 조건 리소스의 유효한 식별자를 지정해야 합니다.
- DocumentReference:이 리소스 유형은에서 지원됩니다 HealthLake. 이 리소스 유형은 모든 유형의 문서 참조를 지원합니다. 자세한 내용은 FHIR 설명서 인덱스의 리소스 유형: DocumentReference을 참조하세요. HealthLake 또한에서 찾을 수 있는 문서의 통합 자연어 처리(NLP)를 제공합니다 DocumentReference. 자세한 내용은 에서 리소스 유형의 자연어 처리(NLP)를 기반으로 자동화된 FHIR DocumentReference 리소스 생성 사용 AWS HealthLake을 참조하십시오.
- 위치:이 리소스 유형에는 부수적 위치(사전 지정 또는 승인 없이 의료에 사용되는 위치)와 공식적으로 지정된 전용 위치가 모두 포함됩니다. 자세한 내용은 FHIR 설명서 인덱스의 리소스 유형: 위치를 참조하세요.
- 관찰: 환자, 디바이스 또는 기타 주제에 대한 측정 및 간단한 어설션입니다.는 리소스에 있는 문서를 기반으로 새 관찰 DocumentReference 리소스를 HealthLake 생성합니다. 가 새 리소스를 HealthLake 생성하는 방법에 대한 자세한 내용은 섹션을 참조하세요에서 리소스 유형의 자연어 처리(NLP)를 기반으로 자동화된 FHIR DocumentReference 리소스 생성 사용 AWS HealthLake. 이러한 추가 사항은 POST 요청 시 기본적으로 제외됩니다. 이를 포함하려면 검색에서 관측 리소스의 유효한 식별자를 지정해야 합니다. 자세한 내용은 FHIR 설명서 인덱스의 리소스 유형: 관찰을 참조하세요.

각 탭은 지정된 리소스 유형을 검색하는 방법의 예를 보여줍니다. 여기에는 요청 본문에서 요청을 지정하는 방법의 예가 포함되어 있습니다.

Age

다음을 사용하여 `Patient` 리소스 유형에 대한 POST기반 검색 요청을 수행합니다. 이 검색은 `eq` 검색 비교기를 사용하여 1997년에 태어난 개인을 검색합니다.

요청URL과 요청 본문을 지정해야 합니다. 다음은 요청의 예입니다URL.

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

검색에서 1997년을 지정하려면 요청 본문에 다음 요소를 추가합니다.

```
birthdate=eq1997
```

JSON 응답

성공하면 200 HTTP 응답 코드와 유사한 JSON 응답을 받게 됩니다.

Condition

다음을 사용하여 Condition 리소스 유형을 POST 요청합니다. 이 검색은 HealthLake 데이터 스토어에서 의료 코드가 포함된 위치를 찾습니다72892002.

요청URL과 요청 본문을 지정해야 합니다. 다음은 요청의 예입니다URL.

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

검색하려는 의료 코드를 지정하려면 요청 본문에이 JSON 요소를 추가합니다.

```
code=72892002
```

JSON 응답

성공하면 200 HTTP 응답 코드를 받게 됩니다. 명확성을 위해 다음 JSON 응답이 잘렸습니다.

```
{
 "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

`DocumentReference` 리소스 유형에 대해 POST 요청할 때 HealthLake의 통합 자연어 처리 결과(NLP)를 보려면 다음과 같이 요청 형식을 지정합니다.

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

참조할 DocumentReference 요소를 지정하려면 섹션을 참조하세요검색 파라미터. 요청 본문의를로 지정합니다JSON.

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

이 쿼리 문자열은 여러 검색 파라미터를 사용하여 통합된 의료 NLP 결과를 생성하는 데 사용되는 Amazon Comprehend Medical API 작업을 검색합니다.

Location

다음을 사용하여 `Location` 리소스 유형을 POST 요청합니다. 이 검색은 주소의 일부로 도시 이름 Boston이 포함된 HealthLake 데이터 스토어의 위치를 찾습니다.

요청URL과 요청 본문을 지정해야 합니다. 다음은 요청의 예입니다URL.

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

검색에서 보스턴을 지정하려면 요청 본문에 다음 요소를 추가합니다.

```
address=Boston
```

JSON 응답

성공하면 200 HTTP 응답 코드를 받게 됩니다. 명확성을 위해 JSON 응답이 잘렸습니다.

```
{
 "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

다음을 사용하여 `Observation` 리소스 유형에 대한 `POST`기반 검색 요청을 수행합니다. 이 검색은 `value-concept` 검색 파라미터를 사용하여 의료 코드를 찾습니다`266919005`. 이 상태는를 나타냅니다`Never smoker`.

요청URL과 요청 본문을 지정해야 합니다. 다음은 요청의 예입니다URL.

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

상태를 지정하려면 `Never smoker`의 본문`value-concept=266919005`에를 설정합니다JSON.

```
value-concept=266919005
```

JSON 응답

성공하면 `200` HTTP 응답 코드를 받게 됩니다. 명확성을 위해 다음 JSON 응답이 잘렸습니다.

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

GET 예제로 검색

GET 요청하여 HealthLake 데이터 스토어를 검색할 수 있습니다.는 요청 본문의 일부가 URI아닌의 일부로 쿼리 파라미터를 제공하는 것 HealthLake 만 지원합니다.

> **Note**
>
> 개인 식별 정보(PII) 또는 보호 대상 건강 정보(PHI)가 포함된 쿼리의 경우 POST 요청을 사용하는 것이 좋습니다. POST 요청에서 PII 또는 PHI는 요청 본문의 일부로 추가되며 전송 중에 암호화됩니다.

이 주제에서는에서 지원되는 리소스 유형을 GET 사용하여를 검색하는 방법의 예를 제공합니다 HealthLake.

- Age: Age는에서 정의된 리소스 유형이 아닙니다FHIR. 대신 연령은 환자 리소스 유형의 일부로 캡처됩니다. 특정 연령 또는 연령 범위에 따라 환자 그룹을 검색하려면를 사용해야 합니다the section called “지원되는 검색 비교기”. 자세한 내용은 FHIR 설명서 인덱스의 리소스 유형: 환자를 참조하세요.
- 조건:이 리소스 유형은 진단, 상황, 임상 조건 및 우려 수준으로 상승한 문제와 같은 임상 개념과 관련된 세부 정보를 저장합니다. 자세한 내용은 FHIR 설명서 인덱스의 리소스 유형: 조건을 참조하세요. 에서 찾을 수 있는 문서를 기반으로 새 조건을 HealthLake 생성합니다 DocumentReference. 이러한 추가 사항은 기본적으로 POST 요청 시 제외됩니다. 이를 포함하려면 검색에서 조건 리소스에 유효한 식별자를 지정해야 합니다.
- DocumentReference:이 리소스 유형은에서 지원됩니다 HealthLake. 이 리소스 유형은 모든 유형의 문서 참조를 지원합니다. 자세한 내용은 FHIR 설명서 인덱스의 리소스 유형: DocumentReference을 참조하세요. HealthLake 또한에서 찾을 수 있는 문서의 통합 자연어 처리(NLP)를 제공합니다 DocumentReference. 자세한 내용은 에서 리소스 유형의 자연어 처리(NLP)를 기반으로 자동화된 FHIR DocumentReference 리소스 생성 사용 AWS HealthLake을 참조하십시오.
- 위치:이 리소스 유형에는 부수적 위치(사전 지정 또는 승인 없이 의료에 사용되는 위치)와 공식적으로 지정된 전용 위치가 모두 포함됩니다. 자세한 내용은 FHIR 설명서 인덱스의 리소스 유형: 위치를 참조하세요.
- 관찰: 환자, 디바이스 또는 기타 주제에 대한 측정 및 간단한 어설션입니다.는 리소스에 있는 문서를 기반으로 새 관찰 DocumentReference 리소스를 HealthLake 생성합니다. 새 리소스를 HealthLake 생성하는 방법에 대한 자세한 내용은 섹션을 참조하세요에서 리소스 유형의 자연어 처리(NLP)를 기반으로 자동화된 FHIR DocumentReference 리소스 생성 사용 AWS HealthLake. 이러한 추가 사항은 POST 요청 시 기본적으로 제외됩니다. 이를 포함하려면 검색에서 관측 리소스의 유효한 식별자를 지정해야 합니다. 자세한 내용은 FHIR 설명서 인덱스의 리소스 유형: 관찰을 참조하세요.

각 탭은 지정된 리소스 유형을 검색하는 방법의 예를 보여줍니다. 여기에는에서 요청을 지정하는 방법의 예URI와 관련 JSON 응답이 포함되어 있습니다.

Age

다음을 사용하여 Patient 리소스 유형에 대한 GET기반 검색 요청을 수행합니다. 이 검색은 eq 검색 비교기를 사용하여 1997년에 태어난 개인을 검색합니다.

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

JSON 응답

성공하면 200 HTTP 응답 코드를 받게 됩니다.

Condition

다음을 사용하여 Condition 리소스 유형을 GET 요청합니다. 이 검색은 HealthLake 데이터 스토어에서 의료 코드가 포함된 위치를 찾습니다72892002.

요청URL과 요청 본문을 지정해야 합니다. 다음은 요청의 예입니다URL.

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

JSON 응답

성공하면 200 HTTP 응답 코드를 받게 됩니다. 명확성을 위해 다음 JSON 응답이 잘렸습니다.

```
{
 "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

이 예제에서는 연쇄상 구균 진단을 받고 아목시실린도 처방받은 환자의 DocumentReference 리소스 유형에 대한 검색 요청을 생성하는 방법을 보여줍니다.

```
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
```

성공하면 다음 JSON 응답을 받게 됩니다.

```
{
    "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

다음을 사용하여 Location 리소스 유형을 GET 요청합니다. 이 검색은 주소의 일부로 도시 이름 Boston이 포함된 HealthLake 데이터 스토어의 위치를 찾습니다.

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

JSON 응답

성공하면 200 HTTP 응답 코드를 받게 됩니다. 명확성을 위해 JSON 응답이 잘렸습니다.


```
{
    "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

다음을 사용하여 Observation 리소스 유형에 대한 GET기반 검색 요청을 수행합니다. 이 검색은 value-concept 검색 파라미터를 사용하여 의료 코드를 찾습니다266919005. 이 상태는를 나타냅니다Never smoker.

요청URL과 쿼리 문자열을 지정해야 합니다. 다음은 요청의 예입니다URL.

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

상태를 지정하려면를 쿼리 문자열value-concept=266919005로 Never smoker설정합니다.

JSON 응답

성공하면 200 HTTP 응답 코드를 받게 됩니다. 명확성을 위해 다음 JSON 응답이 잘렸습니다.

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

FHIR 리소스 기록 읽기

상호 FHIR `history` 작용은 HealthLake 데이터 스토어의 특정 FHIR 리소스 기록을 검색합니다. 이 상호 작용을 사용하여 시간이 지남에 따라 FHIR 리소스의 내용이 어떻게 변경되었는지 확인할 수 있습니다. 또한 감사 로그와 협력하여 수정 전후에 리소스의 상태를 확인하는 데 유용합니다.

> **Note**
>
> FHIR 리소스`history`는 기본적으로 10/25/2024 이후에 생성된 모든 HealthLake 데이터 스토어에서 활성화됩니다. 이 날짜 이전에 데이터 스토어가 생성된 경우 지원 티켓을 제출하여 FHIR `history` 상호 작용을 활성화할 수 있습니다. 를 사용하여 사례를 생성합니다AWS Support Center Console. 사례를 생성하려면에 로그인 AWS 계정 하고 사례 생성을 선택합니다.

상호 `history` 작용은 `HTTP GET` 명령을 사용하여 수행됩니다. FHIR 상호 작용 `update`, 및 `create`는 저장될 리소스의 기록 버전을 `delete` 생성합니다.는 FHIR `history` 상호 작용에 대해 다음 검색 파라미터를 HealthLake 지원합니다.

HealthLake FHIR **`history`** 상호 작용에 지원되는 검색 파라미터

검색 파라미터	설명
`_count : integer`	페이지의 최대 검색 결과 수입니다. 서버는 요청된 수 또는 데이터 스토어에 대해 기본적으로 허용되는 최대 검색 결과 수 중 더 적은 수를 반환합니다.
`_since : instant`	지정된 시간에 또는 그 이후에 생성된 리소스 버전만 포함합니다.
`_at : date(Time)`	날짜 시간 값에 지정된 기간 동안 특정 시점에 최신 상태였던 리소스 버전만 포함합니다. 자세한 내용은 HL7 FHIR RESTful API 설명서`date`의 섹션을 참조하세요.

다음 예제에서는의 FHIR `Patient` 리소스에 대한 페이지당 100개의 과거 검색 결과를 반환합니다 HealthLake. 전체 URL 경로를 보려면 복사 버튼을 스크롤합니다. 의 형식URL은 다음과 같습니다.

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

기록 상호 작용의 반환 콘텐츠는 Bundle유형이 로 설정된 FHIR 리소스에 포함됩니다history. 지정된 버전 기록이 포함되어 있으며, 가장 오래된 버전이 마지막으로 정렬되어 있고 삭제된 리소스가 포함되어 있습니다. history 상호 작용에 대한 자세한 내용은 HL7 FHIR RESTful API 설명서history의 섹션을 참조하세요.

> **Note**
>
> 특정 FHIR 리소스 유형에 history 대해 옵트아웃할 수 있습니다. 옵트아웃하려면를 사용하여 사례를 생성합니다AWS Support Center Console. 사례를 생성하려면에 로그인 AWS 계정하고 사례 생성을 선택합니다.

버전별 FHIR 리소스 기록 읽기

상호 FHIR vread 작용은 HealthLake 데이터 스토어에서 리소스의 버전별 읽기를 수행합니다. 이 상호 작용을 사용하면 과거 특정 시간에 있었던 FHIR 리소스의 내용을 볼 수 있습니다.

HealthLake 는 지원되는 CapabilityStatement.rest.resource.versioning 각 리소스에 대해에서 버전 관리를 지원합니다. 모든 HealthLake 데이터 스토어에는 모든 리소스에 Resource.meta.versionId (vid)가 포함됩니다.

FHIR history 상호 작용이 활성화된 경우(10/25/2024 이후에 생성된 데이터 스토어의 경우 기본적으로 또는 이전 데이터 스토어의 경우 요청에 따라) Bundle 응답에는가의 vid 일부로 포함됩니다location. 다음 예제에서는가 숫자 로 vid 표시됩니다1. 전체 예제를 보려면 번들/번들 응답 예제 (JSON)를 참조하세요.

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

상호 vread 작용은 HTTP GET 명령을 사용하여 수행됩니다. 다음 vread 상호 작용은에서 지정한 FHIR Patient 리소스 메타데이터 버전의 리소스에 지정된 콘텐츠가 있는 단일 인스턴스를 반환합니다vid. 다음 예제에서 전체 URL 경로를 보려면 복사 버튼을 스크롤합니다. 의 형식URL은 다음과 같습니다.

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

> **Note**
>
> FHIR 리소스를 읽을 vread 때 없이 history 상호 작용을 사용하는 경우는 HealthLake 항상 리소스 메타데이터의 최신 버전을 반환합니다.

vread 상호 작용에 대한 자세한 내용은 HL7 FHIR Restful API 설명서vread의 섹션을 참조하세요.

환자 $everything FHIR REST API 작업으로 환자 데이터 가져오기

Patient $everything 작업은 해당 FHIR 환자와 관련된 다른 리소스와 함께 환자 리소스를 쿼리하는 데 사용됩니다. 이 작업은 환자에게 전체 레코드에 대한 액세스 권한을 제공하거나 공급자가 환자와 관련된 대량 데이터 다운로드를 수행하는 데 사용할 수 있습니다.는 특정 환자 ID에 대해 $everything을 HealthLake 지원합니다.

> **Note**
>
> Patient $everything 작업은 현재 2024년 2월 27일 이후에 생성된 데이터 스토어에서 지원됩니다.

환자와 관련된 모든 리소스 가져오기

환자 $everything은 아래 예제와 같이 호출할 수 있는 REST API 작업입니다.

GET Request

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

> **Note**
>
> 응답 리소스는 리소스 유형 및 리소스 ID별로 정렬됩니다.

응답은 항상 Bundle.total으로 채워집니다.

환자 $everything 파라미터

HealthLake 는 다음 쿼리 파라미터를 지원합니다.

파라미터	세부 사항
시작	지정된 시작 날짜 이후에 모든 환자 데이터를 가져옵니다.
end	지정된 종료일 이전에 모든 환자 데이터를 가져옵니다.
since	지정된 날짜 이후에 모든 환자 데이터를 업데이트합니다.
_유형	특정 리소스 유형에 대한 환자 데이터를 가져옵니다.
_개수	환자 데이터를 가져오고 페이지 크기를 지정합니다.

Example - 지정된 시작 날짜 이후의 모든 환자 데이터 가져오기

환자 $everything은 `start` 필터를 사용하여 특정 날짜 이후의 데이터만 쿼리할 수 있습니다.

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

Example - 지정된 종료일 이전에 모든 환자 데이터 가져오기

환자 $everything은 end 필터를 사용하여 특정 날짜 이전의 데이터만 쿼리할 수 있습니다.

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

Example - 지정된 날짜 이후에 모든 환자 데이터를 업데이트합니다.

환자 $everything은 `since` 필터를 사용하여 특정 날짜 이후에 업데이트된 데이터만 쿼리할 수 있습니다.

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

Example - 특정 리소스 유형에 대한 환자 데이터 가져오기

환자 $everything은 _type 필터를 사용하여 응답에 포함할 특정 리소스 유형을 지정할 수 있습니다. 쉼표로 구분된 목록에서 여러 리소스 유형을 지정할 수 있습니다.

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

Example - 환자 데이터 가져오기 및 페이지 크기 지정

환자 $everything은 _count를 사용하여 페이지 크기를 설정할 수 있습니다.

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

환자 $모든 것 **start** 및 **end** 속성

HealthLake 는 시작 및 종료 쿼리 파라미터에 대해 다음과 같은 리소스 속성을 지원합니다.

Resource	리소스 요소
계정	계정.servicePeriod.시작
AdverseEvent	AdverseEvent.date
AllergyIntolerance	AllergyIntolerance.recordedDate
예약	Appointment.start
AppointmentResponse	AppointmentResponse.start

Resource	리소스 요소
AuditEvent	AuditEvent.period.start
기본	Basic.created
BodyStructure	NO_DATE
CarePlan	CarePlan.period.start
CareTeam	CareTeam.period.start
ChargeItem	ChargeItem.occurrenceDateTime, ChargeItem.occurrencePeriod.start, ChargeItem.occurrenceTiming.event
Claim	Claim.billablePeriod.start
ClaimResponse	ClaimResponse.created
ClinicalImpression	ClinicalImpression.date
통신	Communication.sent
CommunicationRequest	CommunicationRequest.occurrenceDateTime, CommunicationRequest.occurrencePeriod.시작
구성	Composition.date
Condition	조건.recordedDate
동의	동의.dateTime
적용 범위	Coverage.period.start
CoverageEligibilityRequest	CoverageEligibilityRequest.created

Resource	리소스 요소
CoverageEligibilityResponse	CoverageEligibilityResponse.created
DetectedIssue	DetectedIssue.식별됨
DeviceRequest	DeviceRequest.authoredOn
DeviceUseStatement	DeviceUseStatement.recordedOn
DiagnosticReport	DiagnosticReport.effective
DocumentManifest	DocumentManifest.created
DocumentReference	DocumentReference.context.period.start
상황	Encounter.period.start
EnrollmentRequest	EnrollmentRequest.created
EpisodeOfCare	EpisodeOfCare.period.start
ExplanationOfBenefit	ExplanationOfBenefit.billablePeriod.시작
FamilyMemberHistory	NO_DATE
플래그	Flag.period.start

Resource	리소스 요소
목표	목표.statusDate
그룹	NO_DATE
ImagingStudy	ImagingStudy.started
예방 접종	Immunization.recorded
ImmunizationEvaluation	ImmunizationEvaluation.date
ImmunizationRecommendation	ImmunizationRecommendation.date
인보이스	Invoice.date
나열	List.date
MeasureReport	MeasureReport.period.start
미디어	Media.issued
MedicationAdministration	MedicationAdministration.effective
MedicationDispense	MedicationDispense.whenPrepared
MedicationRequest	MedicationRequest.authoredOn
MedicationStatement	MedicationStatement.dateAsserted

Resource	리소스 요소
Molecular Sequence	NO_DATE
Nutrition Order	NutritionOrder.dateTime
관측치	Observation.effective
환자	NO_DATE
Person	NO_DATE
절차	Procedure.performed
증명	Provenance.occurredPeriod.start, Provenance.occurredDateTime
QuestionnaireResponse	QuestionnaireResponse.authored
RelatedPerson	NO_DATE
RequestGroup	RequestGroup.authoredOn
ResearchSubject	ResearchSubject.기간
RiskAssessment	RiskAssessment.occurrenceDateTime, RiskAssessment.occurrencePeriod.시작
일정	일정.planningHorizon
ServiceRequest	ServiceRequest.authoredOn
표본	표본.receivedTime

Resource	리소스 요소
SupplyDelivery	SupplyDelivery.occurrenceDateTime, SupplyDelivery.occurrencePeriod.start, SupplyDelivery.occurrenceTiming.event
SupplyRequest	SupplyRequest.authoredOn
VisionPrescription	VisionPrescription.dateWritten

$export를 사용하여 데이터 스토어에서 HealthLake 데이터 내보내기

요청의 `$export` 일부로 FHIR REST API 지정을 사용하여 내보내기 `POST` 요청을 수행하고 요청 본문에 요청 파라미터를 포함합니다. FHIR 사양에 따라 FHIR 서버는 `GET` 요청을 지원해야 하며 `POST` 요청을 지원할 수 있습니다. 추가 파라미터를 지원하려면 내보내기를 시작하는 데 본문이 필요하므로에서 POST 요청을 HealthLake 지원합니다.

> **Important**
>
> HealthLake 2023년 6월 1일 이전에 생성된 데이터 스토어는 시스템 전체 내보내기에 대한 FHIR REST API 기반 내보내기 작업 요청만 지원합니다.
> HealthLake 2023년 6월 1일 이전에 생성된 데이터 스토어는 데이터 스토어의 엔드포인트에서 `GET` 요청을 사용하여 내보내기 상태를 가져오는 것을 지원하지 않습니다.

를 사용하여 수행하는 모든 내보내기 요청은 `ndjson` 형식으로 반환되고 Amazon S3 버킷으로 내보내FHIRRESTAPI집니다. 각 S3 객체에는 단일 FHIR 리소스 유형만 포함됩니다.

한 번에 각 AWS 계정에 대해 단일 내보내기 요청을 할 수 있습니다. 와 연결된 Service Quotas에 대한 자세한 내용은 섹션을 HealthLake참조하세요AWS HealthLake 엔드포인트 및 할당량.

를 사용하여 내보내기를 요청하는 방법에 대한 자세한 내용은 섹션을 FHIR REST API참조하세요FHIR REST API 작업을 사용하여 데이터 스토어에서 HealthLake 데이터 내보내기.

Amazon AthenaSQL에서를 사용하여 AWS HealthLake 데이터 스토어 쿼리

HealthLake 데이터 스토어를 생성하면 고도로 중첩된 FHIR 데이터 구조가 Amazon Athena에 수집되고를 사용하여 쿼리 가능한 Iceberg 테이블로 자동 변환됩니다SQL. 이 새 리소스에 대한 액세스 권한 부여는 AWS Lake Formation을 사용하여 관리됩니다. 각 FHIR 리소스 유형은 Athena에서 개별 테이블로 표시됩니다.

> **Important**
>
> 2022년 11월 14일 이전에 생성된 데이터 스토어의 경우를 사용하여 쿼리하려면 기존 데이터 스토어를 새 데이터 스토어로 마이그레이션해야 합니다SQL. 도움말은 Amazon Athena를 사용하도록 기존 데이터 스토어 마이그레이션를 참조하십시오.

> **Note**
>
> 2023년 2월 20일 이후 HealthLake 데이터 스토어는 기본적으로 통합 자연어 처리(NLP)를 사용하지 않습니다. 데이터 스토어에서이 기능을 활성화하려면 문제 해결 HealthLake의 통합 자연어 처리 기능을 켜려면 어떻게 해야 합니까? 장의 섹션을 참조하세요.

HealthLake 데이터 스토어를 생성하려면 HealthLake 관리자인 IAM 사용자 또는 역할에 추가 IAM 정책 및 서비스 역할을 추가해야 합니다. 권한 설정에 대한 자세한 내용은 섹션을 참조하세요사용을 시작할 권한 설정 AWS HealthLake.

HealthLake 데이터 스토어는 Athena에 Iceberg 테이블로 수집됩니다. Athena에서 Iceberg 테이블이 작동하는 방법에 대한 자세한 내용은 Athena 사용 설명서의 Iceberg 테이블 사용을 참조하세요.

HealthLake 는 Athena의 HealthLake 데이터 스토어 데이터 스토어 READ 작업을 지원합니다. FHIR REST API 작업을 사용한 생성, 읽기, 업데이트 및 삭제(CRUD) 작업에 대한 자세한 내용은 섹션을 참조 HealthLake 데이터 스토어와의 FHIR REST API 상호 작용 사용CRUD하세요.

이 장의 주제에서는 HealthLake 데이터 스토어를 Athena에 연결하는 방법,를 사용하여 쿼리하는 방법 SQL, 추가 분석을 위해 결과를 다른 AWS 서비스와 연결하는 방법을 설명합니다.

목차

Amazon Athena에 데이터 스토어 연결

> **Important**
>
> 2022년 11월 14일 이후 액세스 요구 IAM 사항이 HealthLake 변경되었습니다. 데이터 스토어를 생성하고 Athena에서 데이터 스토어에 대한 액세스 권한을 부여하려면 `AWSLakeFormationDataAdmin` 관리형 정책을 IAM 사용자, 그룹 또는 역할에 추가해야 합니다. `AWSLakeFormationDataAdmin` 정책을 사용하여 데이터 레이크 관리자를 생성하고 Athena의 데이터 스토어에 대한 액세스 권한을 부여할 수 있습니다.

이 주제에서는 Athena 사용자, 그룹 또는 역할을 생성하고 HealthLake 데이터 스토어에 있는 FHIR 리소스에 대한 액세스 권한을 부여하는 데 필요한 단계를 간략하게 설명합니다.

- HealthLake 데이터 스토어에 대한 사용자, 그룹 또는 역할 액세스 권한 부여(AWS Lake Formation 콘솔)
- Athena 계정 설정

사용자, 그룹 또는 역할에 HealthLake 데이터 스토어에 대한 액세스 권한 부여(AWS Lake Formation 콘솔)

> 페르소나: HealthLake 관리자
>
> HealthLake 관리자 페르소나는 AWS Lake Formation의 데이터 레이크 관리자입니다. Lake Formation의 HealthLake 데이터 스토어에 대한 액세스 권한을 부여합니다.

생성된 각 데이터 스토어에 대해 AWS Lake Formation 콘솔에 두 개의 항목이 표시됩니다. 한 가지 항목은 리소스 링크입니다. 리소스 링크 이름은 항상 기울임꼴로 표시됩니다. 각 리소스 링크는 연결된 공유 리소스의 이름 및 소유자와 함께 표시됩니다. 모든 HealthLake 데이터 스토어에서 공유 리소스 소유자는 HealthLake 서비스 계정입니다. 다른 항목은 HealthLake 서비스 계정의 HealthLake 데이터 스토어입니다. 이 절차의 단계는 리소스 링크인 데이터 스토어를 사용합니다.

리소스 링크에 대한 자세한 내용은 Lake Formation 개발자 안내서의 Lake Formation에서 리소스 링크가 작동하는 방식을 참조하세요. AWS

사용자, 그룹 또는 역할이 Athena에서 데이터를 쿼리할 수 있으려면 리소스 데이터베이스에 대한 설명 권한을 부여해야 합니다. 그런 다음 테이블에 Select and Describe를 부여해야 합니다.

STEP 1: HealthLake 데이터 스토어 리소스 링크 데이터베이스에 대한 DESCRIBE 권한을 부여하려면

1. AWS Lake Formation 콘솔을 엽니다. https://console.aws.amazon.com/lakeformation/
2. 기본 탐색 모음에서 데이터베이스를 선택합니다.
3. 데이터베이스 페이지에서 기울임꼴로 표시된 데이터 스토어 이름 옆에 있는 라디오 버튼을 선택합니다.
4. 작업(▾)을 선택합니다.
5. 권한 부여를 선택합니다.
6. 데이터 권한 부여 페이지의 보안 주체에서 IAM 사용자 또는 역할을 선택합니다.
7. IAM 사용자 또는 역할에서 아래쪽 화살표(▾)를 사용하거나 Athena에서 쿼리를 수행할 수 있는 IAM 사용자, 역할 또는 그룹을 검색합니다.
8. LF 태그 또는 카탈로그 리소스 카드에서 명명된 데이터 카탈로그 리소스 옵션을 선택합니다.
9. 데이터베이스에서 아래쪽 화살표(▾)를 사용하여 액세스를 공유할 HealthLake 데이터 스토어 데이터베이스를 선택합니다.
10. 리소스 링크 권한 카드의 리소스 링크 권한에서 설명을 선택합니다.

권한 부여가 성공하면 권한 부여 성공 배너가 나타납니다. 방금 부여한 권한을 보려면 데이터 레이크 권한을 선택합니다. 테이블에서 사용자, 그룹 및 역할을 찾습니다. 권한 열 아래에 나열된 설명이 표시됩니다.

이제 대상에 대한 권한 부여를 사용하여 데이터베이스의 모든 테이블에 Select 및 Describe를 부여해야 합니다.

STEP 2: HealthLake 데이터 스토어 리소스 링크의 모든 테이블에 대한 액세스 권한 부여

1. AWS Lake Formation 콘솔을 엽니다. https://console.aws.amazon.com/lakeformation/
2. 기본 탐색 모음에서 데이터베이스를 선택합니다.
3. 데이터베이스 페이지에서 기울임꼴로 표시된 데이터 스토어 이름 옆에 있는 라디오 버튼을 선택합니다.
4. 작업(▾)을 선택합니다.
5. 대상에 부여를 선택합니다.
6. 데이터 권한 부여 페이지의 보안 주체에서 IAM 사용자 또는 역할을 선택합니다.
7. IAM 사용자 또는 역할에서 아래쪽 화살표(▾)를 사용하거나 Athena에서 쿼리를 수행할 수 있는 IAM 사용자, 그룹 또는 역할을 검색합니다.
8. LF 태그 또는 카탈로그 리소스 카드에서 명명된 데이터 카탈로그 리소스 옵션을 선택합니다.
9. 데이터베이스에서 아래쪽 화살표(▾)를 사용하여 액세스 권한을 부여할 HealthLake 데이터 스토어 데이터베이스를 선택합니다.
10. 테이블에서 모든 테이블을 선택하여 모든 테이블을 HealthLake 사용자와 공유합니다.
11. 테이블 권한 카드의 테이블 권한에서 설명 및 선택을 선택합니다.
12. 권한 부여를 선택합니다.

권한 부여를 선택하면 권한 부여 성공 배너가 나타납니다. 이제 지정된 사용자가 Athena의 HealthLake 데이터 스토어에서 쿼리를 수행할 수 있습니다.

Athena 시작하기

> **HealthLake 사용자**
>
> HealthLake 사용자는 Athena 콘솔 AWS CLI또는 AWS SDKs를 사용하여 HealthLake 관리자가 공유한 HealthLake 데이터 스토어를 쿼리합니다.

Athena를 사용하여 데이터 스토어를 쿼리하려면 다음 세 가지 작업을 수행해야 합니다.

- Lake Formation을 통해 HealthLake 데이터 스토어에 대한 IAM 사용자 또는 역할 액세스 권한을 부여합니다. 자세한 내용은 사용자, 그룹 또는 역할에 HealthLake 데이터 스토어에 대한 액세스 권한 부여(AWS Lake Formation 콘솔)을 참조하십시오.
- HealthLake 데이터 스토어에 대한 작업 그룹을 생성합니다.

- 쿼리 결과를 저장할 Amazon S3 버킷을 지정합니다.

Athena를 시작하려면 사용자, 그룹 또는 역할에 AmazonAthenaFullAccess 및 AmazonS3FullAccess AWS 관리형 정책을 추가합니다. AWS 관리형 정책을 사용하면 새 서비스를 사용할 수 있습니다. AWS 관리형 정책은 모든 AWS 고객이 사용할 수 있기 때문에 특정 사용 사례에 대해 최소 권한을 부여하지 않을 수 있습니다. IAM 정책으로 권한을 설정할 때 작업을 수행하는 데 필요한 권한만 부여합니다. 최소 권한 적용IAM에 대한 자세한 내용은 IAM 사용 설명서의 최소 권한 적용을 참조하세요.

> **Important**
>
> Athena에서 HealthLake 데이터 스토어를 쿼리하려면 Athena 엔진 버전 3을 사용해야 합니다.

작업 그룹은 리소스이므로 IAM기반 정책을 사용하여 특정 작업 그룹에 대한 액세스를 제어할 수 있습니다. 자세한 내용은 Athena 사용 설명서의 작업 그룹을 사용하여 쿼리 액세스 및 비용 제어를 참조하세요.

작업 그룹 설정에 대한 자세한 내용은 Athena 사용 설명서https://docs.aws.amazon.com/athena/latest/ug/workgroups-procedure.html의 섹션을 참조하세요.

> **Note**
>
> Amazon S3 버킷이 있는 리전이며 Athena 콘솔이 일치해야 합니다.

쿼리를 실행하려면 먼저, Amazon S3에서 쿼리 결과 버킷 위치를 지정하거나, 버킷을 지정했고 구성이 클라이언트 설정을 재정의하는 작업 그룹을 사용해야 합니다. 실행되는 모든 쿼리에 대해 출력 파일은 자동으로 저장됩니다.

Athena 콘솔에서 쿼리 결과 위치를 지정하는 방법에 대한 자세한 내용은 Amazon Athena 사용 설명서의 Athena 콘솔을 사용하여 쿼리 결과 위치 지정을 참조하세요.

Athena에서 HealthLake 데이터 스토어를 쿼리하는 방법의 예를 보려면 섹션을 참조하세요를 사용하여 HealthLake 데이터 스토어 쿼리 SQL.

를 사용하여 HealthLake 데이터 스토어 쿼리 SQL

> **Note**
>
> 2023년 2월 20일 이후 HealthLake 데이터 스토어는 기본적으로 통합 자연어 처리(NLP)를 사용하지 않습니다. 데이터 스토어에서이 기능을 활성화하려면 문제 해결 HealthLake의 통합 자연어 처리 기능을 켜려면 어떻게 해야 합니까? 장의 섹션을 참조하세요.
> 이 주제의 모든 예제에서는 Synthea를 사용하여 생성된 가상 데이터를 사용합니다. Synthea 데이터로 미리 로드된 데이터 스토어를 생성하는 방법에 대한 자세한 내용은 섹션을 참조하세요에서 데이터 스토어 생성 AWS HealthLake.

HealthLake 데이터 스토어를 Athena로 가져오면 HealthLake 데이터 스토어의 각 리소스 유형이 테이블로 변환됩니다. 이러한 테이블은 개별적으로 쿼리하거나 SQL기반 쿼리를 사용하여 그룹으로 쿼리할 수 있습니다. 데이터 스토어의 구조로 인해 데이터는 여러 데이터 유형으로 Athena로 가져옵니다. 이러한 데이터 유형에 액세스할 수 있는 SQL 쿼리를 생성하는 방법에 대한 자세한 내용은 Amazon Athena 사용 설명서의 복잡한 유형 및 중첩 구조로 배열 쿼리를 참조하세요.

리소스 유형의 각 요소에 대해 FHIR 사양은 카디널리티를 정의합니다. 요소의 카디널리티는이 요소가 나타날 수 있는 횟수의 하한과 상한을 정의합니다. SQL 쿼리를 생성할 때 이를 고려해야 합니다. 예를 들어 리소스 유형: 환자에서 몇 가지 요소를 살펴보겠습니다.

- 요소: 이름 FHIR 사양은 카디널리티를 로 설정합니다`0..*`.

  요소는 배열로 캡처됩니다.

