



개발자 가이드

Amazon Kinesis Video Streams



Amazon Kinesis Video Streams: 개발자 가이드

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon의 상표 및 브랜드 디자인은 Amazon 외 제품 또는 서비스와 관련하여 고객에게 혼동을 일으킬 수 있는 방식이나 Amazon 브랜드 이미지를 떨어뜨리는 방식으로 사용할 수 없습니다. Amazon이 소유하지 않은 기타 모든 상표는 Amazon과 제휴 관계이거나 관련이 있거나 후원 관계와 관계없이 해당 소유자의 자산입니다.

Table of Contents

Kinesis Video Streams란 무엇입니까?	1
리전 가용성	2
Kinesis Video Streams를 처음 사용하시나요?	3
시스템 사양	4
카메라 요구 사항	4
테스트를 거친 운영 체제	5
SDK 스토리지 요구 사항	5
작동 방식	6
API 및 생산자 라이브러리	7
Kinesis Video Streams API	7
엔드포인트 검색 패턴	9
프로듀서 라이브러리	10
비디오 재생	10
재생 요구 사항	11
HLS를 사용한 비디오 재생	15
MPEG-DASH를 사용한 비디오 재생	25
메타데이터 스트리밍 사용	29
Kinesis 비디오 스트림에 메타데이터 추가	30
Kinesis 비디오 스트림에 내장된 메타데이터 사용	31
스트리밍 메타데이터 제한	32
데이터 모델	33
스트림 헤더 요소	33
스트림 트랙 데이터	39
프레임 헤더 요소	40
MKV 프레임 데이터	40
시작하기	41
계정을 설정합니다.	41
가입하여 AWS 계정	42
관리자 액세스 권한이 있는 사용자 생성	42
키 생성 AWS 계정	43
Kinesis 비디오 스트림 생성	44
콘솔을 사용하여 비디오 스트림을 생성합니다.	44
를 사용하여 비디오 스트림을 생성하십시오. AWS CLI	44
Amazon Kinesis 비디오 스트림으로 데이터 전송	45

SDK와 샘플을 빌드하세요.	45
샘플을 실행하여 Kinesis Video Streams에 미디어를 업로드합니다.	48
승인 객체를 검토하세요.	50
미디어 데이터 사용	51
콘솔에서 미디어 보기	51
HLS를 사용하여 미디어 데이터를 소비하십시오.	51
엣지 에이전트	52
아마존 키네시스 비디오 스트림 엣지 에이전트 API 오퍼레이션	53
아마존 Kinesis Video Streams 엣지 에이전트 모니터링	53
비모드로 배포AWS IoT Greengrass	53
1. 종속성 설치	54
2. IP 카메라 RTSP URL을 위한 리소스 생성	55
3. IAM 권한 정책 생성	57
4. IAM 역할 생성	59
5. AWS IoT 역할 별칭 생성	60
6. AWS IoT 정책 생성	61
7. AWS IoT 사물 생성 및 AWS IoT Core 자격 증명 가져오기	63
8. Amazon Kinesis Video Streams Edge 에이전트의 구축 및 실행	65
9. (선택 사항) CloudWatch 에이전트 설치	75
10. (선택 사항) Amazon Kinesis Video Streams 엣지 에이전트 실행	78
에 배포하세요. AWS IoT Greengrass	80
1. 우분투 인스턴스 생성	81
2. AWS IoT Greengrass 코어 디바이스 설정	82
3. IP 카메라 RTSP URL을 위한 리소스 생성	83
4. TES 역할에 권한 추가	85
5. 시크릿 매니저 컴포넌트 설치	88
6. 디바이스에 Amazon Kinesis Video Streams 엣지 에이전트를 배포하십시오.	91
7. (선택 사항) AWS IoT Greengrass 로그 관리자 구성 요소 설치	99
FAQ	103
Amazon Kinesis Video Streams Edge 에이전트는 어떤 운영 체제를 지원합니까?	103
Amazon Kinesis Video Streams Edge 에이전트는 H.265 미디어를 지원합니까?	103
Amazon Kinesis Video Streams Edge 에이전트는 AL2에서 작동합니까?	103
사물 또는 장치 내에서 여러 스트림을 실행하려면 어떻게 AWS IoT 해야 합니까?	103
전송된 StartEdgeConfigurationUpdate 후 편집하려면 어떻게 해야 합니까?	104
공통적인 예가 ScheduleConfigs 있으신가요?	104
최대 스트리밍 한도가 있나요?	105

오류가 발생한 작업을 다시 시작하려면 어떻게 해야 하나요?	105
Amazon Kinesis Video Streams Edge 에이전트의 상태를 모니터링하려면 어떻게 해야 하나요?	105
VPC를 통한 비디오 스트리밍	106
추가 정보	106
VPC 엔드포인트 프로시저	106
이미지	109
GetImages 시작하기	109
Amazon S3 전송 시작하기	109
UpdateImageGenerationConfiguration	110
DescribeImageGenerationConfiguration	112
프로듀서 MKV 태그	113
다음을 사용하여 Producer SDK에 메타데이터 태그 추가PutEventMetaData	113
제한	114
S3 오브젝트 메타데이터	114
S3 오브젝트 경로 (이미지)	115
스로틀링 방지를 위한 Amazon S3 URI 권장 사항	115
알림	116
UpdateNotificationConfiguration	116
DescribeNotificationConfiguration	116
프로듀서 MKV 태그	113
프로듀서 MKV 태그의 구문	113
MKV 태그 제한	117
.....	117
.....	117
Amazon SNS 주제 페이로드	117
Amazon SNS 메시지 보기	118
보안	120
데이터 보호	121
Kinesis Video Streams의 서버 측 암호화란 무엇입니까?	121
비용, 지역 및 성능 고려 사항	121
서버 측 암호화를 시작하려면 어떻게 해야 하나요?	122
고객 관리 키 생성 및 사용	122
고객 관리 키를 사용할 수 있는 권한	123
IAM을 사용하여 Kinesis Video Streams 리소스에 대한 액세스 제어	125
정책 구문	125