```
[{
 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
}]
```

Athena에서 리소스 유형이 어떻게 수집되었는지 확인하려면 테이블 및 보기에서 해당 유형을 검색합니다. 이 배열의 요소에 액세스하려면 점 표기법을 사용할 수 있습니다. 다음은 given 및의 값에 액세스하는 간단한 예제입니다family.

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

- 요소: MaritalStatus FHIR 사양은 카디널리티를 로 설정합니다0..1.

  이 요소는 로 캡처됩니다JSON.

```
{
 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
}
```

Athena에서 리소스 유형이 어떻게 수집되었는지 확인하려면 테이블 및 뷰에서 해당 유형을 검색합니다. 에서 키-값 페어에 액세스하려면 점 표기법을 사용하면 JSON됩니다. 배열이 아니므로 배열 인덱스가 필요하지 않습니다. 다음은에 대한 값에 액세스하는 간단한 예입니다`text`.

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

액세스 및 검색에 대한 자세한 내용은 Athena 사용 설명서의 쿼리JSON를 JSON참조하세요.

Athena Data Manipulation Language(DML) 쿼리 문은 Trino를 기반으로 합니다. Athena는 Trino의 모든 기능을 지원하지 않으며 상당한 차이가 있습니다. 자세한 내용은 Amazon Athena 사용 설명서의 DML 쿼리, 함수 및 연산자를 참조하세요.

또한 Athena는 데이터 스토어의 쿼리를 생성할 때 발생할 수 있는 여러 HealthLake 데이터 유형을 지원합니다. Athena의 데이터 유형에 대한 자세한 내용은 Amazon Athena 사용 설명서의 Amazon Athena의 데이터 유형을 참조하세요. Amazon Athena

Athena에서 SQL 쿼리가 작동하는 방식에 대한 자세한 내용은 SQL Amazon Athena 사용 설명서의 Amazon Athena 참조를 참조하세요.

각 탭에는 Athena를 사용하여 지정된 리소스 유형 및 관련 요소를 검색하는 방법의 예가 나와 있습니다.

Element: Extension

요소는 데이터 스토어에서 사용자 지정 필드를 생성하는 데 `extension` 사용됩니다.

이 예제에서는 `Patient` 리소스 유형에 있는 `extension` 요소의 기능에 액세스하는 방법을 보여줍니다.

HealthLake 데이터 스토어를 Athena로 가져오면 리소스 유형의 요소가 다르게 구문 분석됩니다. 의 구조`element`는 가변적이므로 스키마에 완전히 지정할 수 없습니다. 이러한 변동성을 처리하기 위해 배열 내의 요소가 문자열로 전달됩니다.

의 표 설명에서 로 `extension` 설명된 요소를 볼 `Patient`수 있습니다. 즉`array<string>`, 인덱스 값을 사용하여 배열 요소에 액세스할 수 있습니다. 그러나 문자열의 요소에 액세스하려면를 사용해야 합니다`json_extract`.

다음은 환자 테이블에 있는 `extension` 요소의 단일 항목입니다.

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

유효한 이지만 JSONAthena는 이를 문자열로 취급합니다.

이 SQL 쿼리 예제에서는 `patient-mothersMaidenName` 및 `patient-birthPlace` 요소가 포함된 테이블을 생성하는 방법을 보여줍니다. 이러한 요소에 액세스하려면 서로 다른 배열 인덱스 및를 사용해야 합니다. `json_extract`.

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

와 관련된 쿼리에 대한 자세한 내용은 Amazon Athena 사용 설명서의 에서 데이터 추출JSON을 JSON참조하세요.

Element: birthDate (Age)

연령은에서 환자 리소스 유형의 요소가 아닙니다FHIR. 다음은 연령을 기준으로 필터링하는 검색의 두 가지 예입니다.

수명은 요소가 아니므로 SQL 쿼리birthDate에를 사용합니다. 요소가 어떻게에 수집되었는지 확인하려면 테이블 및 뷰에서 테이블 이름을 FHIR검색합니다. 문자열 형식임을 알 수 있습니다.

예제 1: 연령 값 계산

이 샘플 SQL 쿼리에서는 기본 제공 SQL 도구 current_date 및를 사용하여 이러한 구성 요소를 추출year합니다. 그런 다음 빼서 환자의 실제 연령을 라는 열로 반환합니다age.

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

예제 2: 이전에 태어났2019-01-01고 인 환자를 필터링합니다male.

SQL 쿼리는 CAST 함수를 사용하여 birthDate 요소를 유형으로 캐스팅DATE하는 방법과 WHERE 절의 두 가지 기준을 기반으로 필터링하는 방법을 보여줍니다. 요소는 기본적으로 유형 문자열로 수집되므로 유형 문자열을 CAST로 수집해야 합니다DATE. 그런 다음 < 연산자를 사용하여 다른 날짜인와 비교할 수 있습니다2019-01-01. 를 사용하여 WHERE 절에 두 번째 기준을 추가할 AND수 있습니다.

```
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

이 예제에서는 도시 이름이 Attleboro인 Location 리소스 유형 내의 위치를 검색합니다.

```
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

리소스 유형 조건은 우려 수준으로 상승한 문제와 관련된 진단 데이터를 저장합니다. HealthLake의 통합 의료 자연어 처리(NLP)는 Condition 리소스 유형에 있는 세부 정보를 기반으로 새 DocumentReference 리소스를 생성합니다. 새 리소스가 생성되면가 SYSTEM_GENERATED meta 요소에 태그를 HealthLake 추가합니다. 이 샘플 SQL 쿼리는 조건 테이블을 검색하고 결과가 제거된 SYSTEM_GENERATED 결과를 반환하는 방법을 보여줍니다.

HealthLake의 통합 자연어 처리(NLP)에 대한 자세한 내용은 섹션을 참조하세요에서 리소스 유형의 자연어 처리(NLP)를 기반으로 자동화된 FHIR DocumentReference 리소스 생성 사용 AWS HealthLake.

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

지정된 문자열 요소 내에서 검색하여 쿼리를 추가로 필터링할 수도 있습니다. modifierextension 요소에는 조건 세트를 생성하는 데 사용된 DocumentReference 리소스에 대한 세부 정보가 포함되어 있습니다. 다시 말하지만 json_extract를 사용하여 Athena로 문자열로 가져온 중첩 JSON 요소에 액세스해야 합니다.

이 샘플 SQL 쿼리는 특정를 기반으로 생성된 모든를 검색Condition하는 방법을 보여줍니다DocumentReference. 를 사용하여 JSON 요소를 문자열로 CAST 설정하여를 사용하여 비교할 수 LIKE 있습니다.

```
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

리소스 유형인 관찰은 환자, 디바이스 또는 기타 주제에 대한 측정치와 간단한 어설션을 저장합니다. HealthLake의 통합 자연어 처리(NLP)는 Observation 리소스에서 발견된 세부 정보를 기반으로 새 DocumentReference 리소스를 생성합니다. 이 샘플 SQL 쿼리에는 WHERE meta.tag[1] is NULL 주석이 포함되어 있으므로 SYSTEM_GENERATED 결과가 포함됩니다.

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

이 열은 로 가져왔습니다`struct`. 따라서 점 표기법을 사용하여 내부 요소에 액세스할 수 있습니다.

Resource type: MedicationStatement

MedicationStatement 는 환자가 복용했거나 현재 복용 중이거나 향후 복용하게 될 약물에 대한 세부 정보를 저장하는 데 사용할 수 있는 FHIR 리소스 유형입니다. HealthLake의 통합 의료 자연어 처리(NLP)는 MedicationStatement 리소스 유형에서 발견된 문서를 기반으로 새 DocumentReference 리소스를 생성합니다. 새 리소스가 생성되면가 `SYSTEM_GENERATED` `meta` 요소에 태그를 HealthLake 추가합니다. 이 샘플 SQL 쿼리는 식별자를 사용하여 단일 환자를 기준으로 필터링하고에서 통합에 의해 추가된 리소스를 찾는 쿼리 HealthLake를 생성하는 방법을 보여줍니다NLP.

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

HealthLake의 통합 의료에 대한 자세한 내용은 섹션을 NLP참조하세요에서 리소스 유형의 자연어 처리(NLP)를 기반으로 자동화된 FHIR DocumentReference 리소스 생성 사용 AWS HealthLake.

복잡한 필터링을 사용한 SQL 쿼리 예제

> **Note**
>
> 2023년 2월 20일 이후 HealthLake 데이터 스토어는 기본적으로 통합 자연어 처리(NLP)를 사용하지 않습니다. 데이터 스토어에서이 기능을 활성화하려면 문제 해결 HealthLake의 통합 자연어 처리 기능을 켜려면 어떻게 해야 합니까? 장의 섹션을 참조하세요.

이 주제의 예제에는 복잡한 필터링 HealthLake을 사용하는 Athena와의 통합에 대한 SQL 쿼리가 포함됩니다.

Example 인구 통계 데이터를 기반으로 필터링 기준 생성

환자 코호트를 생성할 때 올바른 환자 인구통계를 식별하는 것이 중요합니다. 이 샘플 쿼리는 Trino 점 표기법 및를 사용하여 HealthLake 데이터 스토어의 데이터를 필터링`json_extract`하는 방법을 보여 줍니다.

```
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;
```

Athena 콘솔을 사용하면 결과를 추가로 정렬하고 다운로드할 수 있습니다.

Example 환자 및 관련 조건에 대한 필터 생성

이 샘플 쿼리는 HealthLake 데이터 스토어에서 찾은 환자의 모든 관련 조건을 찾고 정렬하는 방법을 보여줍니다.

```
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;
```

Athena 콘솔을 사용하여 이러한 결과를 추가로 정렬하거나 추가 분석을 위해 다운로드할 수 있습니다.

Example 환자 및 관련 관찰에 대한 필터 생성

이 샘플 쿼리는 HealthLake 데이터 스토어에서 찾은 환자의 모든 관련 관찰을 찾고 정렬하는 방법을 보여줍니다.

```
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 환자 및 관련 절차에 대한 필터링 조건 생성

환자와의 연결 절차는 의료의 중요한 측면입니다. 이 SQL 쿼리는 환자 및 프로시저 리소스 유형을 사용하여 Athena에서 이를 수행하는 방법을 보여줍니다. 이 SQL 쿼리는 HealthLake 데이터 스토어에 있는 모든 환자와 관련 절차를 반환합니다.

```
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;
```

이제 Athena 콘솔을 사용하여 추가 분석을 위해 결과를 다운로드하거나 결과를 더 잘 이해할 수 있도록 정렬할 수 있습니다.

Example 환자 및 관련 처방에 대한 필터링 조건 생성

환자가 복용 중인 약물의 현재 목록을 보는 것이 중요합니다. Athena를 사용하여 HealthLake 데이터 스토어에 있는 환자 및 MedicationRequest 리소스 유형을 모두 사용하는 SQL 쿼리를 작성할 수 있습니다.

이 SQL 쿼리는 Athena로 가져온 환자 및 MedicationRequest 테이블을 조인합니다. 또한 점 표기법을 사용하여 처방을 개별 항목으로 구성합니다.

```
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
```

Athena 콘솔을 사용하여 결과를 정렬하거나 추가 분석을 위해 다운로드할 수 있습니다.

Example MedicationStatement 리소스 유형에서 발견된 약물 보기

예제 쿼리는를 사용하여 Athena로 JSON 가져온 중첩된를 구성하는 방법을 보여줍니다SQL. 쿼리는 `meta` 요소를 사용하여 HealthLake의 통합 자연어 처리()에 의해 약물이 추가된 시기를 나타냅니다 NLP. HealthLake의 Amazon Comprehend Medical과의 통합에 대한 자세한 내용은 섹션을 참조하세요[에서 리소스 유형의 자연어 처리(NLP)를 기반으로 자동화된 FHIR DocumentReference 리소스 생성 사용 AWS HealthLake](). 또한 `json_extract`를 사용하여 JSON 문자열 배열 내의 데이터를 검색합니다.

```
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;
```

Athena 콘솔을 사용하여 이러한 결과를 다운로드하거나 정렬할 수 있습니다.

Example 특정 질병 유형 필터링

이 예제는 당뇨병 진단을 받은 18~75세의 환자 그룹을 찾는 방법을 보여줍니다.

```
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%');
```

이제 Athena 콘솔을 사용하여 결과를 정렬하거나 추가 분석을 위해 다운로드할 수 있습니다.

AWS HealthLake 및 인터페이스 VPC 엔드포인트(AWS PrivateLink)

인터페이스 VPC 엔드포인트를 생성 AWS HealthLake 하여 VPC와 간에 프라이빗 연결을 설정할 수 있습니다. 인터페이스 VPC 엔드포인트는 인터넷 게이트웨이AWS PrivateLink, NAT 디바이스, VPN 연결 또는 AWS Direct Connect 연결 HealthLake APIs 없이 비공개로 액세스하는 데 사용할 수 있는 기술인 로 구동됩니다. 의 인스턴스는 HealthLake와 통신하는 데 퍼블릭 IP 주소가 필요하지 VPC 않습니다 APIs. VPC와 HealthLake; 간의 트래픽은 Amazon 네트워크를 벗어나지 않습니다.

각 인터페이스 엔드포인트는 서브넷에서 하나 이상의 탄력적 네트워크 인터페이스로 표현됩니다.

자세한 내용은 Amazon VPC 사용 설명서의 인터페이스 VPC 엔드포인트(AWS PrivateLink)를 참조하세요.

엔드포인트 고려 HealthLake VPC 사항

에 대한 인터페이스 VPC 엔드포인트를 설정하기 전에 Amazon VPC 사용 설명서의 인터페이스 엔드포인트 속성 및 제한 사항을 검토해야 HealthLake합니다.

HealthLake 는에서 모든 API 작업을 호출할 수 있도록 지원합니다VPC.

에 대한 인터페이스 VPC 엔드포인트 생성 HealthLake

Amazon VPC 콘솔 또는 AWS Command Line Interface ()를 사용하여 HealthLake; 서비스에 대한 VPC 엔드포인트를 생성할 수 있습니다AWS CLI. 자세한 내용은 Amazon VPC 사용 설명서의 인터페이스 엔드포인트 생성을 참조하세요.

다음 서비스 이름을 사용하여 HealthLake에 대한 VPC 엔드포인트를 생성합니다.

- com.amazonaws.*region*.healthlake

엔드포인트에 DNS 대해 프라이빗을 켜면 리전의 기본 DNS 이름을 사용하여에 API HealthLake 요청할 수 있습니다. 예: *healthlake.us-east-1.amazonaws.com*.

자세한 내용은 Amazon VPC 사용 설명서의 인터페이스 엔드포인트를 통한 서비스 액세스를 참조하세요.

에 대한 VPC 엔드포인트 정책 생성 HealthLake

에 대한 액세스를 제어하는 엔드포인트에 VPC 엔드포인트 정책을 연결할 수 있습니다 HealthLake. 이 정책은 다음 정보를 지정합니다.

- 작업을 수행할 수 있는 위탁자.
- 수행할 수 있는 작업.
- 작업을 수행할 수 있는 리소스.

자세한 내용은 Amazon VPC 사용 설명서의 VPC 엔드포인트를 사용하여 서비스에 대한 액세스 제어를 참조하세요.

예: HealthLake 작업에 대한 VPC 엔드포인트 정책

다음은에 대한 엔드포인트 정책의 예입니다 HealthLake. 엔드포인트에 연결되면이 정책은 모든 리소스의 HealthLake 모든 보안 주체에 대해 `CreateFHIRDatastore` 작업에 대한 액세스 권한을 부여합니다.

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

에서 리소스 태그 지정 AWS HealthLake

AWS 리소스에 태그 형태로 메타데이터를 지정할 수 있습니다. 각 태그는 사용자 정의 키와 값으로 구성된 레이블입니다. 태그를 사용하면 리소스를 손쉽게 관리, 식별, 정리, 검색 및 필터링할 수 있습니다.

이 주제에서는 일관되고 효과적인 태깅 전략을 구현하는 데 도움이 되는 일반적인 태깅 범주 및 전략에 대해 설명합니다. 다음 섹션에서는 AWS 리소스, 태그 지정, 세부 결제, AWS 자격 증명 및 액세스 관리 ()에 대한 기본 지식을 배웁니다IAM.

각 태그에는 다음 두 가지 부분이 있습니다.

- 태그 키(예: CostCenter환경 또는 프로젝트). 태그 키는 대소문자를 구별합니다.
- 태그 값(예: 111122223333 또는 Production). 태그 키처럼 태그 값은 대/소문자를 구별합니다.

태그를 사용하여 용도, 소유자, 환경 또는 기타 기준으로 리소스를 분류할 수 있습니다. 자세한 내용은 AWS 태그 지정 전략을 참조하세요.

각 리소스의 서비스 콘솔, 서비스 또는에서 한 번에 하나의 리소스에 태그를 추가API, 변경 또는 제거할 수 있습니다AWSCLI.

태그 지정을 활성화하려면 TagResources 가 승인되었는지 확인합니다. 다음 예제와 같은 IAM 정책을 TagResources 연결하여 권한을 부여할 수 있습니다.

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

중요 공지 사항

AWS HealthLake 는 AWS 공동 책임 모델 정책에 따라 고객 데이터를 보호합니다. 즉, 모든 고객 데이터는 전환 및 유휴 상태에서 모두 암호화됩니다. 그러나 데이터 스토어 또는 작업 기반 작업에 대한 모든 고객 입력 이름이 암호화되지는 않습니다. 개인 식별 정보 또는 보호 대상 건강 정보를 포함해서는 안 됩니다. 자세한 내용은 AWS HealthLake 보안 장을 참조하세요.

모범 사례

AWS 리소스에 대한 태깅 전략을 만들 때는 다음 모범 사례를 따르십시오.

- 개인 식별 정보(PII), 개인 건강 정보(PHI) 또는 기타 민감한 정보를 태그에 저장하지 마십시오.
- 대/소문자를 구분하는 표준화된 태그 형식을 사용하고 모든 리소스 유형에 일관되게 적용합니다.
- 리소스 액세스 제어, 비용 추적, 자동화 및 조직 관리와 같은 다양한 용도를 지원하는 태그 지침을 고려합니다.
- 자동화된 도구를 사용하여 리소스 태그를 관리할 수 있습니다. AWS Resource Groups 및 Resource Groups TaggingAPI을 사용하면 태그를 프로그래밍 방식으로 제어할 수 있으므로 태그와 리소스를 자동으로 관리, 검색 및 필터링할 수 있습니다.
- 태그를 더 많이 사용하면 태그 지정이 더 효과적입니다.
- 태그는 사용자가 변경해야 할 때 편집하거나 수정할 수 있지만 액세스 제어 태그를 업데이트하려면 해당 태그를 참조하는 정책도 업데이트하여 리소스에 대한 액세스를 제어해야 합니다.

태그 지정 요구 사항

태그를 지정할 때 요건은 다음과 같습니다.

- 키에는 aws 접두사를 붙일 수 없습니다.
- 키는 태그 집합에 대해 고유해야 합니다.
- 키는 1~128자 사이의 허용된 문자이어야 합니다.
- 값은 0~256자 사이의 허용된 문자이어야 합니다.
- 값은 태그 집합마다 고유할 필요는 없습니다.
- 키와 값의 문자로는 Unicode 문자, 숫자, 공백 그리고 다음 기호가 허용됩니다. _ . : / = + - @.
- 키와 값은 대/소문자를 구분합니다.

데이터 스토어에 태그 추가

데이터 스토어에 태그를 추가하면 AWS 리소스를 식별 및 구성하고 리소스에 대한 액세스를 관리하는 데 도움이 될 수 있습니다. 먼저 데이터 스토어에 하나 이상의 태그(키-값 페어)를 추가합니다. 사용자당 최대 50개의 태그를 사용할 수 있습니다. 키 및 값 필드에 사용할 수 있는 문자에 대한 제한도 있습니다.

태그가 있으면 이러한 태그를 기반으로 데이터 스토어에 대한 액세스를 관리하는 IAM 정책을 생성할 수 있습니다. HealthLake 콘솔 또는를 사용하여 데이터 스토어에 태그를 AWS CLI 추가할 수 있습니다. 리포지토리에 태그를 추가하면 해당 리포지토리에 대한 액세스에 영향을 줄 수 있습니다. 데이터 스토어에 태그를 추가하기 전에 태그를 사용하여 데이터 스토어와 같은 리소스에 대한 액세스를 제어할 수 있는 IAM 정책을 검토해야 합니다.

다음 단계에 따라를 사용하여 HealthLake 데이터 스토어 AWS CLI 에 태그를 추가합니다. 데이터 스토어를 생성할 때 태그를 추가하려면 섹션을 참조하세요에서 데이터 스토어 생성 AWS HealthLake.

터미널 또는 명령줄에서 태그를 추가할 데이터 스토어의 Amazon 리소스 이름(ARN)과 추가하려는 태그의 키 및 값을 지정하여 tag-resource 명령을 실행합니다. 데이터 스토어에 태그를 두 개 이상 추가할 수 있습니다. 키 및 값 필드에 사용할 수 있는 문자에 대한 제한도 있습니다태그 지정 요구 사항. 예를 들어, 데이터 스토어가 생성되는 동안 데이터 스토어에 태그를 추가하려면에서 다음 명령을 사용합니다 AWS CLI. 데이터 스토어의 이름은 Test_Data_Store이고, 키가 있는 두 개의 추가된 태그는 key1 및 key2이며 값은 각각 value1 및 value2입니다.
:

```
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
```

기존 데이터 스토어에 태그를 추가하려면 다음 예제 명령을 실행합니다.

```
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
```

성공하면이 명령은 응답을 반환하지 않습니다.

데이터 스토어에 대한 태그 나열

다음 단계에 따라를 사용하여 HealthLake 데이터 스토어의 AWS 태그 목록을 AWS CLI 봅니다. 태그가 추가되지 않은 경우 반환되는 목록은 비어 있습니다.

터미널 또는 명령줄에서 다음 예제와 list-tags-for-resource 같이 명령을 실행합니다.

```
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"
    }
}
```

데이터 스토어에서 태그 제거

데이터 스토어와 연결된 태그를 하나 이상 제거할 수 있습니다. 태그를 제거할 때 해당 태그와 연결된 다른 AWS 리소스에서 해당 태그가 삭제되지는 않습니다.

터미널 또는 명령줄에서 untag-resource 명령을 실행하여 태그를 제거할 데이터 스토어의 Amazon 리소스 이름(ARN)과 제거할 태그의 태그 키를 지정합니다.

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

성공하면이 명령은 응답을 반환하지 않습니다. 데이터 스토어와 연결된 태그를 확인하려면 명령을 실행합니다 list-tags-for-resource.

모니터링 HealthLake

모니터링은 HealthLake 및 다른 AWS 솔루션의 안정성, 가용성 및 성능을 유지하는 데 중요한 부분입니다.는 다음과 같은 모니터링 도구를 AWS 제공하여 문제가 있을 때 이를 모니터링하고 HealthLake보고하며 적절한 경우 자동 조치를 취합니다.

- Amazon CloudWatch은 AWS 리소스와 실행 중인 애플리케이션을 AWS 실시간으로 모니터링합니다. 지표를 수집 및 추적하고, 사용자 지정 대시보드를 생성하고, 지정된 지표가 특정 임계값에 도달하면 알림을 보내거나 조치를 취하는 경보를 설정할 수 있습니다. 예를 들어 Amazon EC2 인스턴스의 CPU 사용량 또는 기타 지표를 CloudWatch 추적하고 필요한 경우 새 인스턴스를 자동으로 시작할 수 있습니다. 자세한 내용은 Amazon CloudWatch 사용 설명서를 참조하세요.
- AWS CloudTrail는 계정에서 또는 계정을 대신하여 수행한 API 호출 및 관련 이벤트를 캡처합니다 AWS . 그리고 나서 사용자가 지정한 Amazon S3 버킷에 로그 파일을 전송합니다. 호출된 사용자 및 계정 AWS, 해당 호출의 소스 IP 주소, 호출 발생 시점을 식별할 수 있습니다. 자세한 내용은 AWS CloudTrail 사용 설명서를 참조하십시오.

주제

- Amazon HealthLake 을 사용한 모니터링 CloudWatch

Amazon HealthLake 을 사용한 모니터링 CloudWatch

원시 데이터를 수집 CloudWatch하여 읽기 가능하고 실시간에 가까운 지표로 처리하는를 HealthLake 사용하여 모니터링할 수 있습니다. 이러한 통계는 15개월간 보관되므로 기록 정보를 사용하여 웹 애플리케이션 또는 서비스가 어떻게 실행되고 있는지 전체적으로 더 잘 파악할 수 있습니다. 특정 임계값을 주시하다가 해당 임계값이 충족될 때 알림을 전송하거나 조치를 취하도록 경보를 설정할 수도 있습니다. 자세한 내용은 Amazon CloudWatch 사용 설명서를 참조하세요.

지표는 다음을 HealthLake APIs포함하여 모든에 대해 보고됩니다.

- 데이터 스토어 관리 APIs - CreateFHIRDatastore, DeleteFHIRDatastore, DescribeFHIRDatastore, ListFHIRDatastores
- 가져오기 및 내보내기 APIs - S tartFHIRImport작업, L istFHIRImport작업, D escribeFHIRImport작업, S tartFHIRExport작업, L istFHIRExport작업, D escribeFHIRExport작업
- HTTP REST 클라이언트 및 리소스 관리 APIs — CreateResource, DeleteResource, GetCapabilities, ReadResource SearchAll, SearchWithGet, SearchWithPost, UpdateResource.

- 태그 지정 APIs — ListTagsForResource, TagResource, UntagResource

다음 표는 HealthLake에서 사용 가능한 지표와 차원을 열거한 것입니다.

다음 지표가 보고됩니다. 각는 사용자 지정 데이터 범위의 빈도 수로 표시됩니다.

지표

지표	설명
호출 개수	에 대한 호출 수입니다APIs. 이는 계정 또는 지정된 데이터 스토어에 대해 보고될 수 있습니다. 단위: 개 유효한 통계: Sum, Count 측정기준: 작업, 데이터 스토어 ID, 데이터 스토어 유형
성공한 요청	성공한 API 요청 수입니다. 단위: 개 유효한 통계: Sum, Average 차원: 작업, 데이터 스토어, 데이터 스토어 유형
사용자 오류	사용자 오류로 인해 실패한 요청 수입니다. 단위: 개 유효한 통계: Sum, Average 측정기준: 작업, 데이터 스토어 ID, 데이터 스토어 유형
서버 오류	서버 오류로 인해 실패한 요청 수입니다. 단위: 개 유효한 통계: Sum, Average

지표	설명
	측정기준: 작업, 데이터 스토어 ID, 데이터 스토어 유형
조정된(throttle) 요청	제한된 요청 수입니다. 이 지표는 사용자 또는 서버 오류 수에 포함되지 않습니다. 단위: 개 유효한 통계: Sum, Average 측정기준: 작업, 데이터 스토어 ID, 데이터 스토어 유형
지연 시간	사용자 요청을 처리하는 데 밀리초 단위의 시간이 걸렸습니다. 단위: 밀리초 유효 통계: Minimum, Maximum, Average, Sum 측정기준: 작업, 데이터 스토어 ID, 데이터 스토어 유형

다음 차원이 보고됩니다.

차원

Dimensions	설명
Operation	사용된 API 작업
DataStoreID	API 요청에 포함된 데이터 스토어
DataStoreType	데이터 스토어 유형(현재 FHIR R4만 지원됨)

AWS Management Console, AWS CLI또는를 HealthLake 사용하여에 대한 지표를 가져올 수 있습니다 CloudWatch API. Amazon AWS 소프트웨어 개발 키트(SDKs) 또는 도구 중 하나를 통해를

CloudWatch API 사용할 CloudWatch API 수 있습니다. HealthLake 콘솔은의 원시 데이터를 기반으로 그래프를 표시합니다 CloudWatch API.

에서 모니터링할 수 HealthLake 있는 적절한 CloudWatch 권한이 있어야 합니다 CloudWatch. 자세한 내용은 Amazon 사용 설명서의 Amazon 인증 및 액세스 제어를 CloudWatch 참조하세요. CloudWatch

HealthLake 지표 보기

지표를 보려면(CloudWatch 콘솔)

1. AWS Management Console에 로그인하여 CloudWatch 콘솔을 엽니다.
2. 지표를 선택하고 모든 지표를 선택한 다음 AWS/HealthLake를 선택합니다.
3. 차원과 지표 이름을 선택한 다음 그래프에 추가를 선택합니다.
4. 날짜 범위 값을 선택합니다. 선택한 날짜 범위에 대한 지표 개수가 그래프에 표시됩니다.

를 사용하여 경보 생성 CloudWatch

CloudWatch 경보는 지정된 기간 동안 단일 지표를 감시하고 Amazon Simple Notification Service(AmazonSNS) 주제 또는 Auto Scaling 정책에 알림을 보내는 하나 이상의 작업을 수행합니다. 작업 또는 작업은 지정한 여러 기간 동안 지정된 임계값에 대한 지표 값을 기반으로 합니다.는 경보 상태가 변경될 때 Amazon SNS 메시지를 보낼 수도 CloudWatch 있습니다.

CloudWatch 경보는 상태가 변경되고 지정한 기간 동안 지속된 경우에만 작업을 호출합니다.

지표를 보려면(CloudWatch 콘솔)

1. AWS Management Console에 로그인하여 CloudWatch 콘솔을 엽니다.
2. 알람를 선택한 다음 알람 생성을 선택합니다.
3. AWS/HealthLake를 선택한 다음 지표를 선택합니다.
4. 시간 범위에서 모니터링할 시간 범위를 선택한 후, 다음를 선택합니다.
5. 이름 및 설명을 입력합니다.
6. Whenever에서 >=를 선택하고 최대값을 입력합니다.
7. 경보 상태에 도달했을 때 이메일을 보내 CloudWatch 려면 작업 섹션에서이 경보가 발생할 때마다 상태가 임을 선택합니다ALARM. 알림 전송 대상에서 메일링 목록을 선택하거나 새 목록을 선택하고 새 메일링 목록을 생성합니다.
8. 알람 미리보기 섹션에서 경보를 미리 볼 수 있습니다. 경보가 만족스러우면 경보 생성을 선택합니다.

SMART FHIR와의 통합 AWS HealthLake

FHIR 활성화된 HealthLake 데이터 스토어의 대체 가능한 의료 애플리케이션 및 재사용 가능한 기술(SMART)을 사용하면 FHIR 규정 준수 애플리케이션에서 HealthLake 데이터 스토어SMART에 저장된 데이터에 액세스할 수 있습니다. HealthLake 데이터는 타사 권한 부여 서버를 사용하여 요청을 인증 및 승인하고에 추가 리소스를 설정하여 액세스합니다 AWS.

HealthLake 데이터 스토어FHIR에서 SMART에서를 사용하려면 CreateFHIRDatastore API 요청에 다음을 제공해야 합니다.

- 를와 AuthorizationStrategy 동일하게 설정합니다`SMART_ON_FHIR_V1`.
- 권한 부여 서버로 토큰 디코딩을 관리하기 위해 AWS Lambda 생성한 ARN의와 IdpLambdaArn 동일하게를 설정합니다.
- 권한 부여 서버에 지정된 메타데이터 요소를 정의합니다. 이러한 메타데이터 요소는 검색 문서에 반환됩니다. 자세한 내용은 SMART FHIR 활성화된 HealthLake 데이터 스토어의 검색 문서에서 가져오기을 참조하십시오.
- 선택 사항: 권한 부여 서버에서 세분화된 권한 부여를 설정한 FineGrainedAuthorizationEnabled 경우를 활성화합니다.

AWS Command Line Interface (AWS CLI)를 사용하거나 AWS 지원되는 중 하나를 통해 FHIR 활성화된 데이터 스토어SMART에서를 만들 수 있습니다SDKs. FHIR 활성화된 HealthLake 데이터 스토어SMART에서를 생성하는 것은 HealthLake 콘솔을 사용하여 지원되지 않습니다. 자세한 내용은 FHIR 활성화된 데이터 스토어SMART에서 생성을 참조하십시오.

요청에 이러한 파라미터를 규정하려면 다른 AWS 서비스(AWS Secrets Manager 및)에서 리소스를 설정하고 AWS Lambda, 새 IAM 서비스 역할을 생성하고, 규정을 준수하는 권한 부여 서버에 SMART FHIR를 설정해야 합니다. 규정 FHIR 준수 데이터 스토어SMART에서를 구현하는 데 필요한 리소스 설정 섹션을 사용하여 필요한 리소스 설정에 대해 자세히 알아보고 SMART FHIR 애플리케이션의가 상호 작용하는 방식에 대한 개략적인 개요를 확인합니다 HealthLake.

즉, AWS Identity and Access Management 사용자를 통해 사용자 자격 증명을 관리하는 대신 FHIR 규정 준수 권한 부여 서버에서 SMART를 사용합니다.

HealthLake 는 FHIR 1.0SMART에서를 지원합니다. 이 프레임워크에 대한 자세한 내용은 SMART Application Launch Framework 구현 가이드 릴리스 1.0을 참조하세요.

SMART에서를 사용하여 데이터 스토어에 대한 요청을 승인하고 인증하려면 다음을 사용하여를 FHIR HealthLake 지원합니다.

- OpenID(AuthN) 통합: 해당 사용자 또는 클라이언트 애플리케이션이 이라고 주장하는 사람(또는 대상)임을 인증하는 데 사용됩니다.
- OAuth 2.0(AuthZ) 통합: 인증된 요청이 데이터를 읽거나 쓸 수 있는 HealthLake 데이터 스토어의 FHIR 리소스를 승인하는 데 사용됩니다. 이는 권한 부여 서버에 설정된 범위에 의해 정의됩니다.

목차

SMART의에 대한 인증 요구 사항 FHIR

FHIR HealthLake 데이터 스토어의에 있는 FHIR 리소스SMART에 액세스하려면 클라이언트 애플리케이션이 OAuth 2.0 준수 권한 부여 서버의 승인을 받아야 하며 FHIR REST API 요청의 일부로 OAuth 베어러 토큰을 제시해야 합니다. 권한 부여 서버의 엔드포인트를 찾으려면 잘 알려진 Uniform Resource Identifier를 HealthLake SMART 통해 FHIR on Discovery Document를 사용합니다. 이 프로세스에 대한 자세한 내용은 섹션을 참조하세요SMART FHIR 활성화된 HealthLake 데이터 스토어의 검색 문서에서 가져오기.

FHIR HealthLake 데이터 스토어SMART에서를 생성할 때 C reateFHIRDatastore 요청의 `metadata` 요소에 권한 부여 서버의 엔드포인트와 토큰 엔드포인트를 정의해야 합니다. `metadata` 요소를 정의하는 방법에 대한 자세한 내용은 섹션을 참조하세요FHIR 활성화된 HealthLake 데이터 스토어SMART에서 생성.

클라이언트 애플리케이션은 권한 부여 서버 엔드포인트를 사용하여 권한 부여 서비스로 사용자를 인증합니다. 승인 및 인증되면 JSON 웹 토큰(JWT)이 권한 부여 서비스에서 생성되어 클라이언트 애플리케이션으로 전달됩니다. 이 토큰에는 클라이언트 애플리케이션이 사용할 수 있는 FHIR 리소스 범위가 포함되어 있으므로 사용자가 액세스할 수 있는 데이터가 제한됩니다. 선택적으로 시작 범위가 제공된 경우 응답에 이러한 세부 정보가 포함됩니다. 에서 지원하는 FHIR 범위 내 SMART에 대한 자세한 내용은 섹션을 HealthLake참조하세요SMART에서 FHIR OAuth 범위 지원 HealthLake.

권한 부여 서버에서 JWT 부여한를 사용하여 클라이언트 애플리케이션은 FHIR 활성화된 HealthLake 데이터 스토어SMART에서를 FHIR REST API 호출합니다. 를 검증하고 디코딩하려면 Lambda 함수를 생성JWT해야 합니다.는 FHIR REST API 요청이 수신될 때 사용자를 대신하여이 Lambda 함수를 HealthLake 호출합니다. 예제 스타터 Lambda 함수를 보려면 섹션을 참조하세요FHIR 활성화된 HealthLake 데이터 스토어 AWS Lambda 의에서 토큰 검증SMART에 사용.

FHIR 활성화된 HealthLake 데이터 스토어SMART에서를 생성하는 데 필요한 권한 부여 서버 요소

C reateFHIRDatastore 요청에서는 `IdentityProviderConfiguration` 객체의 `metadata` 요소의 일부로 권한 부여 엔드포인트와 토큰 엔드포인트를 제공해야 합니다. 권한 부여 엔드포인트와 토큰 엔드포인트가 모두 필요합니다. C reateFHIRDatastore 요청에서 이를 지정하는 방법의 예를 보려면 섹션을 참조하세요FHIR 활성화된 HealthLake 데이터 스토어SMART에서 생성.

FHIR 활성화된 HealthLake 데이터 스토어에서에 SMART 대한 FHIR REST API 요청을 완료하는 데 필요한 클레임

AWS Lambda 활성화된 FHIR HealthLake 데이터 스토어의에 대한 유효한 FHIR REST API 요청으로 함수SMART에 다음 클레임이 포함되어야 합니다.

- `nbf`: (이전 아님) 클레임 - "nbf"(이전 아님) 클레임은가 처리에 수락되기 전의 시간을 식별JWTMUSTNOT합니다. “nbf” 클레임을 처리하려면 “nbf” 클레임에 date/time MUST be after or equal to the not-before date/time 나열된 현재가 필요합니다. 제공하는 샘플 Lambda 함수는 서버 응답`iat`에서 로 변환합니다`nbf`.
- `exp`: (만료 시간) 클레임 - "exp"(만료 시간) 클레임은가 처리에 수락되지 않아야 하는 만료 시간을 식별JWT합니다.
- `isAuthorized`: 로 설정된 부울입니다`True`. 요청이 권한 부여 서버에서 승인되었음을 나타냅니다.
- `aud`: (대상) 클레임 — "aud" (대상) 클레임JWT은가 의도한 수신자를 식별합니다. 이는 FHIR 활성화된 HealthLake 데이터 스토어 엔드포인트SMART의 이어야 합니다.
- `scope`: 하나 이상의 FHIR 리소스 관련 범위여야 합니다. 이 범위는 권한 부여 서버에서 정의됩니다. 에서 허용하는 FHIR 리소스 관련 범위에 대한 자세한 내용은 섹션을 HealthLake참조하세요HealthLake 데이터 스토어 FHIR 리소스별 범위.

SMART에서 FHIR OAuth 범위 지원 HealthLake

HealthLake 는 OAuth 2.0을 권한 부여 프로토콜로 사용합니다. 권한 부여 서버에서이 프로토콜을 사용하면 클라이언트 애플리케이션이 읽기 및/또는 쓰기 액세스 권한을 가질 수 있는 HealthLake 데이터 스토어의 FHIR 리소스를 정의할 수 있습니다.

FHIR 프레임워크SMART의는 권한 부여 서버에서 요청할 수 있는 범위 세트를 정의합니다. FHIR 프레임워크SMART에서 범위 정의를 보려면 HL7 FHIR 리소스 가이드SMART의 FHIR 범위 섹션을 참조하세요.

예를 들어, 환자가 실험실 결과를 보거나 연락처 세부 정보를 볼 수 있도록 하기 위해 설계된 클라이언트 애플리케이션은 (FHIRREST요청을 통해) `read` 범위만 요청할 수 있는 권한이 있어야 합니다. 이를 범위로 정의하려면 다음과 같은 문자열을 제공합니다`patient/Observation.read`. 이렇게 하면 클라이언트 애플리케이션이 `Observation` 리소스 유형에서 읽기 전용 방식으로 `Patient` 리소스 유형에 대한 액세스를 요청할 수 있습니다.

독립 실행형 시작 범위

HealthLake 는 독립 실행형 시작 모드 범위를 지원합니다`launch/patient`.

독립 실행형 시작 모드에서는 사용자와 환자가 클라이언트 애플리케이션에 알려지지 않았기 때문에 클라이언트 애플리케이션이 환자의 임상 데이터에 대한 액세스를 요청합니다. 따라서 클라이언트 애플리케이션의 권한 부여 요청은 환자 범위 반환을 명시적으로 요청합니다. 인증에 성공하면 권한 부여 서버는 요청된 시작 환자 범위가 포함된 액세스 토큰을 발급합니다. 필요한 환자 컨텍스트는 권한 부여 서버의 응답에서 액세스 토큰과 함께 제공됩니다.

지원되는 시작 모드 범위

범위	설명
`launch/patient`	권한 부여 응답에서 환자 데이터를 반환하도록 요청하는 OAuth 2.0 권한 부여 요청의 파라미터입니다.

HealthLake 데이터 스토어 FHIR 리소스별 범위

HealthLake 는 세 가지 수준의 범위를 정의합니다.

- 환자별 범위는 단일 환자에 대한 특정 데이터에 대한 액세스 권한을 부여합니다. 시작 컨텍스트에 지정된 환자입니다.
- 사용자 수준 범위는 사용자가 액세스할 수 있는 특정 데이터에 대한 액세스 권한을 부여합니다.
- 시스템 수준 범위는 HealthLake 데이터 스토어에 있는 모든 FHIR 리소스에 대한 읽기/쓰기 액세스 권한을 부여합니다.

다음 표에는에서 지원하는 FHIR 리소스 관련 범위를 구성하기 위한 구문이 나와 있습니다 HealthLake. 일반 형식은 다음과 같습니다.

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

HealthLake 데이터 스토어에서 지원되는 권한 부여 범위

범위 구문	예제 범위	결과
`patient/(fhir-resource | '*').('read' | 'write' | '*')`	`patient/AllergyIntolerance.*`	클라이언트 애플리케이션은 알러지에 대한 읽기/쓰기 액세스 권한이 있습니다.
`user/(fhir-resource | '*').('read' | 'write' | '*')`	`user/Observation.read`	클라이언트 애플리케이션은 기록된 모든 관찰에 대한 읽기 액세스 권한을 갖게 됩니다.
`system/('read' | 'write' | *)`	`system/*.*`	클라이언트 애플리케이션에는 모든 데이터에 대한 읽기/쓰기 액세스 권한이 있습니다.

FHIR 활성화된 HealthLake 데이터 스토어 AWS Lambda 의에서 토큰 검증SMART에 사용

FHIR 활성화된 HealthLake 데이터 스토어SMART에서를 생성할 때 `CreateFHIRDatastore` 요청에 AWS Lambda 함수ARN의를 제공해야 합니다. Lambda 함수의는 `IdpLambdaArn` 파라미터를 사용하여 `IdentityProviderConfiguration` 객체에 지정ARN됩니다.

FHIR 활성화된 HealthLake 데이터 스토어에서를 생성하기 전에 Lambda 함수SMART를 생성해야 합니다. 데이터 스토어를 생성한 후에는 Lambda를 변경할 수 ARN 없습니다. 데이터 스토어가 생성될 때 ARN 지정한 Lambda를 보려면 `DescribeFHIRDatastore` API 작업을 사용합니다.

FHIR 활성화된 HealthLake 데이터 스토어의 SMART에서 FHIR REST 요청이 성공하려면 Lambda 함수가 다음을 수행해야 합니다.

- Lambda 함수는 HealthLake 데이터 스토어 엔드포인트에 1초 이내에 응답을 반환해야 합니다.
- 클라이언트 애플리케이션에서 보낸 REST API 요청의 권한 부여 헤더에 제공된 액세스 토큰을 디코딩합니다.
- FHIR REST API 요청을 수행할 수 있는 충분한 권한이 있는 IAM 서비스 역할을 할당합니다.

- FHIR REST API 요청을 완료하려면 다음 클레임이 필요합니다. 자세한 내용은 필수 클레임을 참조하십시오.
  - `nbf`
  - `exp`
  - `isAuthorized`
  - `aud`
  - `scope`

Lambda로 작업할 때는 Lambda 함수 외에도 실행 역할과 리소스 기반 정책을 생성해야 합니다. Lambda 함수의 실행 역할은 함수에 런타임에 필요한 AWS 서비스 및 리소스에 액세스할 수 있는 권한을 부여하는 IAM 역할입니다. 제공하는 리소스 기반 정책은가 사용자를 대신하여 함수를 호출 HealthLake 하도록 허용해야 합니다.

이 주제의 단원에서는 클라이언트 애플리케이션의 예제 요청과 디코딩된 응답, AWS Lambda 함수를 생성하는 데 필요한 단계,가 수임할 HealthLake 수 있는 리소스 기반 정책을 생성하는 방법을 설명합니다.

- 1부: Lambda 함수 생성
- 2부: AWS Lambda 함수에서 사용하는 HealthLake 서비스 역할 생성
- 3부: Lambda 함수의 실행 역할 업데이트
- 4부: Lambda 함수에 리소스 정책 추가
- 5부: Lambda 함수에 대한 동시성 프로비저닝

AWS Lambda 함수 생성

이 주제에서 생성된 Lambda 함수는가 FHIR 활성화된 HealthLake 데이터 스토어에서 SMART에 대한 요청을 HealthLake 수신할 때 트리거됩니다. 클라이언트 애플리케이션의 요청에는 REST API 호출과 액세스 토큰이 포함된 권한 부여 헤더가 포함됩니다.

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

이 주제의 Lambda 함수 예제는 AWS Secrets Manager 를 사용하여 권한 부여 서버와 관련된 자격 증명을 가립니다. Lambda 함수에서 권한 부여 서버 로그인 세부 정보를 직접 제공하지 않는 것이 좋습니다.

Example 권한 부여 전달자 토큰이 포함된 FHIR REST 요청 검증

Lambda 함수 예제는 FHIR 활성화된 HealthLake 데이터 스토어SMART에서 로 전송된 FHIR REST 요청을 검증하는 방법을 보여줍니다. 이 Lambda 함수를 구현하는 방법에 대한 step-by-steps 지침은 섹션을 참조하세요를 사용하여 Lambda 함수 생성 AWS Management Console.

FHIR REST API 요청에 유효한 데이터 스토어 엔드포인트, 액세스 토큰 및 REST 작업이 포함되어 있지 않으면 Lambda 함수가 실패합니다. 필수 권한 부여 서버 요소에 대한 자세한 내용은 섹션을 참조하세요필수 클레임.

```
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()
```

를 사용하여 Lambda 함수 생성 AWS Management Console

이 절차에서는 SMART FHIR 활성화된 HealthLake 데이터 스토어에서에 대한 FHIR REST API 요청을 처리할 때 수임 HealthLake 하려는 서비스 역할을 이미 생성했다고 가정합니다. 서비스 역할을 생성하지 않은 경우에도 Lambda 함수를 생성할 수 있습니다. Lambda 함수가 작동하려면 먼저 서비스 역할 ARN의를 추가해야 합니다. 서비스 역할을 생성하고 Lambda 함수에서 지정하는 방법에 대한 자세한 내용은 섹션을 참조하세요. 를 디코딩하는 데 사용되는 AWS Lambda 함수에 사용할 HealthLake 서비스 역할 생성 JWT

Lambda 함수를 생성하려면(AWS Management Console)

1. Lambda 콘솔의 함수 페이지를 엽니다.
2. 함수 생성(Create function)을 선택합니다.
3. 새로 작성을 선택합니다.

4. 기본 정보에서 함수 이름을 입력합니다. 런타임에서 Python 기반 런타임을 선택합니다.
5. Execution role(실행 역할)에서 Create a new role with basic Lambda permissions(기본 Lambda 권한을 가진 새 역할 생성)를 선택합니다.

   Lambda는 함수에 Amazon에 로그를 업로드할 수 있는 권한을 부여하는 실행 역할을 생성합니다 CloudWatch. Lambda 함수는 함수를 호출할 때 실행 역할을 수임하고 실행 역할을 사용하여에 대한 자격 증명을 생성합니다 AWS SDK.
6. 코드 탭을 선택하고 샘플 Lambda 함수를 추가합니다.

   Lambda 함수가 사용할 서비스 역할을 아직 생성하지 않은 경우 샘플 Lambda 함수가 작동하기 전에 생성해야 합니다. Lambda 함수에 대한 서비스 역할 생성에 대한 자세한 내용은 섹션을 참조하세요를 디코딩하는 데 사용되는 AWS Lambda 함수에 사용할 HealthLake 서비스 역할 생성 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()
```

Lambda 함수의 실행 역할 수정

Lambda 함수를 생성한 후 Secrets Manager를 호출하는 데 필요한 권한을 포함하도록 실행 역할을 업데이트해야 합니다. Secrets Manager에서 생성하는 각 보안 암호에는이 있습니다ARN. 최소 권한을 적용하려면 실행 역할에 Lambda 함수가 실행하는 데 필요한 리소스에만 액세스할 수 있어야 합니다.

콘솔에서 검색하거나 Lambda IAM 콘솔에서 구성을 선택하여 Lambda 함수의 실행 역할을 수정할 수 있습니다. Lambda 함수 실행 역할 관리에 대한 자세한 내용은 섹션을 참조하세요Lambda 실행 역할.

Example 에 대한 액세스 권한을 부여하는 Lambda 함수 실행 역할 **GetSecretValue**

실행 역할에 IAM 작업을 추가GetSecretValue하면 샘플 Lambda 함수가 작동하는 데 필요한 권한이 부여됩니다.

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

이 시점에서 FHIR 활성화된 HealthLake 데이터 스토어에서에 전송된 FHIR REST 요청의 일부로 제공된 액세스 토큰을 검증하는 데 사용할 수 있는 Lambda 함수SMART를 생성했습니다.

를 디코딩하는 데 사용되는 AWS Lambda 함수에 사용할 HealthLake 서비스 역할 생성 JWT

> 페르소나: IAM 관리자
>
> IAM 정책을 추가 또는 제거하고 새 IAM 자격 증명을 생성할 수 있는 사용자입니다.
>
> 서비스 역할
>
> 서비스 역할은 서비스가 사용자를 대신하여 작업을 수행하기 위해 수임하는 IAM 역할입니다. IAM 관리자는 IAM 내에서 서비스 역할을 생성, 수정, 삭제할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 에 권한을 위임할 역할 생성을 AWS 서비스 참조하세요.

JSON 웹 토큰(JWT)이 디코딩된 후 Lambda는 IAM 역할 도 반환해야 합니다ARN. 이 역할에는 REST API 요청을 수행하는 데 필요한 권한이 있어야 합니다. 그렇지 않으면 권한이 부족하여 실패합니다.

를 사용하여 사용자 지정 정책을 설정할 때는 필요한 최소 권한을 부여하는 IAM 것이 가장 좋습니다. 자세한 내용은 IAM 사용 설명서의 최소 권한 적용을 참조하세요.

권한 부여 Lambda 함수에서 지정할 HealthLake 서비스 역할을 생성하려면 두 단계가 필요합니다.

- 먼저 IAM 정책을 생성해야 합니다. 정책은 권한 부여 서버에서 범위를 제공한 FHIR 리소스에 대한 액세스를 지정해야 합니다.
- 둘째, 서비스 역할을 생성해야 합니다. 역할을 생성할 때 신뢰 관계를 지정하고 1단계에서 생성한 정책을 연결합니다. 신뢰 관계는를 서비스 보안 주체 HealthLake 로 지정합니다. 이 단계에서는 HealthLake 데이터 스토어ARN와 AWS 계정 ID를 지정해야 합니다.

새 IAM 정책 생성

권한 부여 서버에서 정의한 범위에 따라 인증된 사용자가 HealthLake 데이터 스토어에서 액세스할 수 있는 FHIR 리소스가 결정됩니다.

생성한 IAM 정책은 정의한 범위에 맞게 조정할 수 있습니다.

IAM 정책 설명의 `Action` 요소에 다음 작업을 정의할 수 있습니다. 테이블의 각 `Action`에 대해를 정의할 수 있습니다`Resource types`. HealthLake 데이터 스토어는 IAM 권한 정책 문의 `Resource` 요소에 정의할 수 있는 유일한 지원 리소스 유형입니다.

개별 FHIR 리소스는 IAM 권한 정책의 요소로 정의할 수 있는 리소스가 아닙니다.

에서 정의한 작업 HealthLake

작업	설명	액세스 레벨	리소스 유형(필수)
CreateResource	리소스 생성 권한을 부여합니다.	쓰기	데이터 스토어 ARN: arn:aws:healthlake:**your-region**:**111122223333**:datastore/fhir/**your-datastore-id**
DeleteResource	리소스를 삭제할 수 있는 권한을 부여합니다.	쓰기	데이터 스토어 ARN: arn:aws:healthlake:**your-region**:**111122223333**:datastore/fhir/**your-datastore-id**
ReadResource	리소스를 읽을 수 있는 권한을 부여합니다.	읽기	데이터 스토어 ARN: arn:aws:healthlake:**your-region**:**111122223333**:datastore/fhir/**your-datastore-id**

작업	설명	액세스 레벨	리소스 유형(필수)
SearchWithGet	GET 메서드로 리소스를 검색할 수 있는 권한을 부여합니다.	읽기	데이터 스토어 ARN: arn:aws:healthlake:**your-region**:**111122223333**:datastore/fhir/**your-datastore-id**
SearchWithPost	POST 메서드로 리소스를 검색할 수 있는 권한을 부여합니다.	읽기	데이터 스토어 ARN: arn:aws:healthlake:**your-region**:**111122223333**:datastore/fhir/**your-datastore-id**
StartFHIRExportJobWithPost	를 사용하여 FHIR 내보내기 작업을 시작할 수 있는 권한을 부여합니다. GET	쓰기	데이터 스토어 ARN: arn:aws:healthlake:**your-region**:**111122223333**:datastore/fhir/**your-datastore-id**
UpdateResource	리소스를 업데이트할 수 있는 권한을 부여합니다.	쓰기	데이터 스토어 ARN: arn:aws:healthlake:**your-region**:**111122223333**:datastore/fhir/**your-datastore-id**

시작하려면를 사용할 수 있습니다`AmazonHealthLakeFullAccess`. 이 정책은 데이터 스토어에 있는 모든 FHIR 리소스에 대해 읽기, 쓰기, 검색 및 내보내기를 부여합니다. 데이터 스토어에 대한 읽기 전용 권한을 부여하려면를 사용합니다`AmazonHealthLakeReadOnlyAccess`.

AWS Management Console AWS CLI또는 IAM를 사용하여 사용자 지정 정책을 생성하는 방법에 대한 자세한 내용은 IAM 사용 설명서의 정책 생성을 IAM SDKs참조하세요.

에 대한 서비스 역할 생성 HealthLake (IAM 콘솔)

이 절차를 사용하여 서비스 역할을 생성합니다. 서비스를 생성할 때 IAM 정책을 지정해야 합니다.

에 대한 서비스 역할을 생성하려면 HealthLake (IAM 콘솔)

1. 에 로그인 AWS Management Console 하고에서 IAM 콘솔을 엽니다https://console.aws.amazon.com/iam/.
2. IAM 콘솔의 탐색 창에서 역할을 선택합니다.

3. 그런 다음 역할 생성을 선택합니다.
4. 신뢰 개체 선택 페이지에서 사용자 지정 신뢰 정책을 선택합니다.
5. 다음으로 사용자 지정 신뢰 정책에서 다음과 같이 샘플 정책을 업데이트합니다. **your-account-id**를 계정 번호로 바꾸고 가져오기 또는 내보내기 작업에 사용할 데이터 스토어ARN의를 추가합니다.

```
{
  "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. 그리고 다음을 선택합니다.
7. 권한 추가 페이지에서 HealthLake 서비스가 수임할 정책을 선택합니다. 정책을 찾으려면 권한 정책에서 해당 정책을 검색합니다.
8. 그런 다음 정책 연결을 선택합니다.
9. 그런 다음 역할 이름 아래의 이름, 검토 및 생성 페이지에서 이름을 입력합니다.
10. (선택 사항) 그런 다음 설명 아래에 역할에 대한 간단한 설명을 추가합니다.
11. 가능한 경우 이 역할의 목적을 식별하는 데 도움이 되는 역할 이름이나 역할 이름 접미사를 입력합니다. 역할 이름은 내에서 고유해야 합니다 AWS 계정. 대소문자는 구별하지 않습니다. 예를 들어, 이름이 **PRODROLE**과 **prodrole**, 두 가지로 지정된 역할을 만들 수는 없습니다. 다양한 주체가 역할을 참조할 수 있기 때문에 역할이 생성된 후에는 역할 이름을 편집할 수 없습니다.
12. 역할 세부 정보를 검토한 다음 역할 생성을 선택합니다.

샘플 Lambda 함수ARN에서 역할을 지정하는 방법을 알아보려면 섹션을 참조하세요AWS Lambda 함수 생성.

Lambda 실행 역할

Lambda 함수의 실행 역할은 함수에 AWS 서비스 및 리소스에 액세스할 수 있는 권한을 부여하는 IAM 역할입니다. 이 페이지에서는 Lambda 함수의 실행 역할을 생성하고 보고 관리하는 방법에 대한 정보를 제공합니다.

기본적으로 Lambda는를 사용하여 새 Lambda 함수를 생성할 때 최소한의 권한으로 실행 역할을 생성합니다 AWS Management Console. 실행 역할에서 부여된 권한을 관리하려면 Lambda 개발자 안내서의 IAM 콘솔에서 실행 역할 생성을 참조하세요.

이 주제에 제공된 샘플 Lambda 함수는 Secrets Manager를 사용하여 권한 부여 서버의 자격 증명을 가립니다.

생성하는 모든 IAM 역할과 마찬가지로 최소 권한 모범 사례를 따르는 것이 중요합니다. 개발 구문 중에 필요한 것 이상의 권한을 부여하는 경우가 있습니다. 모범 사례로 프로덕션 환경에 함수를 게시하기 전에 필요한 권한만 포함하도록 정책을 조정하는 것이 가장 좋습니다. 자세한 내용은 IAM 사용 설명서의 최소 권한 적용을 참조하세요.

가 Lambda 함수를 트리거 HealthLake 하도록 허용

HealthLake 가 사용자를 대신하여 Lambda 함수를 호출할 수 있으므로 다음을 수행해야 합니다.

- `CreateFHIRDatastore` 요청에서 호출 HealthLake 하려는 Lambda 함수ARN의와 `IdpLambdaArn` 동일하게를 설정해야 합니다.
- 가 사용자를 대신하여 Lambda 함수를 호출 HealthLake 하도록 허용하는 리소스 기반 정책이 필요합니다.

가 FHIR 활성화된 HealthLake 데이터 스토어의 SMART에 대한 FHIR REST API 요청을 HealthLake 수신하면 사용자를 대신하여 데이터 스토어 생성 시 지정된 Lambda 함수를 호출할 권한이 필요합니다. HealthLake 액세스 권한을 부여하려면 리소스 기반 정책을 사용합니다. Lambda 함수에 대한 리소스 기반 정책 생성에 대한 자세한 내용은 AWS Lambda 개발자 안내서의 AWS 서비스에서 Lambda 함수를 호출하도록 허용을 참조하세요.

Lambda 함수에 대한 동시성 프로비저닝

> **Important**
>
> HealthLake 에서는 Lambda 함수의 최대 실행 시간이 1초(1000밀리초) 미만이어야 합니다. Lambda 함수가 실행 시간 제한을 초과하면 TimeOut 예외가 발생합니다.

이 예외를 방지하려면 프로비저닝된 동시성을 구성하는 것이 좋습니다. 호출이 증가하기 전에 프로비저닝된 동시성을 할당하면 짧은 지연 시간으로 초기화된 인스턴스에서 모든 요청을 처리하도록 할 수 있습니다. 프로비저닝된 동시성 구성에 대한 자세한 내용은 Lambda 개발자 안내서의 프로비저닝된 동시성 구성을 참조하세요.

Lambda 함수의 평균 실행 시간을 보려면 현재 Lambda 콘솔에서 Lambda 함수의 모니터링 페이지를 사용합니다. 기본적으로 Lambda 콘솔은 함수 코드가 이벤트를 처리하는 데 소요한 평균, 최소 및 최대 시간을 보여주는 기간 그래프를 제공합니다. Lambda 함수 모니터링에 대한 자세한 내용은 Lambda 개발자 안내서의 Lambda 콘솔에서 함수 모니터링을 참조하세요.

Lambda 함수에 대한 동시성을 이미 프로비저닝하고 이를 모니터링하려면 Lambda 개발자 안내서의 동시성 모니터링을 참조하세요.

FHIR 활성화된 HealthLake 데이터 스토어SMART에서 생성

와 함께 FHIR 프레임워크SMART에서를 사용하려면 C reateFHIRDatastore 요청에 지정된 `IdentityProviderConfiguration` 파라미터로 HealthLake 데이터 스토어를 HealthLake생성합니다. `IdentityProviderConfiguration` 파라미터에서 다음 정보를 지정합니다.

- 를와 AuthorizationStrategy 동일하게 설정합니다`SMART_ON_FHIR_V1`.
- 권한 부여 서버로 토큰 디코딩을 관리하기 위해 AWS Lambda 생성한 ARN의와 IdpLambdaArn 동일하게를 설정합니다.
- 권한 부여 서버에 JSON 블록으로 지정된 메타데이터 요소를 정의합니다. 이러한 메타데이터 요소는 검색 문서에 반환됩니다.
- 선택 사항:를 활성화합니다FineGrainedAuthorizationEnabled. 에서 제공하는 세분화된 권한 부여를 사용하려면 `True`를 지정합니다. HealthLake

AWS Command Line Interface (AWS CLI)를 사용하거나 AWS 지원되는 중 하나를 통해 FHIR 활성화된 데이터 스토어SMART에서를 만들 수 있습니다SDKs. FHIR 활성화된 HealthLake 데이터 스토어SMART에서를 생성하는 것은 HealthLake 콘솔을 사용하여 지원되지 않습니다.

AWS CLI 를 사용하여 FHIR 활성화된 HealthLake 데이터 스토어SMART에서 생성

다음 코드 예제를 사용하여를 사용하여 FHIR 활성화된 HealthLake 데이터 스토어SMART에서를 생성할 수 있습니다 AWS CLI. FHIR 활성화된 HealthLake 데이터 스토어SMART에서를 생성할 때는 `identity-provider-configuration` 파라미터를 지정해야 합니다.

`identity-provider-configuration` 파라미터에서를 `FineGrainedAuthorizationEnabled`로 설정하여 세분화된 권한 부여를 선택적으로 활성화할 수 있습니다`True`. 세분화된 권한 부여에 대한 자세한 내용은 섹션을 참조하세요FHIR 활성화된 HealthLake 데이터 스토어SMART에서와 함께 세분화된 권한 부여 사용. 아래 예제에는 줄 바꿈 또는 이스케이프 문자를 \ 나타내는 특수 문자가 포함되어 있습니다. 이는 명확성을 위한 것입니다.

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

성공하면 다음과 같은 JSON 응답을 받게 됩니다.

```
{
  "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"
}
```

FHIR 활성화된 HealthLake 데이터 스토어SMART에서와 함께 세분화된 권한 부여 사용

범위만으로는 요청자가 데이터 스토어에서 액세스할 권한이 있는 데이터에 대한 필수 세부 정보를 제공하지 않습니다. 세분화된 권한 부여를 사용하면 FHIR 활성화된 HealthLake 데이터 스토어의 SMART에 대한 액세스 권한을 부여할 때 더 높은 수준의 특이성을 사용할 수 있습니다. 세분화된 권한 부여를 사용하려면 C reateFHIRDatastore 요청의 `IdentityProviderConfiguration` 파라미터`True`에서를 `FineGrainedAuthorizationEnabled`로 설정합니다.

세분화된 권한 부여를 활성화한 경우 권한 부여 서버는 액세스 토큰과 `id_token` 함께의 `fhirUser` 범위를 반환합니다. 이렇게 하면 클라이언트 애플리케이션에서 사용자에 대한 정보를 검색할 수 있습니다. 클라이언트 애플리케이션은 `fhirUser` 클레임을 현재 사용자를 나타내는 FHIR 리소스URI의 로 취급해야 합니다. `Patient`, `Practitioner` 또는 `RelatedPerson` 유형을 지정할 수 있습니다. 권한 부여 서버의 응답에는 사용자가 액세스할 수 있는 데이터를 정의하는 `user/` 범위도 포함됩니다. 리소스FHIR별 범위와 관련된 범위에 대해 정의된 구문을 사용합니다.

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

다음은 세분화된 권한 부여를 사용하여 데이터 액세스 관련 FHIR 리소스 유형을 추가로 지정하는 방법의 예입니다.

- `fhirUser`가 인 경우 `Practitioner`세분화된 권한 부여에 따라 사용자가 액세스할 수 있는 환자 모음이 결정됩니다. 에 대한 액세스`fhirUser`는 환자가 일반 프랙티셔너`fhirUser`로를 참조하는 경우에만 허용됩니다.

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

- fhirUser가 Patient 또는 이고 요청에서 참조된 환자가 RelatedPerson와 다른 경우 fhirUser세분화된 권한 부여를 통해 요청된 환자의에 fhirUser 대한 액세스가 결정됩니다. 요청된 Patient 리소스에 지정된 관계가 있는 경우 액세스가 허용됩니다.

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

SMART FHIR 활성화된 HealthLake 데이터 스토어의 검색 문서에서 가져오기

클라이언트 애플리케이션이 성공적으로 FHIR REST 요청하려면 HealthLake 데이터 스토어에 정의된 권한 부여 요구 사항을 수집해야 합니다. 이 요청이 성공하려면 권한 부여(베어러 토큰)가 필요하지 않습니다.

이렇게 하려면 GET 요청을 하고 데이터 스토어의 /.well-known/smart-configuration 엔드포인트에 추가합니다.

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

이렇게 JSON 하면 HealthLake 데이터 스토어의 검색 문서가 Blob으로 반환됩니다. 여기에서 HealthLake 데이터 스토어에 정의된 사양 및 기능과 token_endpoint 함께 authorization_endpoint 및를 찾을 수 있습니다.

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

URLs 클라이언트 애플리케이션을 성공적으로 시작하는 데 필요

- 권한 부여 엔드포인트: 클라이언트 애플리케이션 또는 사용자에게 권한을 부여하는 데 URL 필요한 입니다.
- 토큰 엔드포인트: 클라이언트 애플리케이션이 토큰 엔드포인트와 통신하는 데 사용하는 권한 부여 서버의 엔드포인트입니다.

SMART 활성화된 HealthLake 데이터 스토어에서 FHIR REST API 요청

FHIR활성화된 HealthLake 데이터 스토어의 SMART에서 FHIR REST API 요청할 수 있습니다. 다음 예제에서는 권한 부여 헤더JWT에가 포함된 클라이언트 애플리케이션의 요청과 Lambda가 응답을 디코딩하는 방법을 보여줍니다. 클라이언트 애플리케이션 요청이 승인 및 인증된 후에는 권한 부여 서버로부터 무기명 토큰을 받아야 합니다. 에서 SMART FHIR활성화된 HealthLake 데이터 스토어에서에 대한 FHIR REST API 요청을 보낼 때 권한 부여 헤더의 베어러 토큰을 사용합니다.

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

권한 부여 헤더에서 베어러 토큰이 발견되고 ID가 감지되지 AWS IAM 않았기 때문에 SMART FHIR 활성화된 HealthLake 데이터 스토어의가 생성될 때 지정된 Lambda 함수를 HealthLake 호출합니다. 여기에서 Lambda 함수에 의해 토큰이 성공적으로 디코딩되는 경우 로 전송되는 예제 응답입니다 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
}
```

규정 준수 데이터 스토어SMART에서 FHIR를 구현하는 데 필요한 리소스 설정

이 주제에서는 외부 AWS 계정에서 프로비저닝해야 하는 리소스 HealthLake, FHIR 활성화된 HealthLake 데이터 스토어SMART에서 생성, FHIR 클라이언트 애플리케이션의이 권한 부여 서버 및 HealthLake 데이터 스토어와 상호 작용하는 방법에 SMART 대해 설명합니다.

이 워크플로의 단계는 SMART FHIR 요청 처리 방법의 기본 단계와 요청에 필요한 리소스를 정의합니다.

SMART FHIR 요청 시 프로세스에서 세 개의 애플리케이션이 함께 작동합니다.

- 최종 사용자: 일반적으로 FHIR 애플리케이션 SMART 상의 타사를 사용하여 HealthLake 데이터 스토어의 데이터에 액세스하는 환자 또는 임상의입니다.
- SMART FHIR 애플리케이션(클라이언트 애플리케이션이라고 함)의 : HealthLake 데이터 스토어에 있는 데이터에 액세스하려는 애플리케이션입니다.
- 권한 부여 서버: 사용자를 인증하고 액세스 토큰을 발급할 수 있는 OpenID Connect 호환 서버입니다.
- HealthLake 데이터 스토어: Lambda 함수를 사용하여 베어러 토큰을 제공하는 FHIR REST 요청에 응답하는 FHIR 활성화된 HealthLake 데이터 스토어SMART의 입니다.

이러한 애플리케이션이 함께 작동하려면 다음 리소스를 생성해야 합니다.

권한 부여 서버를 설정하고 필요한 범위를 정의한 후 토큰 내부 검사를 처리하는 AWS Lambda 함수를 생성한 후 SMART FHIR 활성화된 HealthLake 데이터 스토어에서를 생성하는 것이 좋습니다.

1. 권한 부여 서버 엔드포인트 설정 - 권한 부여 서버

FHIR 프레임워크SMART에서를 사용하려면 데이터 스토어에서 이루어진 FHIR REST 요청을 검증할 수 있는 타사 권한 부여 서버를 설정해야 합니다. 에서 사용할 권한 부여 서버 엔드포인트 설정에 대한 자세한 내용은 섹션을 HealthLake참조하세요SMART의에 대한 인증 요구 사항 FHIR.

2. 권한 부여 서버의 데이터 HealthLake 스토어에 있는 데이터에 액세스할 수 있는 사용자를 제어하는 범위 정의 - 권한 부여 서버

SMART on FHIR Framework는 OAuth 범위를 사용하여 인증된 요청이 액세스할 수 있는 FHIR 리소스와 정도를 결정합니다. 범위를 정의하는 것은 최소 권한을 설계하는 방법입니다. FHIR 프레임워크

SMART에서에 의해 정의되고에서 지원하는 범위에 대한 자세한 내용은 섹션을 HealthLake 참조하세요SMART에서 FHIR OAuth 범위 지원 HealthLake.

3. 토큰 내부 검사를 수행할 수 있는 AWS Lambda 함수 설정 - AWS 계정

SMART FHIR 활성화된 데이터 스토어의에서 클라이언트 애플리케이션이 보낸 FHIR REST 요청에는 JSON 웹 토큰()이 포함됩니다JWT. Lambda 함수를 디코딩하고 검증할 수 있도록 설정하는 방법에 대한 자세한 내용은 의 디코딩을 JWT참조하세요.

4. FHIR 활성화된 HealthLake 데이터 스토어SMART에서 생성 - AWS 계정

FHIR HealthLake 데이터 스토어SMART에서를 생성하려면를 제공해야 합니다`IdentityProviderConfiguration`. C reateFHIRDatastore 요청에서 필요한 `IdentityProviderConfiguration` 파라미터를 자세히 알아보려면 FHIR 활성화된 HealthLake 데이터 스토어SMART에서 생성 단원을 참조하십시오.

클라이언트 애플리케이션이 데이터 스토어 FHIR 활성화SMART의에서 HealthLake 데이터를 시작하고 요청하는 방법

이 섹션에서는 클라이언트 애플리케이션이 SMART FHIR 컨텍스트에서를 사용하여 시작되고 HealthLake 데이터 스토어에서 성공적으로 FHIR REST 요청할 수 있는 방법을 설명합니다.

1. 클라이언트 애플리케이션이 잘 알려진 Uniform Resource Identifier에 **`GET`** 요청

SMART 활성화된 클라이언트 애플리케이션은 HealthLake 데이터 스토어의 권한 부여 엔드포인트를 찾기 위해 `GET` 요청해야 합니다. 이는 잘 알려진 Uniform Resource Identifier(URI) 요청을 통해 수행됩니다. 이에 대한 자세한 내용은 SMART FHIR 활성화된 HealthLake 데이터 스토어의 검색 문서에서 가져오기를 참조하세요.

2. 액세스 및 범위 요청

클라이언트 애플리케이션은 사용자가 로그인할 수 있도록 권한 부여 서버의 권한 부여 엔드포인트를 사용합니다. 이 프로세스는 사용자를 인증합니다. 범위는 클라이언트 애플리케이션이 액세스할 수 있는 HealthLake 데이터 스토어의 FHIR 리소스를 정의하는 데 사용됩니다. 범위 정의에 대한 자세한 내용은 섹션을 참조하세요SMART에서 FHIR OAuth 범위 지원 HealthLake.

3. 액세스 토큰

이제 사용자가 인증되었으므로 클라이언트 애플리케이션은 권한 부여 서버로부터 JWT 액세스 토큰을 받습니다. 이 토큰은 클라이언트 애플리케이션이 FHIR REST 요청을 보낼 때 제공됩니다 HealthLake.

JWT가 Lambda 함수를 사용하여 디코딩되는 방법에 대한 자세한 내용은 섹션을 참조하세요토큰 검증 수행.

4. FHIR 활성화된 HealthLake 데이터 스토어에서 SMART에 FHIR REST 요청

이제 클라이언트 애플리케이션은 권한 부여 서버에서 제공하는 액세스 토큰을 사용하여 HealthLake 데이터 스토어 엔드포인트에 FHIR REST 요청을 보낼 수 있습니다. 예제 FHIR REST 요청을 보려면 섹션을 참조하세요SMART 활성화된 HealthLake 데이터 스토어에서 FHIR REST API 요청.

5. JWT 액세스 토큰 검증

FHIR REST 요청에서 전송된 액세스 토큰을 검증하려면 Lambda 함수를 사용합니다. 토큰 내부 검사를 수행할 수 있는 Lambda 함수를 생성하는 방법은 단원을 참조하십시오AWS Lambda 함수 생성.

에서 리소스 유형의 자연어 처리(NLP)를 기반으로 자동화된 FHIR DocumentReference 리소스 생성 사용 AWS HealthLake

> **Note**
>
> 2023년 2월 20일 이후 HealthLake 데이터 스토어는 기본적으로 통합 자연어 처리(NLP)를 사용하지 않습니다. 데이터 스토어에서이 기능을 활성화하려면 문제 해결 HealthLake의 통합 자연어 처리 기능을 켜려면 어떻게 해야 합니까? 장의 섹션을 참조하세요.
>
> Amazon Comprehend Medical의 통합를 켠 NLP경우 `DocumentReference` 리소스를 생성하거나 업데이트할 때 AWS 계정에 요금이 발생합니다. 자세한 내용은 AWS HealthLake 요금을 참조하세요.
>
> Amazon Comprehend Medical은 아시아 태평양(뭄바이)에서 사용할 수 없습니다. 아시아 태평양(뭄바이) 리전에서 생성된 HealthLake 데이터 스토어는 통합 자연어 처리()를 지원하지 않습니다NLP.

HealthLake 는 `DocumentReference` 리소스 유형에 저장된 데이터에 대한 비정형 데이터 처리를 위해 Amazon Comprehend Medical을 사용하여 통합 자연어 처리(NLP)를 자동으로 제공합니다. 이렇게 하려면 Amazon Comprehend Medical `DetectEntities-V2`, `InferICD10-CM`및 `InferRxNorm` API 작업을 HealthLake 호출합니다. 결과는 `DocumentReference` 리소스에 확장 프로그램으로 자동으로 추가됩니다. Amazon Comprehend Medical API 작업이 `SIGN`, `SYMPTOM`및 인 특성을 감지하면 `DIAGNOSIS` `Linkage` 리소스 유형이 자동으로 생성됩니다. 새 조건 및 관찰 리소스는 `SIGN`, `SYMPTOM`또는의 특성으로 식별된 개체에서 생성`DIAGNOSIS`되며이 연결 리소스를 사용하여 소스 문서에 연결됩니다.

통합에서 생성된 리소스의 경우 `GET` 요청을 NLP할 수 있지만 이러한 새 리소스 검색은 지원되지 않습니다.

HealthLake의 Athena와의 통합을 사용하여 이러한 확장을 검색하는 방법에 대한 자세한 내용은 섹션을 참조하세요를 사용하여 HealthLake 데이터 스토어 쿼리 SQL.

목차

- Amazon Comprehend Medical API 작업을에 통합하는 방법의 예 HealthLake
- 검색 파라미터

Amazon Comprehend Medical이와 통합되는 방법 HealthLake

HealthLake 는 Amazon Comprehend Medical을 사용하여 `DocumentReference` 리소스 유형에서 찾은 데이터를 추론합니다. Amazon Comprehend Medical API 작업 `DetectEntities-V2`, `InferICD10-CM`및는 의학적 상태를 특성으로 `InferRxNorm` 감지합니다. 각 작업은 서로 다른 인사이트를 제공합니다.

> ⚠ 언어 지원
>
> Amazon Comprehend Medical API 작업은 영어 텍스트로 된 의료 개체만 감지합니다.

- DetectEntities-V2: 임상 텍스트에 다양한 의료 개체가 있는지 검사하고 개체 범주, 위치 및 신뢰도 점수와 같은 특정 정보를 반환합니다.
- InferICD10-CM: 환자 레코드의 의학적 상태를 개체로 감지하고, 세계 보건 기구()의 권한 부여에 따라 CDC의 국립 보건 통계 센터의 ICD-10-CM 지식 기반에 있는 정규화된 개념 식별자에 해당 개체를 연결합니다WHO.
- InferRxNorm: 환자 레코드에 나열된 개체로 약물을 감지하고 National Library of Medicine에서 RxNorm 데이터베이스의 정규화된 개념 식별자에 연결합니다.

각 API 작업에 지원되는 특성은 `SIGN`, 및 `SYMPTOM`입니다`DIAGNOSIS`. 특성이 감지되면 HealthLake 데이터 스토어의 다른 위치에 FHIR대한 규정 준수 확장으로 추가됩니다.

확장이 추가되는 위치입니다.

- `DocumentReference`: Amazon Comprehend Medical API 작업의 결과는 DocumentReference 리소스 유형 내에서 찾을 수 `extension` 있는 각 문서에 로 추가됩니다. 확장의 결과는 두 그룹으로 나뉩니다. 를 기반으로 결과에서 찾을 수 있습니다`URL`.
  - `http://healthlake.amazonaws.com/system-generated-resources/`
    - 이는에서 생성하거나에 추가한 리소스 유형입니다 HealthLake.
  - `http://healthlake.amazonaws.com/aws-cm/`

    - Amazon Comprehend Medical API 작업의 원시 출력이 HealthLake 데이터 스토어에 추가되는 위치입니다.
- `Linkage`:이 리소스 유형은 통합의 결과로 추가되거나 생성됩니다NLP. 특정에 대한 `GET` 요청은 연결된 리소스 목록을 `Linkage` 반환합니다. 가에 추가`Linkage`되었는지 확인하려면 추가된 `"tag": [{"display": "SYSTEM_GENERATED"}]` 키-값 페어를 HealthLake찾습니다. 연결 FHIR 사양에 대한 자세한 내용은 FHIR 설명서 인덱스의 리소스 유형: 연결을 참조하세요.
- FHIR Amazon Comprehend Medical API 작업의 결과로 생성된 리소스 유형입니다.
    - `Observation`: 특성이 `SIGN` 또는 일 때 Amazon Comprehend Medical API operations DetectEntities-V2 및 InferICD10-CM의 결과가 추가되었습니다`SYMPTOM`.
    - `Condition`: 특성이 일 때 Amazon Comprehend Medical API operations DetectEntities-V2 및 InferICD10-CM의 결과가 추가되었습니다`DIAGNOSIS`.
    - `MedicationStatement`: Amazon Comprehend Medical API 작업의 결과가 InferRxNorm 추가되었습니다.

FHIR REST API 작업과 통합

기본적으로 Amazon Comprehend Medical API 작업에서 감지한 특성은 `GET` 요청 시 반환되지 않습니다.

이러한 리소스 유형에 대한 통합 NLP 작업의 결과를 보려면 알려진를 지정해야 합니다`ID`.

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

DocumentReference 리소스 유형 외부의 통합 NLP 작업 결과는 지정된에 Amazon Comprehend Medical API 작업의 결과가 포함된 것으로 `ID` 알려진 `GET` 요청을 통해서만 사용할 수 있습니다.

Amazon Comprehend Medical API 작업을에 통합하는 방법의 예 HealthLake

예제 1: HealthLake 데이터 스토어에 수집된 환자 레코드

다음은 의료 전문가와의 환자 접촉을 기반으로 한 임상 기록의 예입니다.

⚠ 합성 데이터

이 예제의 텍스트는 합성 콘텐츠이며 개인 건강 정보()를 포함하지 않습니다PHI.

```
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)
```

참고로이 정보는 DocumentReference 리소스에서 base64 형식으로 인코딩됩니다. 이 문서가에 수집되고 HealthLake Amazon Comprehend Medical API 작업이 완료되면 리소스 DocumentReference 유형에 대한 GET요청으로 시작하여 결과를 볼 수 있습니다.

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

Amazon Comprehend Medical API 작업이 성공하면 다음과 extension 연결된 내에서 이러한 키-값 페어를 찾습니다. "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."
 }
```

다음 탭은 수집된 의료 기록이 리소스 유형에 따라 HealthLake 데이터 스토어에 보고되는 방법을 보여줍니다.

DocumentReference

에서 단일 DocumentReference 리소스 유형에 대한 결과를 보려면 특정 리소스id의가 제공되는 GET 요청을 합니다.

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

성공하면 200 HTTP 응답 코드와 다음 JSON 응답(명확성을 위해 잘림)이 표시됩니다.

다음은 http://healthlake.amazonaws.com/system-generated-resources/ 부분입니다. 새가 추가Linkage/*e366d29f-2c22-4c19-866e-09603937935a*되었음을 확인할 수 있습니다. HealthLake 가 특정 Observation 및 Condition 리소스 유형에 추론 기반 조사 결과를 추가한 위치를 확인할 수도 있습니다.

이러한 리소스 유형이 어떻게 수정되었는지 확인하려면 관련 탭을 선택합니다.


```
{
  "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

에서 단일 Linkage 리소스 유형에 대한 결과를 보려면 특정 리소스ID의가 제공되는 GET 요청을 하세요.

```
GET https://https://healthlake.your-region.amazonaws.com/
datastore/your-datastore-id/r4/eeb8005725ae22b35b4edbdc68cf2dfd/r4/
Linkage/e366d29f-2c22-4c19-866e-09603937935a
```

성공하면 200 HTTP 응답 코드와 다음과 같은 잘린 JSON 응답을 가져옵니다.

응답에는 item 요소가 포함됩니다. 여기서 키-값 페어는를 수정Condition하는 데 사용되는 특정 DocumentReference 항목을 "type": "source" 나타내며 "type": "alternate" 키-값 페어 아래에 Observations 나열됩니다.

또한 *meta* 요소 및 해당 키-값 페어 *"tag": [{"display": "SYSTEM_GENERATED"}]*가 표시되어 이러한 리소스가에 의해 생성되었음을 나타냅니다 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

에서 단일 Observation 리소스 유형에 대한 결과를 보려면 특정 리소스ID의가 제공되는 GET 요청을 합니다.

```
GET https://https://healthlake.your-region.amazonaws.com/
datastore/your-datastore-id/r4/eeb8005725ae22b35b4edbdc68cf2dfd/r4/
Observation/e366d29f-2c22-4c19-866e-09603937935a
```

Amazon Comprehend Medical API 작업의 결과는 code, meta및 요소로 수정됩니다modifierExtension.

code

유형의 요소입니다CodeableConcept. 자세한 내용은 FHIR 설명서 인덱스CodeableConcept의 섹션을 참조하세요.

HealthLake 는 다음 세 가지 키-값 페어를 추가합니다.

- "system": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/": 여기서는 특정 Amazon Comprehend Medical API 작업을 URL 나타냅니다. 이 경우 InferICD10CM입니다.
- "code": "A52.06": 여기서 A52.06는 Centers for Disease Control의 지식 기반에서 찾을 수 있는 개념을 식별하는 ICD-10-CM 코드입니다.
- "display": "Other syphilitic heart involvement": 여기서 "Other syphilitic heart involvement"는 onlogy의 ICD-10-CM 코드에 대한 긴 설명입니다.

다음 잘린 JSON 응답에는 code 요소만 포함됩니다.

```
"code": {
  "coding":
  [
    {
   "system": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/",
   "code": "A52.06",
   "display": "Other syphilitic heart involvement"
    }
  ],
  "text": "Other syphilitic heart involvement"
 }
```

할당된 ICD-10-CM 코드가 올바르다는 모델의 신뢰도를 이해하려면 modifierExtension 요소를 사용합니다.

meta

meta 요소에는 Amazon Comprehend Medical API 작업에 의해 추가된 세부 정보가 code 요소에 포함되어 있는지 여부를 나타내는 메타데이터가 포함되어 있습니다.

다음 잘린 JSON 응답에는 meta 요소만 포함됩니다.

```
"meta": {
  "lastUpdated": "2022-10-21T19:38:30.879Z",
  "tag": [{
   "display": "SYSTEM_GENERATED"
  }]
 }
```

modifierExtension

modifierExtension 요소에는 code 요소에 있는 할당된 코드의 신뢰도 수준에 대한 자세한 내용이 포함되어 있습니다. 또한 결과 및 관련 Linkage 리소스 유형을 생성하는 데 DocumentReference 사용된 원본에 대한 링크를 다시 제공하는 키-값 페어도 있습니다.

추가된 각 coding 요소에 대해 entity-score 및가에 entity-Concept-Score 추가되었습니다modifierExtension. 키-값 페어의 각 값에 대해 점수가 표시됩니다. 의 경우 entity-score이 점수는 Amazon Comprehend Medical이 탐지 정확도에 대해 갖는 신뢰도 수준입니다. 의 경우 entity-Concept-Score이 점수는 Amazon Comprehend Medical이 개체가 ICD-10-CM 개념에 정확하게 연결되어 있다는 확신 수준입니다.

다음 잘린 JSON 응답에는 modifierExtension 요소만 포함됩니다.

```
"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"
   }
  }
 ]
```

전체 JSON 응답

```
{
  "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

에서 단일 Condition 리소스 유형에 대한 결과를 보려면 특정 리소스ID의가 제공되는 GET 요청을 하세요.

```
GET https://https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/eeb8005725ae22b35b4edbdc68cf2dfd/r4/Condition/b06d343d-ddb8-4f36-82cb-853fcd434dfd
```

Amazon Comprehend Medical API 작업의 결과는 code, meta및 요소로 수정됩니다modifierExtension.

code

유형의 요소입니다CodeableConcept. 자세한 내용은 FHIR 설명서 인덱스CodeableConcept의 섹션을 참조하세요.

HealthLake 는 다음 세 가지 키-값 페어를 추가합니다.

- "system": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/": 여기서는 특정 Amazon Comprehend Medical API 작업을 URL 나타냅니다. 이 경우 InferICD10CM입니다.
- "code": "I70.0": 여기서 A52.06는 Centers for Disease Control의 지식 기반에서 찾을 수 있는 개념을 식별하는 ICD-10-CM 코드입니다.

- "display": "Atherosclerosis of aorta": 여기서 "Other syphilitic heart involvement"는 onlogy의 ICD-10-CM 코드에 대한 긴 설명입니다.

다음 잘린 JSON 응답에는 code 요소만 포함됩니다.

```
"code": {
  "coding":
  [
    {
   "system": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/",
   "code": "I70.0",
   "display": "Atherosclerosis of aorta"
    }
  ],
  "text": "Atherosclerosis of aorta"
 }
```

할당된 ICD-10-CM 코드가 올바르다는 모델의 신뢰도를 이해하려면 modifierExtension 요소를 사용합니다.

meta

meta 요소에는 Amazon Comprehend Medical API 작업에서 추가한 세부 정보가 code 요소에 포함되어 있는지 여부를 나타내는 메타데이터가 포함되어 있습니다.

다음 잘린 JSON 응답에는 meta 요소만 포함됩니다.

```
"meta": {
  "lastUpdated": "2022-10-21T19:38:30.877Z",
  "tag": [{
   "display": "SYSTEM_GENERATED"
  }]
 }
```

modifierExtension

modifierExtension 요소에는 code 요소에 있는 할당된 코드의 신뢰도 수준에 대한 자세한 내용이 포함되어 있습니다. 또한 결과 및 관련 Linkage 리소스 유형을 생성하는 데 DocumentReference 사용된 원본에 대한 링크를 다시 제공하는 키-값 페어도 있습니다.

추가된 각 coding 요소에 대해 entity-score 및가에 entity-Concept-Score 추가되었습니다modifierExtension. 키-값 페어의 각 값에 대해 점수가 표시됩니다. 의 경우 entity-score이 점수는 Amazon Comprehend Medical이 탐지 정확도에 대해 갖는 신뢰도 수준입니다. 의 경우 entity-Concept-Score이 점수는 Amazon Comprehend Medical이 개체가 ICD-10-CM 개념에 정확하게 연결되어 있다는 확신 수준입니다.

다음 잘린 JSON 응답에는 modifierExtension 요소만 포함됩니다.

```
"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"
   }
  }
 ]
```

전체 JSON 응답

```
{
  "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"
 }
```

예제 2: MedicationStatement 리소스 유형이 **DocumentReference** 포함된 A

다음은 의료 전문가와의 환자 접촉을 기반으로 한 임상 기록의 예입니다.

⚠ 합성 데이터

이 예제의 텍스트는 합성 콘텐츠이며 개인 건강 정보()를 포함하지 않습니다PHI.

```
Tom is not prescribed Advil
```

다음 탭은 수집된 의료 기록이 리소스 유형에 따라 HealthLake 데이터 스토어에 보고되는 방법을 보여 줍니다.

DocumentReference

에서 단일 DocumentReference 리소스 유형에 대한 결과를 보려면 특정 리소스ID의가 제공되는 GET 요청을 하세요.

```
GET https://https://healthlake.your-region.amazonaws.com/datastore/your-datastore-id/r4/eeb8005725ae22b35b4edbdc68cf2dfd/r4/DocumentReference/c549125d-a218-421f-b8bf-23614c5e796c
```

성공하면 200 HTTP 응답 코드와 다음과 같은 잘린 JSON 응답이 표시됩니다.

키-값 페어인는이 안에 있는 리소스 유형이 Amazon Comprehend Medical API 작업에 의해 추가extension되었음을 "url": "http://healthlake.amazonaws.com/system-generated-resources/"나타냅니다. 새 Linkage 리소스 유형과 여러 MedicationStatement 리소스를 볼 수 있습니다.

```
"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

에서 단일 Linkage 리소스 유형에 대한 결과를 보려면 특정 리소스ID의가 제공되는를 GET 요청합니다.

```
GET https://https://healthlake.your-region.amazonaws.com/
datastore/your-datastore-id/r4/eeb8005725ae22b35b4edbdc68cf2dfd/r4/
Linkage/394bb244-177b-4409-8657-26b20ed56dd7
```

성공하면 200 HTTP 응답 코드와 다음 JSON 응답이 표시됩니다.

응답에는 item 요소가 포함됩니다. 여기서 키-값 페어는 MedicationStatement 리소스 유형을 수정하는 데 사용되는 특정 DocumentReference 항목을 "type": "source" 나타냅니다.

meta 요소 및 해당 키-값 페어인 도 볼 수 있으며"tag": [{"display": "SYSTEM_GENERATED"}], 이는 이러한 리소스가에 의해 생성되었음을 나타냅니다 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

에서 단일 MedicationStatement 리소스 유형에 대한 결과를 보려면 특정 리소스ID의가 제공되는 GET 요청을 합니다.

```
GET https://https://healthlake.your-region.amazonaws.com/
datastore/your-datastore-id/r4/eeb8005725ae22b35b4edbdc68cf2dfd/r4/
MedicationStatement/9a89b0d3-6681-45ca-9926-27951edce5c7
```

MedicationStatement 리소스 유형은 Amazon Comprehend Medical InferRxNorm API 작업의 결과를 찾을 수 있는 곳입니다. 결과는 medicationCodeableConcept, meta및 요소로 수정됩니다modifierExtension.

medicationCodeableConcept

유형의 요소입니다CodeableConcept. 자세한 내용은 FHIR 설명서 인덱스CodeableConcept의 섹션을 참조하세요.

HealthLake 는 다음 세 가지 키-값 페어를 추가합니다.

- "system": ""http://healthlake.amazonaws.com/aws-cm/infer-rxnorm/: 여기서는 특정 Amazon Comprehend Medical API 작업을 URL 나타냅니다. 이 경우 InferRxNorm.
- "code": "731533": 여기서 731533는 Rx라고도 하는 RxNorm 개념 ID입니다CUI.
- "display": "ibuprofen 200 MG Oral Capsule [Advil]": ibuprofen 200 MG Oral Capsule [Advil]는 RxNorm 개념에 대한 설명입니다.

다음 잘린 JSON 응답에는 MedicationStatement 요소만 포함됩니다.

```
"medicationCodeableConcept": {
```

```
  "coding": [
   {
    "system": "http://healthlake.amazonaws.com/aws-cm/infer-rxnorm/",
    "code": "731533",
    "display": "ibuprofen 200 MG Oral Capsule [Advil]"
   }
  ]
 }
```

meta

meta 요소에는 Amazon Comprehend Medical API 작업에서 추가한 세부 정보가 code 요소에 포함되어 있는지 여부를 나타내는 메타데이터가 포함되어 있습니다.

다음 잘린 JSON 응답에는 meta 요소만 포함됩니다.

```
"meta": {
  "lastUpdated": "2022-10-24T20:05:02.800Z",
  "tag": [
   {
    "display": "SYSTEM_GENERATED"
   }
  ]
 }
```

modifierExtension

modifierExtension 요소에는 결과 및 관련 Linkage 리소스 유형을 생성하는 데 DocumentReference 사용된 원본에 대한 링크를 다시 제공하는 키-값 페어가 포함되어 있습니다.

```
"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"
    }
   }
```

```
]
```

검색 파라미터

다음 표에는 통합 의료에 대한 검색 가능한 속성이 나열되어 있습니다NLP.

검색 파라미터

검색 파라미터	에 대한 일치 항목을 찾습니다.
detectEntities-entity-category	AWS CM 확장 내의 DetectEntities 하위 확장명 내의 개체 범주
detectEntities-entity-text	AWS CM 확장 내의 DetectEntities 하위 확장명 내의 개체 텍스트
detectEntities-엔터티-유형	AWS CM 확장 내의 DetectEntities 하위 확장명 내의 개체 유형
detectEntities-entity-score	AWS CM 확장 내의 DetectEntities 하위 확장명 내 개체 점수
infer-icd10cm-entity-text	AWS CM 확장 내의 InferICD10CM 하위 확장명 내의 개체 텍스트
infer-icd10cm-entity-score	AWS CM 확장 내의 InferICD10CM 하위 확장명 내의 개체 점수
infer-icd10cm-entity-concept-code	AWS CM 확장 내의 InferICD10CM 하위 확장명 내의 개체 개념 코드
infer-icd10cm-entity-concept-description	AWS CM 확장 내의 InferICD10CM 하위 확장명 내의 개체 개념 설명
infer-icd10cm-entity-concept-score	AWS CM 확장 내의 InferICD10CM 하위 확장명 내의 개체 개념 점수
infer-rxnorm-entity-score	AWS CM 확장 내의 InferRxNorm 하위 확장명 내 개체 점수

검색 파라미터	에 대한 일치 항목을 찾습니다.
infer-rxnorm-entity-text	AWS CM 확장 내의 InferRxNorm 하위 확장명 내의 개체 텍스트
infer-rxnorm-entity-concept-code	AWS CM 확장 내의 InferRxNorm 하위 확장명 내의 개체 개념 코드
infer-rxnorm-entity-concept-설명	AWS CM 확장 내의 InferRxNorm 하위 확장명 내의 개체 개념 설명
infer-rxnorm-entity-concept-점수	AWS CM 확장 내의 InferRxNorm 하위 확장명 내의 개체 개념 점수

`EntityText` 및 `EntityCategory`가 동일한 개체의 일부인 기준과 일치시키기 위해는 특수 검색을 HealthLake 제공합니다. 다음 표에서는 내에서 지원되는 특수 검색 파라미터를 설명합니다 HealthLake.

검색 파라미터

검색 파라미터	반환된 일치 항목
detectEntities-entity-text-category	DetectEntities 하위 확장명에 entityText 및 둘 다와 일치하는 개체가 하나 이상 있는 경우entityCategory.
detectEntities-entity-type-score	DetectEntities 하위 확장명에 entityType 및 둘 다와 일치하는 개체가 하나 이상 있는 경우entityScore.
detectEntities-entity-text-score	DetectEntities 하위 확장명에 entityText 및 둘 다와 일치하는 개체가 하나 이상 있는 경우entityScore.
detectEntities-entity-text-type	DetectEntities 하위 확장명에 entityText 및 둘 다와 일치하는 개체가 하나 이상 있는 경우entityType.
detectEntities-entity-category-score	entityCategory 및 둘 다와 일치하는 개체가 하나 이상 있는 경우entityScore.

검색 파라미터	반환된 일치 항목
infer-icd10 cm-entity-text-concept코드	InferICD10CM 하위 확장에와 일치하는 개체가 하나 이상 있고 코드 entityText 와 일치하는 해당 개체 conceptCode 에 대해 하나 이상이 있는 경우.
infer-icd10cm-entity-text-concept-score	InferICD10CM 하위 확장에와 일치하는 개체가 하나 이상 entityText 있고 점수와 일치하는 conceptScore 개체가 하나 이상 있는 경우.
infer-icd10cm-entity-concept-description-concept-score	InferICD10CM 하위 확장의 개체 내에 개념 설명 및와 일치하는 개념이 하나 이상 있는 경우conceptScore.
infer-rxnorm-entity-text-개념-코드	InferRxNorm 하위 확장명에 코드와 일치하는 개체가 하나 이상 entityText 있고 해당 개체 conceptCode 에 대해 코드와 일치하는 개체가 하나 이상 있는 경우.
infer-rxnorm-entity-text-개념-점수	InferRxNorm 하위 확장명에와 일치하는 개체가 하나 이상 entityText 있고 점수와 일치하는 conceptScore 개체가 하나 이상 있는 경우.
infer-rxnorm-entity-concept-description-concept-score	InferRxNorm 하위 확장의 개체 내에 개념 설명 및와 일치하는 개념이 하나 이상 있는 경우conceptScore.

의 보안 AWS HealthLake

의 클라우드 보안 AWS 이 최우선 순위입니다. AWS 고객은 가장 보안에 민감한 조직의 요구 사항을 충족하도록 구축된 데이터 센터 및 네트워크 아키텍처의 이점을 누릴 수 있습니다.

보안은 AWS 와 사용자 간의 공동 책임입니다. 공동 책임 모델은 이 사항을 클라우드의 보안 및 클라우드 내 보안으로 설명합니다.

- 클라우드 보안 AWS 은 AWS 클라우드에서 AWS 서비스를 실행하는 인프라를 보호하는 역할을 합니다. AWS 또한는 안전하게 사용할 수 있는 서비스를 제공합니다. 타사 감사자는 AWS 규정 준수 프로그램 일환으로 보안의 효과를 정기적으로 테스트하고 확인합니다. 에 적용되는 규정 준수 프로그램에 대한 자세한 내용은 규정 준수 프로그램 AWS 제공 범위 내 서비스규정 준수 프로그램 HealthLake참조하세요.
- 클라우드의 보안 - 사용자의 책임은 사용하는 AWS 서비스에 따라 결정됩니다. 또한 귀하는 귀사의 데이터 민감도, 귀사의 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다.

이 설명서는를 사용할 때 공동 책임 모델을 적용하는 방법을 이해하는 데 도움이 됩니다 HealthLake. 다음 주제에서는 보안 및 규정 준수 목표를 충족 HealthLake 하도록를 구성하는 방법을 보여줍니다. 또한 HealthLake 리소스를 모니터링하고 보호하는 데 도움이 되는 다른 AWS 서비스를 사용하는 방법도 알아봅니다.

주제

- 의 데이터 보호 AWS HealthLake
- 에서에 REST 대한 암호화 AWS HealthLake
- 에 대한 전송 중 암호화 AWS HealthLake
- 에 대한 자격 증명 및 액세스 관리 AWS HealthLake
- AWS CloudTrail을 사용하여 AWS HealthLake API 호출 로깅
- 에 대한 규정 준수 검증 AWS HealthLake
- 의 복원력 AWS HealthLake
- AWS HealthLake의 인프라 보안
- AWS HealthLake의 보안 모범 사례

의 데이터 보호 AWS HealthLake

AWS 공동 책임 모델의 데이터 보호에 적용됩니다 AWS HealthLake. 이 모델에 설명된 대로 AWS 는 모든를 실행하는 글로벌 인프라를 보호할 책임이 있습니다 AWS 클라우드. 사용자는 인프라에서 호스팅되는 콘텐츠를 관리해야 합니다. 사용하는 AWS 서비스 의 보안 구성과 관리 태스크에 대한 책임도 사용자에게 있습니다. 데이터 개인 정보 보호에 대한 자세한 내용은 데이터 개인 정보 보호를 FAQ참조하세요. 유럽의 데이터 보호에 대한 자세한 내용은 AWS 보안 블로그의 AWS 공동 책임 모델 및 GDPR 블로그 게시물을 참조하세요.

데이터 보호를 위해 자격 증명을 보호하고 AWS 계정 AWS IAM Identity Center 또는 AWS Identity and Access Management ()를 사용하여 개별 사용자를 설정하는 것이 좋습니다IAM. 이렇게 하면 개별 사용자에게 자신의 직무를 충실히 이행하는 데 필요한 권한만 부여됩니다. 또한 다음과 같은 방법으로 데이터를 보호하는 것이 좋습니다.

- 각 계정에 다단계 인증(MFA)을 사용합니다.
- SSL/TLS를 사용하여 AWS 리소스와 통신합니다. TLS 1.2가 필요하며 TLS 1.3을 권장합니다.
- 를 사용하여 API 및 사용자 활동 로깅을 설정합니다 AWS CloudTrail. CloudTrail 추적을 사용하여 AWS 활동을 캡처하는 방법에 대한 자세한 내용은 AWS CloudTrail 사용 설명서의 CloudTrail 추적 작업을 참조하세요.
- AWS 암호화 솔루션과 내부의 모든 기본 보안 제어를 사용합니다 AWS 서비스.
- Amazon S3에 저장된 민감한 데이터를 검색하고 보호하는 데 도움이 되는 Amazon Macie와 같은 고급 관리형 보안 서비스를 사용하세요.
- 명령줄 인터페이스 또는를 AWS 통해 FIPS에 액세스할 때 140-3 검증 암호화 모듈이 필요한 경우 FIPS 엔드포인트를 API사용합니다. 사용 가능한 FIPS 엔드포인트에 대한 자세한 내용은 연방 정보 처리 표준(FIPS) 140-3을 참조하세요.

고객의 이메일 주소와 같은 기밀 정보나 중요한 정보는 태그나 이름 필드와 같은 자유 형식 텍스트 필드에 입력하지 않는 것이 좋습니다. 여기에는 콘솔, API AWS CLI또는를 사용하여 HealthLake 또는 다른 AWS 서비스 로 작업하는 경우가 포함됩니다 AWS SDKs. 이름에 사용되는 태그 또는 자유 형식 텍스트 필드에 입력하는 모든 데이터는 청구 또는 진단 로그에 사용될 수 있습니다. 외부 서버에 URL를 제공하는 경우 해당 서버에 대한 요청을 검증URL하기 위해에 자격 증명 정보를 포함하지 않는 것이 좋습니다.

에서에 REST 대한 암호화 AWS HealthLake

HealthLake 는 기본적으로 서비스 소유 AWS Key Management Service(AWS KMS) 키를 사용하여 저장 시 민감한 고객 데이터를 보호하기 위한 암호화를 제공합니다. 고객 관리형 KMS 키도 지원되며 데이터 스토어에서 파일을 가져오고 내보내는 데 필요합니다. 고객 관리형 KMS 키에 대한 자세한 내용은 Amazon Key Management Service를 참조하세요. 고객은 데이터 스토어를 생성할 때 AWS 소유 KMS 키 또는 고객 관리형 KMS 키를 선택할 수 있습니다. 데이터 스토어가 생성된 후에는 암호화 구성을 변경할 수 없습니다. 데이터 스토어가 AWS 소유 KMS 키를 사용하는 경우 로 표시되고 저장 시 암호화에 사용되는 특정 키 AWS_OWNED_KMS_KEY 가 표시되지 않습니다.

AWS 소유 KMS 키

HealthLake 는 기본적으로 이러한 키를 사용하여 개인 식별 정보 또는 개인 상태 정보(PHI) 저장 데이터와 같은 잠재적으로 민감한 정보를 자동으로 암호화합니다. AWS 소유 KMS 키는 계정에 저장되지 않습니다. 여러 AWS 계정에서 사용할 수 있도록를 AWS 소유하고 관리하는 KMS 키 모음의 일부입니다. AWS 서비스는 AWS 소유 KMS 키를 사용하여 데이터를 보호할 수 있습니다. AWS 소유 KMS 키를 보거나 관리하거나 사용하거나 그 사용을 감사할 수 없습니다. 하지만 사용자는 데이터를 암호화하는 키를 보호하기 위해 어떤 작업을 수행하거나 어떤 프로그램을 변경할 필요가 없습니다.

AWS 소유 KMS 키를 사용하는 경우 월별 요금 또는 사용 요금이 부과되지 않으며 계정의 AWS KMS 할당량에 포함되지 않습니다. 자세한 내용은 AWS 소유 키를 참조하세요.

고객 관리형 KMS 키

HealthLake 는 사용자가 생성, 소유 및 관리하는 대칭 고객 관리형 KMS 키의 사용을 지원하여 기존 AWS 소유 암호화를 통해 두 번째 암호화 계층을 추가합니다. 이 암호화 계층을 완전히 제어할 수 있으므로 다음과 같은 작업을 수행할 수 있습니다.

- 주요 정책, IAM 정책 및 권한 부여 설정 및 유지
- 키 암호화 자료 교체
- 키 정책 활성화 및 비활성화
- 태그 추가
- 키 별칭 만들기
- 삭제를 위한 스케줄 키

CloudTrail 를 사용하여 AWS KMS 가 사용자를 대신하여에 HealthLake 보내는 요청을 추적할 수도 있습니다. 추가 AWS KMS 요금이 적용됩니다. 자세한 내용은 고객 소유 키를 참조하세요.

고객 관리형 키 만들기

AWS 관리 콘솔 또는를 사용하여 대칭 고객 관리형 키를 생성할 수 있습니다 AWS KMS APIs.

AWS Key Management Service 개발자 안내서의 대칭 고객 관리형 키 생성 단계를 따릅니다.

키 정책에서는 고객 관리형 키에 대한 액세스를 제어합니다. 모든 고객 관리형 키에는 키를 사용할 수 있는 사람과 키를 사용하는 방법을 결정하는 문장이 포함된 정확히 하나의 키 정책이 있어야 합니다. 고객 관리형 키를 만들 때 키 정책을 지정할 수 있습니다. 자세한 내용은 AWS Key Management Service 개발자 안내서의 고객 관리형 키 액세스 관리를 참조하세요.

HealthLake 리소스와 함께 고객 관리형 키를 사용하려면 키 정책에서 kms:CreateGrant 작업을 허용해야 합니다. 이렇게 하면 지정된 키에 대한 액세스를 제어하는 고객 관리형 KMS 키에 권한이 추가되어 사용자에게 필요한 kms:grant 작업에 HealthLake 대한 액세스 권한이 부여됩니다. 자세한 내용은 권한 부여 사용을 참조하세요.

고객 관리형 KMS 키를 HealthLake 리소스와 함께 사용하려면 키 정책에서 다음 API 작업을 허용해야 합니다.

- kms: 권한 부여 작업에 대한 액세스를 허용하는 특정 고객 관리형 KMS 키에 권한 부여를 CreateGrant 추가합니다.
- kms: 키를 검증하는 데 필요한 고객 관리형 키 세부 정보를 DescribeKey 제공합니다. 이것은 모든 작업에 필요합니다.
- kms:모든 쓰기 작업에 대해 유휴 리소스를 암호화할 수 있는 액세스 권한을 GenerateDataKey 제공합니다.
- kms:Decrypt는 암호화된 리소스에 대한 읽기 또는 검색 작업에 대한 액세스를 제공합니다.

다음은 사용자가 해당 키로 암호화된 데이터 스토어를 생성하고 상호 작용할 수 AWS HealthLake 있도록 허용하는 정책 설명 예제입니다.

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

고객 관리형 KMS 키 사용에 필요한 IAM 권한

고객 관리형 KMS 키를 사용하여 AWS KMS 암호화가 활성화된 데이터 스토어를 생성할 때는 키 정책과 HealthLake 데이터 스토어를 생성하는 사용자 또는 역할에 대한 IAM 정책 모두에 필요한 권한이 있습니다.

kms:ViaService condition 키를 사용하여 KMS 키 사용을에서 시작된 요청으로만 제한할 수 있습니다 HealthLake.

키 정책에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 IAM 정책 활성화를 참조하세요.

리포지토리를 생성하는 IAM 사용자, IAM 역할 또는 AWS 계정에는 kms:CreateGrant,kms:GenerateDataKey및 kms:DescribeKey 권한과 필요한 HealthLake 권한이 있어야 합니다.

에서 권한 부여를 HealthLake 사용하는 방법 AWS KMS

HealthLake 에서는 고객 관리형 KMS 키를 사용하기 위한 권한 부여가 필요합니다. 고객 관리형 KMS 키로 암호화된 데이터 스토어를 생성하면는 CreateGrant 요청을 AWS에 전송하여 사용자를 대신하여 권한을 HealthLake 생성합니다KMS. 의 권한 부여AWSKMS는 고객 계정의 KMS 키에 대한 HealthLake 액세스 권한을 부여하는 데 사용됩니다.

가 사용자를 대신하여 HealthLake 생성하는 권한 부여는 취소하거나 사용 중지해서는 안 됩니다. 계정의 AWS KMS 키를 사용할 수 있는 HealthLake 권한을 부여하는 권한을 취소하거나 사용 중지하는 경우는이 데이터에 액세스하거나 데이터 스토어에 푸시된 새 FHIR 리소스를 암호화하거나 풀링될 때 복

호화할 수 HealthLake 없습니다. 권한 부여를 취소하거나 사용 중지하면 변경 HealthLake사항이 즉시 발생합니다. 액세스 권한을 취소하려면 권한 부여를 취소하는 대신 데이터 스토어를 삭제해야 합니다. 데이터 스토어가 삭제되면는 사용자를 대신하여 권한 부여를 HealthLake 중단합니다.

HealthLake에 대한 암호화 키 모니터링 대상

CloudTrail 를 사용하여 고객 관리형 KMS 키를 사용할 때 AWS KMS 가 사용자를 대신하여에 HealthLake 보내는 요청을 추적할 수 있습니다. 로그의 CloudTrail 로그 항목은 userAgent 필드에 healthlake.amazonaws.com 표시하여가 수행한 요청을 명확하게 구분합니다 HealthLake.

다음 예제는 고객 관리형 키로 암호화된 데이터에 액세스 DescribeKey 하기 위해에서 호출한 AWS KMS 작업을 모니터링 HealthLake 하기 위한 GenerateDataKey CreateGrant, , 복호화 및에 대한 CloudTrail 이벤트입니다.

다음은를 사용하여가 고객이 제공한 KMS 키 CreateGrant 에 HealthLake 액세스하도록 허용하는 방법을 보여줍니다.는이 KMS 키를 사용하여 유휴 상태의 모든 고객 데이터를 암호화 HealthLake 할 수 있습니다.

사용자는 자체 권한 부여를 생성할 필요가 없습니다.는 AWS에 CreateGrant 요청을 전송하여 사용자를 대신하여 권한 부여를 HealthLake 생성합니다KMS. 의 권한 부여 AWS KMS 는 고객 계정의 AWS KMS 키에 대한 HealthLake 액세스 권한을 부여하는 데 사용됩니다.

```
         {
"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"
```

```
}
```

다음 예제에서는를 사용하여 사용자가 데이터를 저장 GenerateDataKey 하기 전에 데이터를 암호화하는 데 필요한 권한을 갖도록 하는 방법을 보여줍니다.

```
         {
    "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"
}
```

다음 예제에서는가 복호화 작업을 HealthLake 호출하여 저장된 암호화된 데이터 키를 사용하여 암호화된 데이터에 액세스하는 방법을 보여줍니다.

```
         {
    "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"
}
```

다음 예제에서는가 DescribeKey 작업을 HealthLake 사용하여 AWS KMS 고객 소유 AWS KMS 키가 사용 가능한 상태인지 확인하고 작동하지 않는 경우 사용자가 문제를 해결하는 데 도움이 되는 방법을 보여줍니다.

```
            {
    "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"
}
```

자세히 알아보기

다음 리소스는 저장 데이터 암호화에 대한 자세한 정보를 제공합니다.

AWS Key Management Service 기본 개념에 대한 자세한 내용은 설명서를 참조하세요 AWS KMS .

AWS KMS 설명서의 보안 모범 사례에 대한 자세한 내용을 참조하세요.

에 대한 전송 중 암호화 AWS HealthLake

AWS HealthLake 는 TLS 1.2를 사용하여 퍼블릭 엔드포인트와 백엔드 서비스를 통해 전송 중인 데이터를 암호화합니다.

에 대한 자격 증명 및 액세스 관리 AWS HealthLake

AWS Identity and Access Management (IAM)는 관리자가 AWS 리소스에 대한 액세스를 안전하게 제어하는 데 도움이 AWS 서비스 되는 입니다. IAM 관리자는 HealthLake 리소스를 사용하도록 인증(로그인) 및 권한 부여(권한 부여)를 받을 수 있는 사용자를 제어합니다. IAM는 추가 비용 없이 사용할 수 AWS 서비스 있는 입니다.

주제

- 대상
- ID를 통한 인증
- 정책을 사용하여 액세스 관리
- 에서를 AWS HealthLake 사용하는 방법 IAM
- 에 대한 자격 증명 기반 정책 예제 AWS HealthLake
- AWS 에 대한 관리형 정책 AWS HealthLake
- AWS HealthLake 자격 증명 및 액세스 문제 해결

대상

사용 방법 AWS Identity and Access Management (IAM)은 수행하는 작업에 따라 다릅니다 HealthLake.

서비스 사용자 - HealthLake 서비스를 사용하여 작업을 수행하는 경우 관리자는 필요한 자격 증명과 권한을 제공합니다. 더 많은 HealthLake 기능을 사용하여 작업을 수행하면 추가 권한이 필요할 수 있습니다. 액세스 권한 관리 방법을 이해하면 관리자에게 올바른 권한을 요청하는 데 도움이 됩니다.

HealthLake의 기능에 액세스할 수 없는 경우 AWS HealthLake 자격 증명 및 액세스 문제 해결을 참조하세요.

서비스 관리자 - 회사에서 HealthLake 리소스를 책임지고 있는 경우에 대한 전체 액세스 권한이 있을 수 있습니다 HealthLake. 서비스 사용자가 액세스해야 하는 HealthLake 기능과 리소스를 결정하는 것은 사용자의 작업입니다. 그런 다음 IAM 관리자에게 요청을 제출하여 서비스 사용자의 권한을 변경합니다. 이 페이지의 정보를 검토하여 IAM의 기본 개념을 이해하십시오. 회사에서 IAM를 사용하는 방법에 대한 자세한 내용은 섹션을 HealthLake참조하세요에서를 AWS HealthLake 사용하는 방법 IAM.

IAM 관리자 - IAM 관리자인 경우에 대한 액세스를 관리하기 위한 정책을 작성하는 방법에 대한 세부 정보를 알고 싶을 수 있습니다 HealthLake. 에서 사용할 수 있는 자격 HealthLake 증명 기반 정책 예제를 보려면 섹션을 IAM참조하세요에 대한 자격 증명 기반 정책 예제 AWS HealthLake.

ID를 통한 인증

인증은 자격 증명 AWS 으로에 로그인하는 방법입니다. 로 AWS 계정 루트 사용자, IAM 사용자로 또는 IAM 역할을 수임하여 인증(로그인 AWS)되어야 합니다.

자격 증명 소스를 통해 제공된 자격 증명을 사용하여 페더레이션 자격 증명 AWS 으로에 로그인할 수 있습니다. AWS IAM Identity Center (IAM Identity Center) 사용자, 회사의 Single Sign-On 인증 및 Google 또는 Facebook 자격 증명은 페더레이션 자격 증명의 예입니다. 페더레이션형 ID로 로그인하는 경우 관리자가 이전에 IAM 역할을 사용하여 ID 페더레이션을 설정한 상태입니다. 페더레이션을 사용하여 AWS 에 액세스하면 간접적으로 역할을 수임하게 됩니다.

사용자 유형에 따라 AWS Management Console 또는 AWS 액세스 포털에 로그인할 수 있습니다. 에 로그인하는 방법에 대한 자세한 내용은 AWS 로그인 사용 설명서의에 로그인하는 방법을 AWS참조하세요. AWS 계정

AWS 프로그래밍 방식으로에 액세스하는 경우는 자격 증명을 사용하여 요청에 암호화 방식으로 서명할 수 있는 소프트웨어 개발 키트(SDK)와 명령줄 인터페이스(CLI)를 AWS 제공합니다. AWS 도구를 사용하지 않는 경우 직접 요청에 서명해야 합니다. 권장 방법을 사용하여 직접 요청에 서명하는 방법에 대한 자세한 내용은 IAM 사용 설명서의 AWS API 요청에 대한 서명 버전 4를 참조하세요.

사용하는 인증 방법에 상관없이 추가 보안 정보를 제공해야 할 수도 있습니다. 예를 들어,는 멀티 팩터 인증(MFA)을 사용하여 계정의 보안을 강화하도록 AWS 권장합니다. 자세한 내용은 사용 설명서의 다중 인증 및 사용 설명서AWS 의 다중 인증 IAM 섹션을 참조하세요IAM. AWS IAM Identity Center

AWS 계정 루트 사용자

를 생성할 때 계정의 모든 AWS 서비스 및 리소스에 대한 완전한 액세스 권한이 있는 하나의 로그인 자격 증명으로 AWS 계정시작합니다. 이 자격 증명을 테 AWS 계정 루트 사용자라고 하며 계정을 생성하

는 데 사용한 이메일 주소와 암호로 로그인하여 액세스합니다. 일상적인 작업에 루트 사용자를 사용하지 않을 것을 강력히 권장합니다. 루트 사용자 보안 인증 정보를 보호하고 루트 사용자만 수행할 수 있는 태스크를 수행하는 데 사용하십시오. 루트 사용자로 로그인해야 하는 태스크의 전체 목록은 IAM 사용자 안내서의 루트 사용자 보안 인증이 필요한 작업을 참조하세요.

페더레이션 자격 증명

가장 좋은 방법은 관리자 액세스가 필요한 사용자를 포함하여 사람이 자격 증명 공급자와의 연동을 사용하여 임시 자격 증명을 사용하여 AWS 서비스 에 액세스하도록 요구하는 것입니다.

페더레이션 자격 증명은 엔터프라이즈 사용자 디렉터리, 웹 자격 증명 공급자, AWS Directory Service, Identity Center 디렉터리 또는 자격 증명 소스를 통해 제공된 자격 증명을 사용하여 AWS 서비스 에 액세스하는 모든 사용자의 사용자입니다. 페더레이션 자격 증명에 액세스할 때 역할을 AWS 계정수임하고 역할은 임시 자격 증명을 제공합니다.

중앙 집중식 액세스 관리를 위해 AWS IAM Identity Center을(를) 사용하는 것이 좋습니다. IAM Identity Center에서 사용자 및 그룹을 생성하거나 모든 및 애플리케이션에서 사용할 수 있도록 자체 자격 증명 소스의 사용자 AWS 계정 및 그룹 집합에 연결하고 동기화할 수 있습니다. IAM Identity Center에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서의 IAM Identity Center란 무엇입니까?를 참조하세요.

IAM 사용자 및 그룹

IAM 사용자는 한 사람 또는 애플리케이션에 대한 특정 권한이 AWS 계정 있는 내 자격 증명입니다. 가능한 경우 암호 및 액세스 키와 같은 장기 보안 인증 정보가 있는 IAM 사용자를 생성하는 대신 임시 보안 인증 정보를 사용하는 것이 좋습니다. 그러나 IAM 사용자와 장기 보안 인증이 필요한 특정 사용 사례가 있는 경우 액세스 키를 교체하는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 장기 보안 인증이 필요한 사용 사례의 경우 정기적으로 액세스 키 교체를 참조하세요.

IAM 그룹은 IAM 사용자 컬렉션을 지정하는 자격 증명입니다. 사용자는 그룹으로 로그인할 수 없습니다. 그룹을 사용하여 여러 사용자의 권한을 한 번에 지정할 수 있습니다. 그룹을 사용하면 대규모 사용자 집합의 권한을 더 쉽게 관리할 수 있습니다. 예를 들어 라는 그룹을 지정IAMAdmins하고 해당 그룹에 IAM 리소스를 관리할 수 있는 권한을 부여할 수 있습니다.

사용자는 역할과 다릅니다. 사용자는 한 사람 또는 애플리케이션과 고유하게 연결되지만, 역할은 해당 역할이 필요한 사람이라면 누구나 수임할 수 있습니다. 사용자는 영구적인 장기 자격 증명을 가지고 있지만, 역할은 임시 자격 증명만 제공합니다. 자세한 내용은 IAM 사용 설명서의 IAM 사용자 사용 사례를 참조하세요.

IAM 역할

IAM 역할은 특정 권한이 AWS 계정 있는 내 자격 증명입니다. 이 역할은 IAM 사용자와 비슷하지만, 특정 개인과 연결되지 않습니다. 에서 IAM 역할을 일시적으로 수임하려면 사용자에서 역할(콘솔)로 전환할 AWS Management Console수 있습니다. IAM 또는 AWS API 작업을 호출하거나 사용자 지정를 AWS CLI 사용하여 역할을 수임할 수 있습니다URL. 역할 사용 방법에 대한 자세한 내용은 IAM 사용 설명서의 역할 수임 방법을 참조하세요.

임시 자격 증명이 있는 IAM 역할은 다음과 같은 상황에서 유용합니다.

- 페더레이션 사용자 액세스 - 페더레이션 ID에 권한을 부여하려면 역할을 생성하고 해당 역할의 권한을 정의합니다. 페더레이션 ID가 인증되면 역할이 연결되고 역할에 정의된 권한이 부여됩니다. 페더레이션 역할에 대한 자세한 내용은 IAM 사용 설명서의 타사 자격 증명 공급자(페더레이션)에 대한 역할 생성을 참조하세요. IAM Identity Center를 사용하는 경우 권한 세트를 구성합니다. 인증 후 자격 증명이 액세스할 수 있는 항목을 제어하기 위해 IAM Identity Center는 권한 세트를의 역할과 상호 연관시킵니다IAM. 권한 집합에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서의 권한 집합을 참조하세요.
- 임시 IAM 사용자 권한 - IAM 사용자 또는 역할은 IAM 역할을 수임하여 특정 작업에 대해 다른 권한을 일시적으로 수임할 수 있습니다.
- 크로스 계정 액세스 – IAM 역할을 사용하여 다른 계정의 사용자(신뢰할 수 있는 보안 주체)가 내 계정의 리소스에 액세스하도록 할 수 있습니다. 역할은 계정 간 액세스를 부여하는 기본적인 방법입니다. 그러나 일부 에서는 (역할을 프록시로 사용하는 대신) 정책을 리소스에 직접 연결할 AWS 서비스 수 있습니다. 교차 계정 액세스에 대한 역할과 리소스 기반 정책의 차이점을 알아보려면 IAM 사용 설명서의 에서 교차 계정 리소스 액세스를 IAM 참조하세요.
- 교차 서비스 액세스 - 일부는 다른에서 기능을 AWS 서비스 사용합니다 AWS 서비스. 예를 들어 서비스에서 호출할 때 해당 서비스가 Amazon에서 애플리케이션을 실행EC2하거나 Amazon S3에 객체를 저장하는 것이 일반적입니다. 서비스는 직접 호출하는 보안 주체의 권한을 사용하거나, 서비스 역할을 사용하거나, 또는 서비스 연결 역할을 사용하여 이 작업을 수행할 수 있습니다.
  - 전달 액세스 세션(FAS) - IAM 사용자 또는 역할을 사용하여에서 작업을 수행할 때 보안 주체로 AWS간주됩니다. 일부 서비스를 사용하는 경우 다른 서비스에서 다른 작업을 시작하는 작업을 수행할 수 있습니다.는를 호출하는 보안 주체의 권한을 다운스트림 서비스에 AWS 서비스 대한 요청과 AWS 서비스함께 FAS 사용합니다. FAS 요청은 서비스가 다른 AWS 서비스 또는 리소스와의 상호 작용이 필요한 요청을 수신하는 경우에만 수행됩니다. 이 경우, 두 작업을 모두 수행할 수 있는 권한이 있어야 합니다. FAS 요청 시 정책 세부 정보는 액세스 세션 전달을 참조하세요.
  - 서비스 역할 - 서비스 역할은 서비스가 사용자를 대신하여 작업을 수행하기 위해 수임하는 IAM 역할입니다. IAM 관리자는 IAM 내에서 서비스 역할을 생성, 수정, 삭제할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 에 권한을 위임할 역할 생성을 AWS 서비스 참조하세요.

- 서비스 연결 역할 - 서비스 연결 역할은에 연결된 서비스 역할의 한 유형입니다 AWS 서비스. 서비스는 사용자를 대신하여 작업을 수행하기 위해 역할을 수임할 수 있습니다. 서비스 연결 역할은에 표시 AWS 계정 되며 서비스가 소유합니다. IAM 관리자는 서비스 연결 역할의 권한을 볼 수 있지만 편집은 할 수 없습니다.

- Amazon에서 실행되는 애플리케이션 EC2 - IAM 역할을 사용하여 EC2 인스턴스에서 실행되고 AWS CLI 또는 AWS API 요청을 하는 애플리케이션의 임시 자격 증명을 관리할 수 있습니다. 이는 EC2 인스턴스 내에 액세스 키를 저장할 때 권장되는 방법입니다. EC2 인스턴스에 AWS 역할을 할당하고 모든 애플리케이션에서 사용할 수 있도록 하려면 인스턴스에 연결된 인스턴스 프로파일을 생성합니다. 인스턴스 프로파일에는 역할이 포함되어 있으며, EC2 인스턴스에서 실행되는 프로그램이 임시 자격 증명을 얻을 수 있도록 합니다. 자세한 내용은 IAM 사용 설명서EC2의 IAM 역할 사용을 참조하세요.

정책을 사용하여 액세스 관리

정책을 AWS 생성하고 자격 증명 또는 리소스에 연결하여 AWS 에서 액세스를 제어합니다. 정책은 자격 증명 또는 리소스와 연결된 AWS 경우 권한을 정의하는의 객체입니다.는 보안 주체(사용자, 루트 사용자 또는 역할 세션)가 요청할 때 이러한 정책을 AWS 평가합니다. 정책에서 권한은 요청이 허용되거나 거부되는 지를 결정합니다. 대부분의 정책은에 JSON 문서 AWS 로 저장됩니다. JSON 정책 문서의 구조 및 내용에 대한 자세한 내용은 IAM 사용 설명서의 JSON 정책 개요를 참조하세요.

관리자는 정책을 사용하여 AWS JSON 대상에 액세스할 수 있는 사용자를 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

기본적으로, 사용자 및 역할에는 어떠한 권한도 없습니다. 사용자에게 사용자가 필요한 리소스에서 작업을 수행할 권한을 부여하려면 IAM 관리자가 IAM 정책을 생성하면 됩니다. 그런 다음 관리자가 IAM 정책을 역할에 추가하고, 사용자가 역할을 수임할 수 있습니다.

IAM 정책은 작업을 실행하기 위한 방법과 상관없이 작업을 정의합니다. 예를 들어, `iam:GetRole` 작업을 허용하는 정책이 있다고 가정합니다. 해당 정책이 있는 사용자는 AWS Management Console, AWS CLI또는에서 역할 정보를 가져올 수 있습니다 AWS API.

ID 기반 정책

자격 증명 기반 정책은 IAM 사용자, 사용자 그룹 또는 역할과 같은 자격 증명에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 사용자 및 역할이 어떤 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 제어합니다. 자격 증명 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 고객 관리형 정책을 사용하여 사용자 지정 IAM 권한 정의를 참조하세요.

ID 기반 정책은 인라인 정책 또는 관리형 정책으로 한층 더 분류할 수 있습니다. 인라인 정책은 단일 사용자, 그룹 또는 역할에 직접 포함됩니다. 관리형 정책은의 여러 사용자, 그룹 및 역할에 연결할 수 있는 독립 실행형 정책입니다 AWS 계정. 관리형 정책에는 AWS 관리형 정책 및 고객 관리형 정책이 포함됩니다. 관리형 정책 또는 인라인 정책 중에서 선택하는 방법을 알아보려면 IAM 사용 설명서의 관리형 정책 및 인라인 정책 중에서 선택을 참조하세요.

리소스 기반 정책

리소스 기반 정책은 리소스에 연결하는 JSON 정책 문서입니다. 리소스 기반 정책의 예는 IAM 역할 신뢰 정책과 Amazon S3 버킷 정책입니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 정책이 연결된 리소스의 경우, 정책은 지정된 보안 주체가 해당 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 정의합니다. 리소스 기반 정책에서 보안 주체를 지정해야 합니다. 보안 주체에는 계정, 사용자, 역할, 페더레이션 사용자 또는이 포함될 수 있습니다 AWS 서비스.

리소스 기반 정책은 해당 서비스에 있는 인라인 정책입니다. 리소스 기반 정책IAM에서는의 AWS 관리형 정책을 사용할 수 없습니다.

액세스 제어 목록(ACLs)

액세스 제어 목록(ACLs)은 리소스에 액세스할 수 있는 권한이 있는 보안 주체(계정 멤버, 사용자 또는 역할)를 제어합니다. ACLs는 JSON 정책 문서 형식을 사용하지 않지만 리소스 기반 정책과 유사합니다.

Amazon S3 AWS WAF및 AmazonVPC은를 지원하는 서비스의 예입니다ACLs. 에 대한 자세한 내용은 Amazon Simple Storage Service 개발자 안내서의 액세스 제어 목록(ACL) 개요를 ACLs참조하세요.

기타 정책 타입

AWS 는 덜 일반적인 추가 정책 유형을 지원합니다. 이러한 정책 타입은 더 일반적인 정책 유형에 따라 사용자에게 부여되는 최대 권한을 설정할 수 있습니다.

- 권한 경계 - 권한 경계는 자격 증명 기반 정책이 IAM 개체(IAM 사용자 또는 역할)에 부여할 수 있는 최대 권한을 설정하는 고급 기능입니다. 개체에 대한 권한 경계를 설정할 수 있습니다. 그 결과로 얻는 권한은 엔티티의 ID 기반 정책과 그 권한 경계의 교집합입니다. `Principal` 필드에서 사용자나 역할을 지정하는 리소스 기반 정책은 권한 경계를 통해 제한되지 않습니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 권한 경계에 대한 자세한 정보는 IAM 사용 설명서의 IAM 엔터티에 대한 권한 경계를 참조하세요.
- 서비스 제어 정책(SCPs) -의 조직 또는 조직 단위(OU)에 대한 최대 권한을 지정하는 JSON 정책 SCPs입니다 AWS Organizations. AWS Organizations 는 비즈니스가 소유 AWS 계정 한 여러를 그

룹화하고 중앙에서 관리하기 위한 서비스입니다. 조직의 모든 기능을 활성화하면 서비스 제어 정책(SCPs)을 모든 계정에 적용할 수 있습니다. 는 각각을 포함하여 멤버 계정의 엔터티에 대한 권한을 SCP 제한합니다 AWS 계정 루트 사용자. 조직 및에 대한 자세한 내용은 AWS Organizations 사용 설명서의 서비스 제어 정책을 SCPs참조하세요.

- 리소스 제어 정책(RCPs) - 소유한 각 리소스에 연결된 JSON 정책을 업데이트하지 않고 계정의 리소스에 사용할 수 있는 최대 권한을 설정하는 데 사용할 수 있는 IAM 정책RCPs입니다. 는 멤버 계정의 리소스에 대한 권한을 RCP 제한하며 조직에 속하는지 여부에 AWS 계정 루트 사용자관계없이를 포함한 자격 증명에 대한 유효 권한에 영향을 미칠 수 있습니다. 를 AWS 서비스 지원하는 목록을 RCPs포함하여 조직 및에 대한 자세한 내용은 AWS Organizations 사용 설명서의 리소스 제어 정책(RCPs)을 RCPs참조하세요.
- 세션 정책 – 세션 정책은 역할 또는 페더레이션 사용자에 대해 임시 세션을 프로그래밍 방식으로 생성할 때 파라미터로 전달하는 고급 정책입니다. 결과적으로 얻는 세션의 권한은 사용자 또는 역할의 ID 기반 정책의 교차와 세션 정책입니다. 또한 권한을 리소스 기반 정책에서 가져올 수도 있습니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 자세한 정보는 IAM 사용 설명서의 세션 정책을 참조하세요.

여러 정책 유형

여러 정책 유형이 요청에 적용되는 경우, 결과 권한은 이해하기가 더 복잡합니다. 가 여러 정책 유형이 관련될 때 요청을 허용할지 여부를 AWS 결정하는 방법을 알아보려면 IAM 사용 설명서의 정책 평가 로직을 참조하세요.

에서를 AWS HealthLake 사용하는 방법 IAM

IAM를 사용하여에 대한 액세스를 관리하기 전에 사용할 수 있는 IAM 기능에 대해 HealthLake알아봅니다 HealthLake.

IAM에서 사용할 수 있는 기능 AWS HealthLake

IAM 기능	HealthLake 지원
ID 기반 정책	예
리소스 기반 정책	아니요
정책 작업	예
정책 리소스	예

IAM 기능	HealthLake 지원
정책 조건 키	예
ACLs	아니요
ABAC (정책의 태그)	예
임시 보안 인증	예
보안 주체 권한	예
서비스 역할	예
서비스 연결 역할	아니요

HealthLake 및 기타 AWS 서비스가 대부분의 IAM 기능을 사용하는 방법을 개괄적으로 알아보려면 IAM 사용 설명서의 AWS 에서 작업하는 서비스를 IAM 참조하세요.

에 대한 자격 증명 기반 정책 AWS HealthLake

ID 기반 정책 지원: 예

자격 증명 기반 정책은 IAM 사용자, 사용자 그룹 또는 역할과 같은 자격 증명에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 사용자 및 역할이 어떤 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 제어합니다. 자격 증명 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 고객 관리형 정책을 사용하여 사용자 지정 IAM 권한 정의를 참조하세요.

IAM 자격 증명 기반 정책을 사용하면 허용되거나 거부되는 작업과 리소스뿐 아니라 작업이 허용되거나 거부되는 조건을 지정할 수 있습니다. ID 기반 정책에서는 보안 주체가 연결된 사용자 또는 역할에 적용되므로 보안 주체를 지정할 수 없습니다. JSON 정책에서 사용할 수 있는 모든 요소에 대해 알아보려면 IAM 사용 설명서의 IAM JSON 정책 요소 참조를 참조하세요.

에 대한 자격 증명 기반 정책 예제 AWS HealthLake

HealthLake 자격 증명 기반 정책의 예를 보려면 섹션을 참조하세요에 대한 자격 증명 기반 정책 예제 AWS HealthLake.

내 리소스 기반 정책 AWS HealthLake

리소스 기반 정책 지원: 아니요

리소스 기반 정책은 리소스에 연결하는 JSON 정책 문서입니다. 리소스 기반 정책의 예는 IAM 역할 신뢰 정책과 Amazon S3 버킷 정책입니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 정책이 연결된 리소스의 경우, 정책은 지정된 보안 주체가 해당 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 정의합니다. 리소스 기반 정책에서 보안 주체를 지정해야 합니다. 보안 주체에는 계정, 사용자, 역할, 페더레이션 사용자 또는이 포함될 수 있습니다 AWS 서비스.

크로스 계정 액세스를 활성화하려는 경우 전체 계정이나 다른 계정의 IAM 엔터티를 리소스 기반 정책의 보안 주체로 지정할 수 있습니다. 리소스 기반 정책에 교차 계정 보안 주체를 추가하는 것은 트러스트 관계 설정의 절반밖에 되지 않는다는 것을 유념합니다. 보안 주체와 리소스가 다른 경우 신뢰할 수 AWS 계정있는 계정의 IAM 관리자는 보안 주체 개체(사용자 또는 역할)에게 리소스에 액세스할 수 있는 권한도 부여해야 합니다. 엔터티에 ID 기반 정책을 연결하여 권한을 부여합니다. 하지만 리소스 기반 정책이 동일 계정의 보안 주체에 액세스를 부여하는 경우, 추가 자격 증명 기반 정책이 필요하지 않습니다. 자세한 내용은 IAM 사용 설명서의 에서 교차 계정 리소스 액세스를 IAM 참조하세요.

에 대한 정책 작업 AWS HealthLake

정책 작업 지원: 예

관리자는 정책을 사용하여 AWS JSON 대상에 액세스할 수 있는 사용자를 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

JSON 정책의 `Action` 요소는 정책에서 액세스를 허용하거나 거부하는 데 사용할 수 있는 작업을 설명합니다. 정책 작업은 일반적으로 연결된 AWS API 작업과 이름이 동일합니다. 일치하는 API 작업이 없는 권한 전용 작업과 같은 몇 가지 예외가 있습니다. 정책에서 여러 작업이 필요한 몇 가지 작업도 있습니다. 이러한 추가 작업을 일컬어 종속 작업이라고 합니다.

연결된 작업을 수행할 수 있는 권한을 부여하기 위한 정책에 작업을 포함합니다.

HealthLake 작업 목록을 보려면 서비스 승인 참조의에서 정의한 작업을 AWS HealthLake 참조하세요.

의 정책 작업은 작업 앞에 다음 접두사를 HealthLake 사용합니다.

```
healthlake
```

단일 문에서 여러 작업을 지정하려면 각 작업을 쉼표로 구분합니다.

```
"Action": [
      "healthlake:action1",
      "healthlake:action2"
         ]
```

HealthLake 자격 증명 기반 정책의 예를 보려면 섹션을 참조하세요에 대한 자격 증명 기반 정책 예제 AWS HealthLake.

에 대한 정책 리소스 AWS HealthLake

정책 리소스 지원: 예

관리자는 정책을 사용하여 AWS JSON 대상에 액세스할 수 있는 사용자를 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

`Resource` JSON 정책 요소는 작업이 적용되는 객체를 지정합니다. 문에는 `Resource`또는 `NotResource`요소가 반드시 추가되어야 합니다. 가장 좋은 방법은 Amazon 리소스 이름(ARN)을 사용하여 리소스를 지정하는 것입니다. 리소스 수준 권한이라고 하는 특정 리소스 유형을 지원하는 작업에 대해 이를 수행할 수 있습니다.

작업 나열과 같이 리소스 수준 권한을 지원하지 않는 작업의 경우, 와일드카드(*)를 사용하여 해당 문이 모든 리소스에 적용됨을 나타냅니다.

```
"Resource": "*"
```

HealthLake 리소스 유형 및 해당의 목록을 보려면 서비스 승인 참조의에서 정의한 리소스를 AWS HealthLake ARNs참조하세요. ARN 각 리소스의를 지정할 수 있는 작업을 알아보려면에서 정의한 작업을 AWS HealthLake 참조하세요.

HealthLake 자격 증명 기반 정책의 예를 보려면 섹션을 참조하세요에 대한 자격 증명 기반 정책 예제 AWS HealthLake.

에 대한 정책 조건 키 AWS HealthLake

서비스별 정책 조건 키 지원: 예

관리자는 정책을 사용하여 AWS JSON 대상에 액세스할 수 있는 사용자를 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

`Condition` 요소(또는 `Condition` 블록)를 사용하면 문이 발효되는 조건을 지정할 수 있습니다. `Condition` 요소는 옵션입니다. 같거나 작음과 같은 조건 연산자를 사용하여 정책의 조건을 요청의 값과 일치시키는 조건식을 생성할 수 있습니다.

한 문에서 여러 `Condition` 요소를 지정하거나 단일 `Condition` 요소에서 여러 키를 지정하는 경우, AWS 는 논리적 `AND` 작업을 사용하여 평가합니다. 단일 조건 키에 여러 값을 지정하는 경우는 논리적 `OR` 작업을 사용하여 조건을 AWS 평가합니다. 문의 권한을 부여하기 전에 모든 조건을 충족해야 합니다.

조건을 지정할 때 자리 표시자 변수를 사용할 수도 있습니다. 예를 들어, IAM 사용자에게 IAM 사용자 이름으로 태그가 지정된 경우에만 리소스에 액세스할 수 있는 권한을 부여할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 IAM 정책 요소: 변수 및 태그를 참조하세요.

AWS 는 전역 조건 키와 서비스별 조건 키를 지원합니다. 모든 AWS 전역 조건 키를 보려면 IAM 사용 설명서의 AWS 전역 조건 컨텍스트 키를 참조하세요.

HealthLake 조건 키 목록을 보려면 서비스 승인 참조의에 대한 조건 키를 AWS HealthLake 참조하세요. 조건 키를 사용할 수 있는 작업 및 리소스를 알아보려면에서 정의한 작업을 AWS HealthLake 참조하세요.

HealthLake 자격 증명 기반 정책의 예를 보려면 섹션을 참조하세요에 대한 자격 증명 기반 정책 예제 AWS HealthLake.

의 액세스 제어 목록(ACLs) AWS HealthLake

지원: ACLs 아니요

액세스 제어 목록(ACLs)은 리소스에 액세스할 수 있는 권한이 있는 보안 주체(계정 멤버, 사용자 또는 역할)를 제어합니다. ACLs는 JSON 정책 문서 형식을 사용하지 않지만 리소스 기반 정책과 유사합니다.

를 사용한 속성 기반 액세스 제어(ABAC) AWS HealthLake

지원ABAC(정책의 태그): 예

속성 기반 액세스 제어(ABAC)는 속성을 기반으로 권한을 정의하는 권한 부여 전략입니다. 에서는 AWS이러한 속성을 태그라고 합니다. IAM 엔터티(사용자 또는 역할) 및 많은 AWS 리소스에 태그를 연결할 수 있습니다. 엔터티 및 리소스에 태그를 지정하는 것은의 첫 번째 단계입니다ABAC. 그런 다음

보안 주체의 태그가 액세스하려는 리소스의 태그와 일치할 때 작업을 허용하는 ABAC 정책을 설계합니다.

ABAC는 빠르게 성장하는 환경에서 유용하며 정책 관리가 번거로워지는 상황에 도움이 됩니다.

태그에 근거하여 액세스를 제어하려면 `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` 또는 `aws:TagKeys` 조건 키를 사용하여 정책의 조건 요소에 태그 정보를 제공합니다.

서비스가 모든 리소스 유형에 대해 세 가지 조건 키를 모두 지원하는 경우, 값은 서비스에 대해 예입니다. 서비스가 일부 리소스 유형에 대해서만 세 가지 조건 키를 모두 지원하는 경우, 값은 부분적입니다.

에 대한 자세한 내용은 IAM 사용 설명서의 ABAC 권한 부여로 권한 정의를 ABAC참조하세요. 설정 단계가 포함된 자습서를 보려면 IAM 사용 설명서의 속성 기반 액세스 제어 사용(ABAC)을 ABAC참조하세요.

에서 임시 자격 증명 사용 AWS HealthLake

임시 자격 증명 지원: 예

임시 자격 증명을 사용하여 로그인할 때 작동하지 AWS 서비스 않는 경우도 있습니다. 임시 자격 증명으로 AWS 서비스 작업하는를 비롯한 자세한 내용은 IAM 사용 설명서의 AWS 서비스 로 작업하는 IAM를 참조하세요.

사용자 이름과 암호를 제외한 방법을 AWS Management Console 사용하여에 로그인하는 경우 임시 자격 증명을 사용합니다. 예를 들어 회사의 Single Sign-On(SSO) 링크를 AWS 사용하여에 액세스하면 해당 프로세스가 임시 자격 증명을 자동으로 생성합니다. 또한 콘솔에 사용자로 로그인한 다음 역할을 전환할 때 임시 자격 증명을 자동으로 생성합니다. 역할 전환에 대한 자세한 내용은 IAM 사용 설명서의 사용자에서 IAM 역할로 전환(콘솔)을 참조하세요.

AWS CLI 또는를 사용하여 임시 자격 증명을 수동으로 생성할 수 있습니다 AWS API. 그런 다음 이러한 임시 자격 증명을 사용하여 장기 액세스 키를 사용하는 대신 동적으로 임시 자격 증명을 생성하는 access AWS. AWS recommends에 액세스할 수 있습니다. 자세한 내용은 의 임시 보안 자격 증명을 IAM참조하세요.

에 대한 교차 서비스 보안 주체 권한 AWS HealthLake

전달 액세스 세션 지원(FAS): 예

IAM 사용자 또는 역할을 사용하여에서 작업을 수행하는 경우 보안 주체로 AWS간주됩니다. 일부 서비스를 사용하는 경우 다른 서비스에서 다른 작업을 시작하는 작업을 수행할 수 있습니다.는를 호출하는 보안 주체의 권한을 다운스트림 서비스에 AWS 서비스 대한 요청과 AWS 서비스함께 FAS 사용합니

다. FAS 요청은 서비스가 다른 AWS 서비스 또는 리소스와의 상호 작용이 필요한 요청을 수신하는 경우에만 수행됩니다. 이 경우, 두 작업을 모두 수행할 수 있는 권한이 있어야 합니다. FAS 요청 시 정책 세부 정보는 액세스 세션 전달을 참조하세요.

AWS HealthLake의 서비스 역할

서비스 역할 지원: 예

서비스 역할은 서비스가 사용자를 대신하여 작업을 수행하기 위해 수임하는 IAM 역할입니다. IAM 관리자는 IAM 내에서 서비스 역할을 생성, 수정, 삭제할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 에 권한을 위임할 역할 생성을 AWS 서비스 참조하세요.

에 대한 전체 액세스에 필요한 서비스 역할 및 인라인 정책에 대한 자세한 내용은 섹션을 AWS HealthLake참조하세요사용을 시작할 권한 설정 AWS HealthLake.

> **Warning**
>
> 서비스 역할에 대한 권한을 변경하면 HealthLake 기능이 중단될 수 있습니다. 가 관련 지침을 HealthLake 제공하는 경우에만 서비스 역할을 편집합니다.

에 대한 서비스 연결 역할 AWS HealthLake

서비스 링크 역할 지원: 아니요

서비스 연결 역할은에 연결된 서비스 역할의 한 유형입니다 AWS 서비스. 서비스는 사용자를 대신하여 작업을 수행하기 위해 역할을 수임할 수 있습니다. 서비스 연결 역할은에 표시 AWS 계정 되며 서비스가 소유합니다. IAM 관리자는 서비스 연결 역할의 권한을 볼 수 있지만 편집은 할 수 없습니다.

서비스 연결 역할 생성 또는 관리에 대한 자세한 내용은 IAM으로 작업하는AWS 서비스 단원을 참조하세요. 서비스 연결 역할 열에서 `Yes`이(가) 포함된 서비스를 테이블에서 찾습니다. 해당 서비스에 대한 서비스 연결 역할 설명서를 보려면 예 링크를 선택합니다.

에 대한 자격 증명 기반 정책 예제 AWS HealthLake

기본적으로 사용자 및 역할에는 HealthLake 리소스를 생성하거나 수정할 수 있는 권한이 없습니다. 또한 AWS Management Console, AWS Command Line Interface (AWS CLI) 또는를 사용하여 작업을 수행할 수 없습니다 AWS API. 사용자에게 사용자가 필요한 리소스에서 작업을 수행할 권한을 부여하려면 IAM 관리자가 IAM 정책을 생성하면 됩니다. 그런 다음 관리자가 IAM 정책을 역할에 추가하고, 사용자가 역할을 수임할 수 있습니다.

이러한 예제 정책 문서를 사용하여 IAM 자격 증명 기반 JSON 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 IAM 정책 생성(콘솔)을 참조하세요.

각 리소스 유형에 대한의 형식을 포함하여 HealthLake에서 정의한 작업 및 리소스 유형에 ARNs 대한 자세한 내용은 서비스 승인 참조의 에 대한 작업, 리소스 및 조건 키를 AWS HealthLake 참조하세요.

주제

- 정책 모범 사례
- AWS HealthLake 콘솔 사용
- 에서 AWS HealthLake 데이터 스토어에 액세스 Amazon Athena
- 사용자가 자체 권한을 볼 수 있도록 허용

정책 모범 사례

자격 증명 기반 정책은 계정에서 HealthLake 리소스를 생성, 액세스 또는 삭제할 수 있는지 여부를 결정합니다. 이 작업으로 인해 AWS 계정에 비용이 발생할 수 있습니다. ID 기반 정책을 생성하거나 편집할 때는 다음 지침과 권장 사항을 따릅니다.

- AWS 관리형 정책을 시작하고 최소 권한으로 전환 - 사용자 및 워크로드에 권한 부여를 시작하려면 많은 일반적인 사용 사례에 대한 권한을 부여하는 AWS 관리형 정책을 사용합니다. 에서 사용할 수 있습니다 AWS 계정. 사용 사례에 맞는 AWS 고객 관리형 정책을 정의하여 권한을 추가로 줄이는 것이 좋습니다. 자세한 정보는 IAM 사용 설명서의 AWS 관리형 정책 또는 직무에 관한AWS 관리형 정책을 참조하세요.
- 최소 권한 적용 – IAM 정책을 사용하여 권한을 설정하는 경우 태스크를 수행하는 데 필요한 권한만 부여합니다. 이렇게 하려면 최소 권한으로 알려진 특정 조건에서 특정 리소스에 대해 수행할 수 있는 작업을 정의합니다. IAM을 사용하여 권한을 적용하는 방법에 대한 자세한 정보는 IAM 사용 설명서에 있는 IAM의 정책 및 권한을 참조하세요.
- IAM 정책의 조건을 사용하여 액세스 추가 제한 – 정책에 조건을 추가하여 작업 및 리소스에 대한 액세스를 제한할 수 있습니다. 예를 들어 정책 조건을 작성하여를 사용하여 모든 요청을 전송하도록 지정할 수 있습니다SSL. AWS 서비스와 같은 특정를 통해 사용되는 경우 조건을 사용하여 서비스 작업에 대한 액세스 권한을 부여할 수도 있습니다 AWS CloudFormation. 자세한 내용은 IAM 사용 설명서의 IAM JSON 정책 요소: 조건을 참조하세요.
- IAM Access Analyzer를 사용하여 IAM 정책을 검증하여 안전하고 기능적인 권한을 보장합니다. IAM Access Analyzer는 정책이 정책 언어(JSON) 및 IAM 모범 사례를 준수하도록 새 정책 및 기존 IAM 정책을 검증합니다. IAM Access Analyzer는 안전하고 기능적인 정책을 작성하는 데 도움이 되는

100개 이상의 정책 확인 및 실행 가능한 권장 사항을 제공합니다. 자세한 내용은 IAM 사용 설명서의 IAM Access Analyzer를 사용한 정책 검증을 참조하세요.

- 다중 인증 필요(MFA) -에서 IAM 사용자 또는 루트 사용자가 필요한 시나리오가 있는 경우 추가 보안을 MFA 위해를 AWS 계정켭니다. API 작업을 호출할 MFA 때를 요구하려면 정책에 MFA 조건을 추가합니다. 자세한 내용은 IAM 사용 설명서의 를 사용한 보안 API 액세스를 MFA 참조하세요.

IAM의 모범 사례에 대한 자세한 내용은 IAM 사용 설명서의 IAM의 보안 모범 사례를 참조하세요.

AWS HealthLake 콘솔 사용

AWS HealthLake 콘솔에 액세스하려면 최소 권한 세트가 있어야 합니다. 이러한 권한을 통해의 HealthLake 리소스에 대한 세부 정보를 나열하고 볼 수 있어야 합니다 AWS 계정. 최소 필수 권한보다 더 제한적인 ID 기반 정책을 생성하는 경우, 콘솔이 해당 정책에 연결된 엔티티(사용자 또는 역할)에 대해 의도대로 작동하지 않습니다.

AWS CLI 또는 에만 호출하는 사용자에게 최소 콘솔 권한을 허용할 필요는 없습니다 AWS API. 대신 수행하려는 API 작업과 일치하는 작업에만 액세스하도록 허용합니다.

에 대한 전체 액세스 권한을 얻으려면 IAM 사용자 또는 역할에 `AmazonHealthLakeFullAccess` 및 정책을 HealthLake연결합니다`AWSLakeFormationDataAdmin`. 또한 서비스 역할인 HealthLake 인라인 정책을 연결해야 합니다. 서비스 역할은 서비스가 사용자를 대신하여 작업을 수행하기 위해 수임하는 IAM 역할입니다. IAM 관리자는 IAM 내에서 서비스 역할을 생성, 수정, 삭제할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 에 권한을 위임할 역할 생성을 참조하세요 AWS 서비스. 필요한 서비스 역할을 생성하는 인라인 정책에 대한 자세한 내용은 섹션을 참조하세요사용을 시작할 권한 설정 AWS HealthLake. 또한 AWS Lake Formation 콘솔 또는를 사용하여 HealthLake Data AWS Lake Formation Lake 관리자가 되도록 관리자를 CLI 할당해야 합니다. 자세한 내용은 사용을 시작할 권한 설정 AWS HealthLake 단원을 참조하십시오.

에서 AWS HealthLake 데이터 스토어에 액세스 Amazon Athena

사용자 및 역할에의 HealthLake 데이터 스토어에 대한 액세스 권한을 제공하려면 역할 또는 사용자에게 `AmazonAthenaFullAccess` 및 IAM를 Amazon Athena연결합니다`AmazonS3FullAccess`. `Select` 및 `Describe` 권한은에서 관리하는 테이블에도 필요합니다 AWS Lake Formation. AWS Lake Formation 콘솔의 AWS Lake Formation 관리자가 또는를 통해 AWS Lake Formation 테이블 권한을 부여합니다CLI. 자세한 내용은 사용을 시작할 권한 설정 AWS HealthLake 단원을 참조하세요.

사용자가 자체 권한을 볼 수 있도록 허용

이 예제는 IAM 사용자가 자신의 사용자 자격 증명에 연결된 인라인 및 관리형 정책을 볼 수 있도록 허용하는 정책을 생성하는 방법을 보여 줍니다. 이 정책에는 콘솔에서 또는 AWS CLI 또는를 사용하여 프로그래밍 방식으로이 작업을 완료할 수 있는 권한이 포함되어 있습니다 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 에 대한 관리형 정책 AWS HealthLake

AWS 관리형 정책은에서 생성 및 관리하는 독립 실행형 정책입니다 AWS. AWS 관리형 정책은 사용자, 그룹 및 역할에 권한 할당을 시작할 수 있도록 많은 일반적인 사용 사례에 대한 권한을 제공하도록 설계되었습니다.

AWS 관리형 정책은 모든 AWS 고객이 사용할 수 있으므로 특정 사용 사례에 대해 최소 권한 권한을 부여하지 않을 수 있습니다. 사용 사례에 고유한 고객 관리형 정책을 정의하여 권한을 줄이는 것이 좋습니다.

AWS 관리형 정책에 정의된 권한은 변경할 수 없습니다. 가 AWS 관리형 정책에 정의된 권한을 AWS 업데이트하면 정책이 연결된 모든 보안 주체 자격 증명(사용자, 그룹 및 역할)에 영향을 미칩니다. AWS 는 새 AWS 서비스 가 시작되거나 기존 서비스에 새 API 작업을 사용할 수 있게 되면 AWS 관리형 정책을 업데이트할 가능성이 높습니다.

자세한 내용은 IAM사용 설명서의 AWS 관리형 정책을 참조하세요.

AWS 관리형 정책: AmazonHealthLakeFullAccess

`AmazonHealthLakeFullAccess` 정책은에 대한 전체 액세스를 제공합니다 HealthLake. 이 정책을 사용자 또는 역할에 연결하면 사용자는 HealthLake 를 사용하여에서 데이터에 액세스, 쿼리, 가져오기 및 내보낼 수 있습니다 HealthLake. 에서 많은 일반적인 작업을 수행하려면 사용자 또는 역할에 정책을 추가 HealthLake해야 합니다. 자세한 내용은 사용을 시작할 권한 설정 AWS HealthLake 및 HealthLake 작업 및 권한을 참조하세요.

`AmazonHealthLakeFullAccess` 정책을 IAM 자격 증명에 연결할 수 있습니다.

이 정책은 사용자 및 역할이를 사용하여 쿼리, 검색, 가져오기 및 내보내기를 수행할 수 있는 *`administrative and contributor`* 권한을 부여하며 HealthLake,가 이러한 권한이 있는 사용자 및 역할을 대신하여 작업을 HealthLake 수행할 수 있도록 합니다.

권한 세부 정보

이 정책에는 다음 문이 포함됩니다.

```
{
 "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 관리형 정책: AmazonHealthLakeReadOnlyAccess

`AmazonHealthLakeReadOnlyAccess` 정책은 다른 AWS 서비스의 및 관련 리소스에 대한 읽기 전용 액세스 HealthLake 및 권한을 부여합니다. HealthLake 데이터 스토어를 쿼리하고 볼 수 있는 기능을 부여하려는 사용자에게이 정책을 적용하되, 데이터 스토어를 생성하거나 변경할 수 있는 기능은 적용하지 않습니다.

`AmazonHealthLakeReadOnlyAccess` 정책을 IAM 자격 증명에 연결할 수 있습니다.

이 정책은 사용자와 역할이 쿼리할 수 있는 *`read-only`* 권한을 부여합니다 HealthLake.

권한 세부 정보

이 정책에는 다음 문이 포함됩니다.

```
{
```

```
    "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 작업 및 권한

다음 표에는의 일반적인 작업 HealthLake 과 이를 수행하는 데 필요한 권한이 나열되어 있습니다.

HealthLake 작업	필수 권한
에서 데이터 스토어 생성 HealthLake	`AmazonHealthLakeFullAccess` ,`AmazonLakeFormationDataAdmin` 인라인 정책 및에서 관리하는 AWS Lake Formation 관리자 권한 AWS Lake Formation
에서 데이터 스토어 삭제 HealthLake	`AmazonHealthLakeFullAccess` , `AmazonLakeFormationDataAdmin` , 인라인 정책 및에서 관리하는 AWS Lake Formation 관리자 권한 AWS Lake Formation
에서 데이터 스토어 나열, 검색 또는 쿼리 HealthLake	`AmazonHealthLakeReadOnlyAccess`

HealthLake 작업	필수 권한
를 사용하여 데이터 스토어 쿼리 Amazon Athena	`AmazonAthenaFullAccess` 에서 관리하는 테이블에 대한 `AmazonS3FullAccess` , AWS Lake Formation `Select` 및 `Describe` 권한 AWS Lake Formation
에서 데이터 가져오기 HealthLake	가져오기 작업에 대한 권한 설정을 참조하세요.
에서 데이터 내보내기 HealthLake	데이터 스토어에서 파일 내보내기(AWS SDKs)을 참조하세요.

HealthLake AWS 관리형 정책에 대한 업데이트

이 서비스가 이러한 변경 사항을 추적하기 시작한 시점부터 HealthLake 에 대한 AWS 관리형 정책 업데이트에 대한 세부 정보를 봅니다. 이 페이지의 변경 사항에 대한 자동 알림을 받으려면 HealthLake 문서 기록 페이지에서 RSS 피드를 구독하세요.

변경 사항	설명	날짜
AmazonHealthLakeFullAccess	`AmazonHealthLakeFullAccess` 에 대한 전체 액세스를 허용하는 데 필요한 정책입니다 HealthLake.	2022년 11월 14일
AmazonHealthLakeReadOnlyAccess	`AmazonHealthLakeReadOnlyAccess` 에 대한 읽기 전용 액세스에 필요한 정책입니다 HealthLake.	2022년 11월 14일
HealthLake 변경 사항 추적 시작	HealthLake 는 AWS 관리형 정책에 대한 변경 사항 추적을 시작했습니다.	2022년 11월 14일

AWS HealthLake 자격 증명 및 액세스 문제 해결

다음 정보를 사용하여 HealthLake 및 작업 시 발생할 수 있는 일반적인 문제를 진단하고 수정할 수 있습니다IAM.

주제

- 에서 작업을 수행할 권한이 없음 AWS HealthLake
- iam을 수행할 권한이 없습니다.PassRole
- 내 AWS 계정 외부의 사람이 내 AWS HealthLake 리소스에 액세스하도록 허용하고 싶습니다.

에서 작업을 수행할 권한이 없음 AWS HealthLake

에서 작업을 수행할 권한이 없다는 AWS Management Console 메시지가 표시되면 관리자에게 문의하여 지원을 받아야 합니다. 관리자는 사용자 이름과 비밀번호를 제공한 사람입니다.

다음 예제 오류는 `mateojackson` IAM 사용자가 콘솔을 사용하여 가상 *`my-example-widget`* 리소스에 대한 세부 정보를 보려고 하지만 가상 `healthlake:`*`GetWidget`* 권한이 없는 경우에 발생합니다.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 healthlake:GetWidget on resource: my-example-widget
```

이 경우 Mateo는 *`my-example-widget`* 작업을 사용하여 `healthlake:`*`GetWidget`* 리소스에 액세스하도록 허용하는 정책을 업데이트하라고 관리자에게 요청합니다.

iam을 수행할 권한이 없습니다.PassRole

`iam:PassRole` 작업을 수행할 수 있는 권한이 없다는 오류가 수신되면 HealthLake에 역할을 전달할 수 있도록 정책을 업데이트해야 합니다.

일부는 새 서비스 역할 또는 서비스 연결 역할을 생성하는 대신 기존 역할을 해당 서비스에 전달할 수 있도록 AWS 서비스 허용합니다. 이렇게 하려면 사용자가 서비스에 역할을 전달할 수 있는 권한을 가지고 있어야 합니다.

다음 예제 오류는 `marymajor`라는 IAM 사용자가 콘솔을 사용하여 HealthLake에서 작업을 수행하려고 하는 경우에 발생합니다. 하지만 작업을 수행하려면 서비스 역할이 부여한 권한이 서비스에 있어야 합니다. Mary는 서비스에 역할을 전달할 수 있는 권한을 가지고 있지 않습니다.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

이 경우, Mary가 `iam:PassRole` 작업을 수행할 수 있도록 Mary의 정책을 업데이트해야 합니다.

도움이 필요한 경우 AWS 관리자에게 문의하십시오. 관리자는 로그인 자격 증명을 제공한 사람입니다.

내 AWS 계정 외부의 사람이 내 AWS HealthLake 리소스에 액세스하도록 허용하고 싶습니다.

다른 계정의 사용자 또는 조직 외부의 사람이 리소스에 액세스할 때 사용할 수 있는 역할을 생성할 수 있습니다. 역할을 수임할 신뢰할 수 있는 사람을 지정할 수 있습니다. 리소스 기반 정책 또는 액세스 제어 목록(ACLs)을 지원하는 서비스의 경우 이러한 정책을 사용하여 리소스에 대한 액세스 권한을 부여할 수 있습니다.

자세히 알아보려면 다음을 참조하세요.

- 가 이러한 기능을 HealthLake 지원하는지 여부를 알아보려면 섹션을 참조하세요에서를 AWS HealthLake 사용하는 방법 IAM.
- 소유 AWS 계정 한의 리소스에 대한 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 소유 AWS 계정 한 다른의 IAM 사용자에게 액세스 권한 제공을 참조하세요.
- 리소스에 대한 액세스 권한을 타사에 제공하는 방법을 알아보려면 IAM 사용 설명서의 타사 AWS 계정 소유에 대한 액세스 권한 제공을 AWS 계정참조하세요.
- ID 페더레이션을 통해 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 외부에서 인증된 사용자에게 액세스 권한 제공(ID 페더레이션)을 참조하세요.
- 교차 계정 액세스를 위한 역할 및 리소스 기반 정책 사용의 차이점을 알아보려면 IAM 사용 설명서의 에서 교차 계정 리소스 액세스를 IAM 참조하세요.

AWS CloudTrail을 사용하여 AWS HealthLake API 호출 로깅

AWS HealthLake 는 사용자 AWS CloudTrail, 역할 또는 AWS 서비스 in HealthLake. CloudTrail Capture에서에 대한 모든 API 호출을 이벤트 HealthLake 로 캡처하는 서비스에 통합되어 있습니다. 캡처된 호출에는 HealthLake 콘솔의 호출과 HealthLake API 작업에 대한 코드 호출이 포함됩니다. 추적을 생성하는 경우에 CloudTrail 대한 이벤트를 포함하여 Amazon S3 버킷으로 이벤트를 지속적으로 전송할 수 있습니다 HealthLake. 추적을 구성하지 않은 경우에도 CloudTrail 콘솔의 이벤트 기록에서 최신 이벤트를 볼 수 있습니다. 에서 수집한 정보를 사용하여에 수행된 요청 CloudTrail, 요청이 수행된 HealthLakeIP 주소, 요청을 수행한 사람, 요청이 수행된 시간 및 추가 세부 정보를 확인할 수 있습니다.

자세한 내용은 AWS CloudTrail 사용 설명서를 CloudTrail참조하세요.

AWS HealthLake 의 정보 CloudTrail

CloudTrail 는 AWS 계정을 생성할 때 계정에서 활성화됩니다. 에서 활동이 발생하면 HealthLake해당 활동이 CloudTrail 이벤트 기록의 다른 AWS 서비스 이벤트와 함께 이벤트에 기록됩니다. AWS 계정에서 최신 이벤트를 확인, 검색 및 다운로드할 수 있습니다. 자세한 내용은 이벤트 기록을 사용하여 CloudTrail 이벤트 보기를 참조하세요.

에 대한 이벤트를 포함하여 AWS 계정의 이벤트에 대한 지속적인 기록을 위해 추적을 HealthLake생성합니다. 추적 CloudTrail 을 사용하면가 Amazon S3 버킷에 로그 파일을 전송할 수 있습니다. 콘솔에서 추적을 생성하면 기본적으로 모든 AWS 지역에 추적이 적용됩니다. 추적은 AWS 파티션의 모든 리전에서 이벤트를 로깅하고 지정한 Amazon S3 버킷으로 로그 파일을 전송합니다. 또한 CloudTrail 로그에서 수집된 이벤트 데이터를 추가로 분석하고 이에 따라 조치를 취하도록 다른 AWS 서비스를 구성할 수 있습니다. 자세한 내용은 다음 자료를 참조하세요.

- 트레일 생성 개요
- CloudTrail 지원되는 서비스 및 통합
- 에 대한 Amazon SNS 알림 구성 CloudTrail
- 여러 리전에서 CloudTrail 로그 파일 수신 및 여러 계정에서 로그 파일 수신 CloudTrail

모든 HealthLake 작업은에서 로깅 CloudTrail 되며 FHIR REST를 사용하여 수행되는 작업에 대한 HealthLake API 참조 및이 개발자 안내서에 문서화되어 있습니다API. 예를 들어 다음 작업에 대한 호출은 CloudTrail 로그 파일에 항목을 생성합니다.

- `DescribeFHIRImportJob`
- `DescribeFHIRExportJob`
- `StartFHIRImportJob`
- `ListFHIRImportJobs`
- `StartFHIRExportJob`
- `ListFHIRExportJobs`
- `CreateFHIRDatastore`
- `ListFHIRDatastores`
- `DeleteFHIRDatastore`
- `DescribeFHIRDatastore`
- `UpdateResource`
- `CreateResource`

- DeleteResource
- ReadResource
- GetCapabilities
- SearchWithGet
- SearchWithPost

모든 이벤트 또는 로그 항목에는 요청을 생성했던 사용자에 대한 정보가 포함됩니다. 자격 증명을 이용하면 다음을 쉽게 판단할 수 있습니다.

- 요청이 루트 또는 AWS Identity and Access Management (IAM) 사용자 자격 증명으로 이루어졌는지 여부입니다.
- 역할 또는 페더레이션 사용자에 대한 임시 보안 인증을 사용하여 요청이 생성되었는지 여부.
- 요청이 다른 AWS 서비스에 의해 이루어졌는지 여부.

자세한 내용은 CloudTrail userIdentity 요소를 참조하세요.

AWS HealthLake 로그 파일 항목 이해

추적은 사용자가 지정한 Amazon S3 버킷으로 이벤트를 전송할 수 있도록 하는 구성입니다. CloudTrail 로그 파일에는 하나 이상의 로그 항목이 포함됩니다. 이벤트는 모든 소스의 단일 요청을 나타내며 요청된 작업, 작업 날짜 및 시간, 요청 파라미터 등에 대한 정보를 포함합니다. CloudTrail 로그 파일은 퍼블릭 API 호출의 정렬된 스택 추적이 아니므로 특정 순서로 표시되지 않습니다.

다음 예제에서는 CreateFHIRDatastore 작업을 보여주는 CloudTrail 로그 항목을 보여줍니다.

```
{
    "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"
}
```

에 대한 규정 준수 검증 AWS HealthLake

타사 감사자는 여러 규정 준수 프로그램의 AWS HealthLake 일환으로의 보안 및 AWS 규정 준수를 평가합니다. HealthLake 여기에는가 포함됩니다HIPAA.

AWS 서비스 가 특정 규정 준수 프로그램의 범위 내에 있는지 알아보려면 규정 준수 프로그램 AWS 서비스 범위규정 준수 프로그램 범위 섹션을 참조하고 관심 있는 규정 준수 프로그램을 선택합니다. 일반 정보는 AWS 규정 준수 프로그램 참조하세요.

를 사용하여 타사 감사 보고서를 다운로드할 수 있습니다 AWS Artifact. 자세한 내용은 에서 보고서 다운로드 AWS Artifact에서 .

사용 시 규정 준수 책임은 데이터의 민감도, 회사의 규정 준수 목표 및 관련 법률과 규정에 따라 AWS 서비스 결정됩니다.는 규정 준수를 지원하기 위해 다음 리소스를 AWS 제공합니다.

- 보안 규정 준수 및 거버넌스 - 이러한 솔루션 구현 가이드에서는 아키텍처 고려 사항을 설명하고 보안 및 규정 준수 기능을 배포하는 단계를 제공합니다.
- Amazon Web Services의 HIPAA 보안 및 규정 준수를 위한 설계 -이 백서에서는 기업이 AWS 를 사용하여 HIPAA적격 애플리케이션을 생성하는 방법을 설명합니다.

> **Note**
>
> 모두 HIPAA 자격이 AWS 서비스 있는 것은 아닙니다. 자세한 내용은 HIPAA 적격 서비스 참조를 참조하세요.

- AWS 규정 준수 리소스 -이 워크북 및 가이드 모음은 업계 및 위치에 적용될 수 있습니다.
- AWS 고객 규정 준수 가이드 - 규정 준수의 관점에서 공동 책임 모델을 이해합니다. 이 가이드는 여러 프레임워크(미국 국립표준기술연구소(), 결제카드 산업보안표준위원회(NIST), 국제표준화기구(ISO) 포함)에서 보안 및 보안 제어에 대한 지침을 AWS 서비스 매핑하기 위한 모범 사례를 요약PCI 합니다.
- AWS Config 개발자 안내서의 규칙을 사용하여 리소스 평가 -이 AWS Config 서비스는 리소스 구성이 내부 관행, 업계 지침 및 규정을 얼마나 잘 준수하는지 평가합니다.
- AWS Security Hub - 이를 AWS 서비스 통해 내 보안 상태를 포괄적으로 볼 수 있습니다 AWS. Security Hub는 보안 제어를 사용하여 AWS 리소스를 평가하고 보안 업계 표준 및 모범 사례에 대한 규정 준수를 확인합니다. 지원되는 서비스 및 제어 목록은 Security Hub 제어 참조를 참조하세요.
- Amazon GuardDuty - 환경의 의심스럽거나 악의적인 활동을 모니터링하여 사용자 AWS 계정, 워크로드DSS, 컨테이너 및 데이터에 대한 잠재적 위협을 AWS 서비스 탐지합니다.는 특정 규정 준수 프

레임워크에서 요구하는 침입 탐지 요구 사항을 충족하여 PCI와 같은 다양한 규정 준수 요구 사항을 해결하는 데 도움이 될 GuardDuty 수 있습니다.

- AWS Audit Manager - 이를 AWS 서비스 통해 AWS 사용량을 지속적으로 감사하여 규정 및 업계 표준의 위험과 규정 준수를 관리하는 방법을 간소화할 수 있습니다.

의 복원력 AWS HealthLake

AWS 글로벌 인프라는 AWS 리전 및 가용 영역을 기반으로 구축됩니다. AWS 리전은 지연 시간이 짧고 처리량이 높으며 중복성이 높은 네트워킹과 연결된 물리적으로 분리되고 격리된 여러 가용 영역을 제공합니다. 가용 영역을 사용하면 중단 없이 영역 간에 자동으로 장애 극복 조치가 이루어지는 애플리케이션 및 데이터베이스를 설계하고 운영할 수 있습니다. 가용 영역은 기존의 단일 또는 다중 데이터 센터 인프라보다 가용성, 내결함성, 확장성이 뛰어납니다.

AWS 리전 및 가용 영역에 대한 자세한 내용은 AWS 글로벌 인프라를 참조하세요.

AWS 글로벌 인프라 외에도는 데이터 복원력 및 백업 요구 사항을 지원하는 데 도움이 되는 여러 기능을 HealthLake 제공합니다.

AWS HealthLake의 인프라 보안

관리형 서비스인는 Amazon Web Services: 보안 프로세스 개요 백서에 설명된 AWS 글로벌 네트워크 보안 절차로 AWS HealthLake 보호됩니다. https://d0.awsstatic.com/whitepapers/Security/AWS_Security_Whitepaper.pdf

AWS 게시된 API 호출을 사용하여 네트워크를 HealthLake 통해에 액세스합니다. 클라이언트는 전송 계층 보안(TLS) 1.0 이상을 지원해야 합니다. TLS 1.2 이상을 사용하는 것이 좋습니다. 또한 클라이언트는 Ephemeral Diffie-Hellman(PFS) 또는 Elliptic Curve Ephemeral Diffie-Hellman()과 같은 완벽한 순방향 보안(DHE)을 갖춘 암호 제품군을 지원해야 합니다ECDHE. Java 7 이상의 최신 시스템은 대부분 이러한 모드를 지원합니다.

또한 요청은 액세스 키 ID 및 IAM 주체와 관련된 보안 액세스 키를 사용하여 서명해야 합니다. 또는 AWS Security Token Service(AWS STS)를 사용하여 임시 자격 증명을 생성하여 요청에 서명할 수 있습니다.

AWS HealthLake의 보안 모범 사례

AWS HealthLake 는 자체 보안 정책을 개발하고 구현할 때 고려해야 할 여러 보안 기능을 제공합니다. 다음 모범 사례는 일반적인 지침이며 완벽한 보안 솔루션을 나타내지는 않습니다. 이러한 모범 사례는 환경에 적절하지 않거나 충분하지 않을 수 있으므로 참고용으로만 사용해 주십시오.

- 최소 권한 액세스 구현.
- 가능하면 Customer-Managed-Keys(CMKs)를 사용하여 데이터를 암호화합니다. 에 대한 자세한 내용은 Amazon Key Management ServiceCMKs를 참조하세요.
- 데이터 스토어PII에서 PHI 또는를 쿼리할 GET 때에서 검색이 POST아닌에서 검색을 사용합니다.
- 민감하고 중요한 감사 함수에 대한 액세스를 제한합니다.
- 업데이트 또는 대량 가져오기를 통해 리소스를 생성할 때 데이터 스토어 및 작업의 이름PII, 표시되는 필드 또는 논리적 FHIR ID()를 포함하여 PHI 또는를 사용하지 APIs마십시오LID.
- 생성, 읽기, 업데이트, 삭제 또는 검색 요청을 전송할 때 HTTP 헤더PHI에서를 사용하지 마십시오.
- AWS HealthLake 사용을 AWS CloudTrail 감사하고 예기치 않은 활동이 없는지 확인하려면를 활성화합니다.
- Amazon S3 버킷을 안전하게 사용하기 위한 모범 사례를 검토합니다. 자세한 내용은 Amazon S3 사용 설명서의 보안 모범 사례를 참조하세요.

AWS HealthLake 엔드포인트 및 할당량

다음 섹션에는 할당 AWS HealthLake 량 및 엔드포인트에 대한 정보가 포함되어 있습니다. 조정 가능한 할당량의 경우 Service Quotas 콘솔을 사용하여 할당량 증가를 요청할 수 있습니다. 자세한 내용은 Service Quotas 사용 설명서의 할당량 증가 요청을 참조하세요.

서비스 엔드포인트

이 표에는 지정된 리전에서 사용 가능한 HealthLake 서비스 엔드포인트가 나와 있습니다.

리전 이름	지역	엔드포인트	프로토콜
미국 동부(오하이오)	us-east-2	healthlake.us-east-2.amazonaws.com healthlake-fips.us-east-2.amazonaws.com	HTTPS HTTPS
미국 동부(버지니아 북부)	us-east-1	healthlake.us-east-1.amazonaws.com healthlake-fips.us-east-1.amazonaws.com	HTTPS HTTPS
미국 서부(오레곤)	us-west-2	healthlake.us-west-2.amazonaws.com healthlake-fips.us-west-2.amazonaws.com	HTTPS HTTPS
아시아 태평양(뭄바이)	ap-south-1	healthlake.ap-south-1.amazonaws.com	HTTPS
아시아 태평양(시드니)	ap-southeast-2	healthlake.ap-southeast-2.amazonaws.com	HTTPS
유럽(런던)	eu-west-2	healthlake.eu-west-2.amazonaws.com	HTTPS

에 대한 서비스 할당량 HealthLake

다음은의 기본 할당량입니다 HealthLake.

명칭	기본값	조정 가능	설명
음성 메시지의 문자 수.	각 지원되는 리전: 10,000개	아니요	DocumentReference 리소스 유형(POST/PUT 요청) 내의 개별 의료 기록의 최대 문자 수입니다.
동시 S tartFHIRImport작업 수	지원되는 각 리전: 1	아니요	최대 동시 S tartFHIRImport작업입니다.
concurrentStartFHIRExportJob 작업 수	지원되는 각 리전: 1	아니요	최대 동시 S tartFHIRExport작업입니다.
계정당 데이터 저장소 수	지원되는 각 지역: 10개	예	계정당 기본 활성 데이터 저장소 최대 수.
S tartFHIRImport작업의 파일 수	지원되는 각 리전: 10,000개	아니요	S tartFHIRImport작업의 최대 파일 수입니다.
번들당 리소스 수	지원되는 각 리전: 160개	아니요	번들 요청에 허용되는 리소스 최대 수.
계정당 번들 요청 속도	각 지원되는 리전: 20개	예	계정당 초당 수행할 수 있는 POST 번들 요청의 최대 수입니다.
데이터 저장소당 번들 요청 속도	지원되는 각 리전: 10	예	데이터 스토어당 초당 수행할 수 있는 POST 번들

명칭	기본값	조정 가능	설명
			요청의 최대 수입니다. 2023년 8월 21일 이전에 생성된 데이터 저장소는 초당 요청 1개로 제한됩니다.
계정DELETE당를 사용한 C ancelFHIR Export작업 요청 속도	지원되는 각 리전: 1	아니요	계정당 분당 수행할 수 DELETE 있는를 사용하는 최대 C ancelFHIRExport 작업 요청 수입니다.
계정당 C reateFHIRDatastore 요청 속도	지원되는 각 리전: 1	아니요	계정당 분당 수행할 수 있는 최대 C reateFHIR Datastore 요청 수입니다.
계정당 DELETE 요청 속도	지원되는 각 리전: 2,000	예	계정당 초당 수행할 수 있는 최대 DELETE 요청 수입니다.
데이터 스토어당 DELETE 요청 속도	지원되는 각 리전: 1,000	예	데이터 스토어당 초당 수행할 수 있는 최대 DELETE 요청 수입니다. 2023년 8월 21일 이전에 생성된 데이터 저장소는 초당 요청 1개로 제한됩니다.
계정당 D eleteFHIRDatastore 요청 속도	지원되는 각 리전: 1	아니요	계정당 분당 수행할 수 있는 최대 D eleteFHIR Datastore 요청 수입니다.

명칭	기본값	조정 가능	설명
계정당 D escribeFHIRDatastore 요청 속도	지원되는 각 지역: 10개	아니요	계정당 초당 수행할 수 있는 최대 D escribeFHIRDatastore 요청 수입니다.
계정당 D escribeFHIRExport작업 요청 속도	지원되는 각 지역: 10개	아니요	계정당 초당 수행할 수 있는 최대 D escribeFHIRExport작업 요청 수입니다.
계정GET당를 사용하는 D escribeFHIRExport작업 요청 속도	지원되는 각 지역: 10개	아니요	계정당 초당 수행할 수 GET 있는를 사용하는 최대 D escribeFHIRExport작업 요청 수입니다.
계정당 D escribeFHIRImport작업 요청 속도	지원되는 각 지역: 10개	아니요	계정당 초당 수행할 수 있는 최대 D escribeFHIRImport작업 요청 수입니다.
계정당 검색 요청 속도	지원되는 각 지역: 10개	아니요	계정당 초당 수행할 수 있는 Discovery 요청 최대 수입니다.
계정당 GET 요청 속도	지원되는 각 리전: 6,000개	예	계정당 초당 수행할 수 있는 최대 GET 요청 수입니다.

명칭	기본값	조정 가능	설명
데이터 스토어당 GET 요청 속도	지원되는 각 리전: 3,000	예	데이터 스토어당 초당 수행할 수 있는 최대 GET 요청 수입니다. 2023년 8월 21일 이전에 생성된 데이터 저장소는 초당 요청 1개로 제한됩니다.
계정당 GetCapabilities 요청 속도	지원되는 각 지역: 10개	아니요	계정당 초당 수행할 수 있는 최대 GetCapabilities 요청 수입니다.
계정당 L istFHIRDatastores 요청 속도	지원되는 각 지역: 10개	아니요	계정당 초당 수행할 수 있는 최대 L istFHIRDatastores 요청 수입니다.
계정당 L istFHIRExportJobs 요청 속도	지원되는 각 지역: 10개	아니요	계정당 초당 수행할 수 있는 최대 L istFHIRExportJobs 요청 수입니다.
계정당 L istFHIRImportJobs 요청 비율	지원되는 각 지역: 10개	아니요	계정당 초당 수행할 수 있는 최대 L istFHIRImportJobs 요청 수입니다.
계정당 ListTagsforResource 요청 속도	지원되는 각 지역: 10개	아니요	계정당 초당 수행할 수 있는 최대 ListTagsforResource 요청 수입니다.
계정당 POST 요청 속도	지원되는 각 리전: 2,000	예	계정당 초당 수행할 수 있는 최대 POST 요청 수입니다.

명칭	기본값	조정 가능	설명
데이터 스토어당 POST 요청 속도	지원되는 각 리전: 1,000	예	데이터 스토어당 초당 수행할 수 있는 최대 POST 요청 수입니다. 2023년 8월 21일 이전에 생성된 데이터 저장소는 초당 요청 1개로 제한됩니다.
계정당 PUT 요청 속도	지원되는 각 리전: 2,000	예	계정당 초당 수행할 수 있는 최대 PUT 요청 수입니다.
데이터 스토어당 PUT 요청 속도	지원되는 각 리전: 1,000	예	데이터 스토어당 초당 수행할 수 있는 최대 PUT 요청 수입니다. 2023년 8월 21일 이전에 생성된 데이터 저장소는 초당 요청 1개로 제한됩니다.
계정당 S tartFHIRExport작업 요청 속도	지원되는 각 리전: 1	아니요	계정당 분당 수행할 수 있는 최대 S tartFHIRExport 작업 요청 수입니다.
계정POST당를 사용한 S tartFHIRExport 작업 요청 속도	지원되는 각 리전: 1	아니요	계정당 분당 수행할 수 POST 있는를 사용하는 최대 S tartFHIRExport작업 요청 수입니다.
계정당 S tartFHIRImport작업 요청 속도	지원되는 각 리전: 1	아니요	계정당 분당 수행할 수 있는 최대 S tartFHIRImport 작업 요청 수입니다.

명칭	기본값	조정 가능	설명
계정당 TagResource 요청 속도	지원되는 각 지역: 10개	아니요	초당 수행할 수 있는 최대 TagResource 요청 수입니다.
계정당 UntagResource 요청 속도	지원되는 각 지역: 10개	아니요	계정당 초당 수행할 수 있는 최대 UntagResource 요청 수입니다.
계정GET당를 사용한 검색 요청 속도	지원되는 각 리전: 200	예	계정당 초당 수행할 수 GET 있는를 사용하는 최대 검색 요청 수입니다.
데이터 스토어GET당를 사용한 검색 요청 속도	지원되는 각 리전: 100	예	데이터 스토어당 초당 수행할 수 GET 있는를 사용하는 최대 검색 요청 수입니다.
계정POST당를 사용한 검색 요청 속도	지원되는 각 리전: 200	예	초당 수행할 수 POST 있는를 사용하는 최대 검색 요청 수입니다.
데이터 스토어POST당를 사용한 검색 요청 속도	지원되는 각 리전: 100	예	데이터 스토어당 초당 수행할 수 POST 있는를 사용하는 최대 검색 요청 수입니다.
가져온 개별 파일의 크기	지원되는 각 리전: 5GB	아니요	S tartFHIRImport작업에 포함된 개별 파일의 최대 크기(GB)입니다.
가져오기 작업 전체 크기	지원되는 각 리전: 500GB	아니요	가져오기 작업에 포함된 모든 파일의 최대 크기(GB)

문제 해결

다음 설명서는 사용 시 발생할 수 있는 문제를 해결하는 데 도움이 될 수 있습니다 AWS HealthLake.

주제

HealthLake 데이터 스토어를 생성할 수 없는 이유는 무엇인가요?

2022년 11월 14일에 새 데이터 스토어를 생성하는 데 필요한 IAM 권한을 HealthLake 업데이트했습니다. 에 액세스하는 사용자 또는 역할에 연결된 정책을 업데이트하지 않은 경우 다음 오류가 HealthLake 발생합니다.

```
AccessDeniedException: Insufficient Lake Formation permission(s): Required Database on
 Catalog
```

데이터 스토어 생성에 대한 업데이트된 IAM 정책 요구 사항을 보려면 AWS 관리형 정책을 참조하세요 AmazonHealthLakeFullAccess. IAM 사용자 또는 역할에 이러한 정책을 추가하는 방법에 대한 지침은 step-by-step 섹션을 참조하세요사용을 시작할 권한 설정 AWS HealthLake.

데이터 스토어를 생성하려면 대칭 고객 소유 또는 Amazon 소유 KMS 키도 사용해야 합니다. IAM 정책에 올바른 권한이 있는지 확인합니다. 자세한 내용은 AWS Key Management Service 개발자 안내서AWS Key Management Service의 섹션을 AWS KMS참조하세요.

계정당 허용되는 데이터 스토어 수를 초과했습니다.

HealthLake 에는 계정당 10개의 데이터 스토어 할당량이 있습니다. 할당량 증가를 요청하는 방법을 알아보려면 AWS 지원 센터를 방문하세요.

에 대한 권한 부여를 생성하려면 어떻게 FHIR RESTful 해야 합니까 APIs?

사용자는 서명 버전 4 서명 프로세스를 사용하여 HTTP 클라이언트를 통해 전송된 요청에 인증을 HealthLake API 추가해야 합니다. 자세한 내용은 서명 버전 4 서명 프로세스를 참조하세요.

AWS SDK for Python을 사용하여 sigv4 인증을 생성하려면 다음 예제와 유사한 스크립트를 생성합니다.

```
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())
```

PythonAWSSDK용를 사용하여 sigv4 권한 부여를 사용하는 방법에 대한 자세한 내용은 Boto3 보안 인증 항목에서 확인할 수 있습니다.

데이터가 FHIR R4 형식이 아닙니다.를 계속 사용할 수 있나요 HealthLake?

FHIR R4 형식의 데이터만 HealthLake 데이터 스토어로 가져올 수 있습니다. 사용자가 데이터를 변환하는 데 도움이 되는 제품을 제공하는 파트너 목록은 AWS HealthLake 파트너를 참조하세요.

고객 관리형 KMS 키로 암호화된 데이터 스토어에 FHIR RESTful APIs를 사용할 때 AccessDenied 오류가 발생하는 이유는 무엇입니까?

사용자 또는 역할이 데이터 스토어에 액세스하려면 고객 관리형 키와 IAM 정책 모두에 대한 권한이 필요합니다. 사용자는 고객 관리형 키를 사용하는 데 필요한 IAM 권한이 있어야 합니다. 사용자가 고객 관리형 KMS 키를 사용할 수 있는 권한을 부여한 HealthLake 권한을 취소하거나 사용 중지한 경우 HealthLake 는 오류를 반환합니다 AccessDenied.

HealthLake 에는 고객 데이터에 액세스하고, 데이터 스토어로 가져온 새 FHIR 리소스를 암호화하고, 요청 시 FHIR 리소스를 해독할 수 있는 권한이 있어야 합니다.

자세한 내용은 키 액세스 문제 해결을 참조하세요.

내 가져오기가 실패한 이유는 무엇입니까?

가져오기 작업이 성공하면 output inputFileName.ndjson 파일이 있는 폴더가 생성되지만 개별 레코드를 가져오지 못할 수 있습니다. 이 경우 가져오기에 실패한 레코드의 매니페스트와 함께

두 번째 FAILURE 폴더가 생성됩니다. 매니페스트 파일에 액세스할 작업 출력 위치는 입니다 JobPropertiesJobOutputDataConfig.S3Configuration.S3Uri.

이 매니페스트 파일에는 성공한 모든 응답의 위치(successOutput.successOutputS3Uri), 실패한 모든 응답의 위치(failureOutput.failureOutputS3Uri) 및 추가 작업 지표와 같은 작업 출력에 대한 세부 정보가 포함되어 있습니다. 매니페스트 파일의 내용은 프로그래밍 방식으로 구문 분석할 수 있습니다. 다음 샘플 매니페스트 파일에는 Amazon S3 버킷의 입력 및 출력과 스캔한 리소스 수 및 성공적으로 가져온 리소스 수에 대한 정보가 나열되어 있습니다.

```
            {
            "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
            }
```

가져오기 작업이 실패한 이유를 분석하려면 D escribeFHIRImport작업을 사용하여를 분석API합니다 JobProperties. 다음은 권장됩니다.

- 상태가 FAILED 이고 메시지가 있는 경우 실패는 입력 데이터 크기 또는 HealthLake 할당량을 초과하는 입력 파일 수와 같은 작업 파라미터와 관련이 있습니다.
- 가져오기 작업 상태가 COMPLETED_WITH_인 경우 매니페스트 파일인 Manifest.json에서 성공적으로 가져오지 못한 파일에 대한 정보를 ERRORS확인합니다.
- 가져오기 작업 상태가 FAILED 이고 메시지가 없는 경우 작업 출력 위치로 이동하여 매니페스트 파일인 Manifest.json에 액세스합니다.

각 입력 파일에 대해 가져오기에 실패한 리소스의 입력 파일 이름이 있는 실패 출력 파일이 있습니다. 응답에는 입력 데이터의 위치에 해당하는 행 번호(lineId), FHIR 응답 객체(UpdateResourceResponse) 및 응답의 상태 코드(statusCode)가 포함됩니다.

샘플 출력 파일은 다음과 같습니다.

```
{"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}
```

이 예제는 입력 파일의 해당 입력 라인에서 3, 4, 7, 9, 15행에 장애가 있음을 보여줍니다. 이러한 각 줄에 대한 설명은 다음과 같습니다.

- 3행에서 응답은 입력 파일의 3행에 resourceType 제공된가 유효하지 않다고 설명합니다.
- 5행에서 응답은 입력 파일의 5행에 FHIR 검증 오류가 있음을 설명합니다.
- 라인 7에서 응답은 입력으로 resourceId 제공된에 검증 문제가 있음을 설명합니다.
- 라인 9에서 응답은 입력 파일에 유효한 리소스 ID가 포함되어야 한다고 설명합니다.
- 15행에서 입력 파일의 응답은 파일이 유효한 JSON 형식이 아니라는 것입니다.

처리할 수 없는 리소스를 찾으 DocumentReference려면 어떻게 해야 합니까?

DocumentReference 리소스가 유효하지 않은 경우 HealthLake 는 통합 의료 NLP 출력 대신 검증 오류를 나타내는 확장을 제공합니다. NLP 처리 중에 검증 오류가 발생한 DocumentReference 리소스를 찾기 위해 고객은 검색 키 cm-decoration-status 및 검색 값 VALIDATION_ERROR와 함께 HealthLake의 검색 함수를 사용할 수 있습니다. 이 검색에는 오류의 특성을 설명하는 오류 메시지와 함께 검증 오류가 발생한 모든 DocumentReference 리소스가 나열됩니다. 검증 오류가 있는 DocumentReference 리소스의 확장 필드 구조는 다음 예제와 유사합니다.

```
"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/"
          }
    ]
```

NLP 장식으로 인해 중첩된 객체가 10,000개 이상 생성되는 경우에도 VALIDATION_ERROR가 발생할 수 있습니다. 이 경우 처리 전에 문서를 더 작은 문서로 분할해야 합니다.

Amazon Athena를 사용하도록 기존 데이터 스토어 마이그레이션

2022년 11월 14일 이전에 생성된 데이터 스토어는 작동하지만를 사용하여 Athena에서 쿼리할 수 없습니다SQL. Athena를 사용하여 기존 데이터 스토어를 쿼리하려면 먼저 새 데이터 스토어로 마이그레이션해야 합니다.

데이터를 새 데이터 스토어로 마이그레이션하려면

1. 새 데이터 스토어를 생성합니다.
2. 기존의 데이터를 Amazon S3 버킷으로 내보냅니다.
3. Amazon S3 버킷에서 새 데이터 스토어로 데이터를 가져옵니다.

Amazon S3 버킷으로 데이터를 내보내면 추가 요금이 발생합니다. 추가 요금은 내보내는 데이터의 크기에 따라 다릅니다.

Athena의 검색 결과를 다른 AWS 서비스에 연결

Athena의 검색 결과를 다른 AWS 서비스와 공유할 때 문제가 발생할 수 있습니다.

를 SQL 검색 쿼리의 `json_extract[1]` 일부로 사용하면 문제가 발생할 수 있습니다.

이 문제를 해결하려면를 로 업데이트해야 합니다CATVAR.

저장 결과, 테이블(정적) 또는 보기(동적)를 생성하려고 할 때이 문제가 발생할 수 있습니다.

새 데이터 스토어로 데이터를 가져온 후 Athena 콘솔이 작동하지 않습니다.

데이터를 새 데이터 스토어로 가져온 후에는 데이터를 즉시 사용하지 못할 수 있습니다. 이는 데이터가 Iceberg 테이블에 수집될 시간을 허용하기 위한 것입니다. 나중에 다시 시도하세요.

새 데이터 레이크 관리자를 추가할 PutDataLakeSettings 때 Lake Formation 권한 오류: lakeformation:가 발생하는 이유는 무엇입니까?

IAM 사용자 또는 역할에 AWSLakeFormationDataAdmin AWS 관리형 정책이 포함된 경우 새 데이터 레이크 관리자를 추가할 수 없습니다. 다음을 포함하는 오류가 발생합니다.

```
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
```

AWS 관리형 정책은 IAM 사용자 또는 역할을 AWS Lake Formation 데이터 레이크 관리자로 추가하는 데 AdministratorAccess 필요합니다. IAM 사용자 또는 역할에도 작업이 포함된 경우 작업이 실패AWSLakeFormationDataAdmin합니다. AWSLakeFormationDataAdmin AWS 관리형 정책에는 AWS Lake Formation API 작업에 대한 설명 거부가 포함되어 있습니다PutDataLakeSetting.

AdministratorAccess AWS 관리형 정책을 AWS 사용하여에 대한 전체 액세스 권한이 있는 관리자도 AWSLakeFormationDataAdmin 정책에 의해 제한될 수 있습니다.

HealthLake의 통합 자연어 처리 기능을 켜려면 어떻게 해야 합니까?

2023년 2월 20일부터 HealthLake 데이터 스토어의 기본 동작이 변경되었습니다.

현재 데이터 스토어: 현재 모든 HealthLake 데이터 스토어는 base64 인코딩 DocumentReference 리소스에서 자연어 처리(NLP) 사용을 중지합니다. 즉NLP, 새 DocumentReference 리소스는를 사용하여 분석되지 않으며 리소스 유형의 텍스트를 기반으로 새 DocumentReference 리소스가 생성되지 않습니다. 기존 DocumentReference 리소스의 경우를 통해 생성된 데이터 및 리소스는 NLP 유지되지만 2023년 2월 20일 이후에는 업데이트되지 않습니다.

새 데이터 스토어:2023년 2월 20일 이후에 생성된 HealthLake 데이터 스토어는 base64 인코딩 DocumentReference 리소스에서 자연어 처리(NLP)를 수행하지 않습니다.

이 기능을 켜려면를 사용하여 사례를 생성해야 합니다AWS Support Center Console. 사례를 생성하려면에 로그인 AWS 계정한 다음 사례 생성을 선택합니다. 사례 및 사례 관리 생성에 대한 자세한 내용은 Support 사용 설명서의 지원 사례 및 사례 관리 생성을 참조하세요.

내 데이터 스토어 상태가 생성에서 변경되지 않음

새 HealthLake 데이터 스토어를 생성하려고 하는데 데이터 스토어 상태가 생성에서 변경되지 않는 경우를 사용하도록 Athena를 업데이트해야 합니다 AWS Glue Data Catalog.

자세한 내용은 Amazon Athena 사용 설명서의 AWS Glue 데이터 카탈로그로 업그레이드를 step-by-step 참조하세요.

를 성공적으로 업그레이드 AWS Glue Data Catalog한 후 이제 데이터 스토어를 생성할 수 있습니다.

이전 데이터 스토어를 제거하려면를 사용하여 사례를 생성하여 시작합니다AWS Support Center Console. 사례를 생성하려면에 로그인 AWS 계정한 다음 사례 생성을 선택합니다. 자세한 내용은 Support 사용 설명서의 지원 사례 및 사례 관리 생성을 참조하세요.

SDK 데이터 스토어 생성 상태가 예외 또는 알 수 없음 상태를 반환합니다.

목록 데이터 스토어 또는 설명 데이터 스토어 API 호출이 예외 또는 알 수 없는 데이터 스토어 상태를 반환SDK하는 경우를 최신 버전으로 업데이트하십시오.

413Request13Request Entity Too Large 오류를 가져오기 위해 10MB 문서를 사용하는 FHIR POST API 작업입니다. HealthLake

AWS HealthLake 에는 지연 시간 및 제한 시간을 늘리지 않도록 5MB의 동기식 생성 및 업데이트 API 제한이 있습니다.

대량 가져오기를 사용하여 바이너리를 ResourceType 사용하여 최대 164MB의 대용량 문서를 수집할 수 있습니다API.

AWS HealthLake 개발자 안내서의 문서 기록

다음 표에서는 AWS HealthLake 릴리스의 설명서 변경 사항을 설명합니다.

- API 버전: 최신
- 최신 설명서 업데이트: 10/25/2024

변경 사항	설명	날짜
HealthLake 는 이제 FHIR `history` 및 `vread` 상호 작용을 지원합니다.	HealthLake 는 이제 특정 리소스의 기록을 검색하기 위한 FHIR `history` 상호 작용과 리소스의 버전별 읽기를 수행하기 위한 `vread` 상호 작용을 지원합니다.	2024년 10월 25일
HealthLake 는 이제 새 FHIR 검색 파라미터, 확장 및 리소스 유형을 지원합니다.	HealthLake 는 이제 새 FHIR 검색 파라미터, 확장 및 리소스 유형을 지원합니다.	2023년 12월 9일
HealthLake 는 이제 FHIR 프레임워크SMART에서를 지원합니다.	HealthLake 는 이제 FHIR 활성화된 HealthLake 데이터 스토어SMART에서 생성을 지원합니다.	2023년 5월 31일
HealthLake 에서 이제 프로파일 검증 지원	HealthLake 는 이제 FHIR 프로파일 검증을 지원합니다.	2023년 5월 31일
HealthLake 에서 이제 지원 `export`	HealthLake 는 이제 FHIR REST API 작업을 사용하여 파일 내보내기를 지원합니다`export`.	2023년 5월 31일
아시아 태평양(뭄바이) 리전	AWS HealthLake 이제 아시아 태평양(뭄바이) 리전에서를 사용할 수 있습니다.	2023년 4월 4일

통합 자연어 처리가 꺼짐	HealthLake 는 2023년 2월 20일부터 모든 데이터 스토어에서 통합 자연어 처리(NLP)를 끕니다.	2023년 2월 20일
HealthLake Amazon Athena와 통합	이제 Athena를 사용하여 2022년 11월 14일 이후에 생성된 데이터 스토어를 쿼리할 수 있습니다.	2022년 11월 14일
총 가져오기 작업 크기 증가	S tartFHIRImport작업 요청의 모든 파일의 최대 총 크기는 이제 500GB입니다.	2022년 10월 3일
번들 지원	HealthLake 는 이제 여러 리소스를 수집하기 위한 번들 리소스 유형을 지원합니다.	2022년 8월 5일
의 CRUD 작업에 대한 할당량 업데이트 HealthLake	HealthLake 는 이제 CRUD 요청에 대해 더 높은 제한을 지원합니다.	2022년 7월 14일
지원 포함	HealthLake 는 이제 데이터 스토어 쿼리_include에서를 지원합니다.	2022년 7월 14일
AWS HealthLake 이제를 일반적으로 사용할 수 있습니다.	HealthLake 이제를 일반적으로 사용할 수 있습니다.	2020년 7월 30일

AWS 용어집

최신 AWS 용어는 AWS 용어집 참조의 AWS 용어집을 참조하세요.

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.