Kinesis Video Streams를 위한 작업	126
Kinesis Video Streams용 Amazon 리소스 이름(ARN)	127
다른 IAM 계정에 Kinesis 비디오 스트림에 대한 액세스 권한 부여	127
정책 예제	130
를 사용하여 Kinesis Video Streams 리소스에 대한 액세스 제어 AWS IoT	132
AWS IoT ThingName 스트림 이름으로	133
AWS IoT CertificateId 스트림 이름으로	139
AWS IoT 자격 증명을 사용하여 하드 코딩된 스트림 이름으로 스트리밍할 수 있습니다.	141
모니터링	142
규정 준수 검증	142
복원력	143
인프라 보안	143
보안 모범 사례	144
최소 권한 액세스 구현	144
IAM 역할 사용	144
API 호출을 모니터링하는 CloudTrail 데 사용합니다.	145
생산자 라이브러리	146
Kinesis 비디오 스트림 프로듀서 클라이언트	146
Kinesis 비디오 스트림 프로듀서 라이브러리	147
관련 주제	148
Java 생산자 라이브러리	148
절차: Java 생산자 SDK 사용	148
1 단계: 코드 다운로드 및 구성	149
2단계: 코드 작성 및 검토	151
3단계: 코드 실행 및 확인	152
Android 생산자 라이브러리	153
절차: Android 생산자 SDK 사용	153
사전 조건	154
1 단계: 코드 다운로드 및 구성	157
2단계: 코드 검사	159
3단계: 코드 실행 및 확인	161
C++ 생산자 라이브러리	162
객체 모델	162
스트림에 미디어 넣기	162
콜백 인터페이스	163
절차: C++ 생산자 SDK 사용	163

1 단계: 코드 다운로드 및 구성	164
2단계: 코드 작성 및 검토	165
3단계: 코드 실행 및 확인	171
C++ 프로듀서 SDK를 GStreamer 플러그인으로 사용	172
C++ 프로듀서 SDK를 도커 컨테이너에서 GStreamer 플러그인으로 사용	172
로깅 사용	172
C 생산자 라이브러리	173
객체 모델	173
스트림에 미디어 넣기	174
절차: C 생산자 SDK 사용	175
2단계: 코드 다운로드	176
2단계: 코드 작성 및 검토	176
3단계: 코드 실행 및 확인	179
라즈베리 파이용 C++ 프로듀서 SDK	181
사전 조건	182
Kinesis Video Streams에 쓰기 권한을 가진 IAM 사용자 생성	182
라즈베리파이를 Wi-Fi 네트워크에 연결하세요.	184
라즈베리파이에 원격으로 연결	184
라즈베리파이 카메라 설정하기	185
소프트웨어 사전 요구 사항 설치	186
Kinesis Video Streams C++ 프로듀서 SDK를 다운로드하고 빌드하십시오.	187
Kinesis 비디오 스트림으로 비디오를 스트리밍하고 라이브 스트림을 보십시오.	187
레퍼런스	188
프로듀서 SDK 한도	189
오류 코드 참조	192
NAL 적응 플래그	240
생산자 구조	241
스트림 구조	243
콜백	260
스트림 구문 분석기 라이브러리	268
절차: Kinesis 비디오 스트림 파서 라이브러리 사용	268
사전 조건	268
1단계: 코드 다운로드 및 구성	269
다음 단계	269
2단계: 코드 작성 및 검토	269
StreamingMkvReader	270

FragmentMetadataVisitor	270
OutputSegmentMerger	272
KinesisVideoExample	273
다음 단계	276
3단계: 코드 실행 및 확인	276
예	278
예: Kinesis Video Streams로 데이터 전송	278
예: Kinesis Video Streams에서 데이터 검색	278
예: 비디오 데이터 재생	278
사전 조건	278
스트리머 플러그인 - kvssink	279
GStreamer 요소를 다운로드, 빌드 및 구성합니다.	280
GStreamer 요소를 실행하세요.	280
시작 명령	281
도커 컨테이너에서 GStreamer 요소를 실행합니다.	283
파라미터 참조	286
PutMedia API	298
1단계: 코드 다운로드 및 구성	299
2단계: 코드 작성 및 검토	300
3단계: 코드 실행 및 확인	302
RTSP 및 도커	303
자습서 비디오	303
사전 조건	303
도커 이미지를 빌드하십시오.	304
RTSP 예제 애플리케이션을 실행합니다.	304
렌더러	306
사전 조건	306
렌더러 예제 실행	306
작동 방식	307
모니터링	310
다음을 통한 모니터링 지표 CloudWatch	310
CloudWatch 지표 지침	325
를 사용하여 Amazon Kinesis Video Streams 엣지 에이전트를 모니터링하기 CloudWatch	329
CloudWatch Amazon Kinesis Video Streams 엣지 에이전트에 대한 메트릭 지침	331
을 사용하여 CloudTrail API 호출 로깅	333
아마존 키네시스 비디오 스트림 및 CloudTrail	334

예: Amazon Kinesis Video Streams 로그 파일 항목	335
할당량	339
컨트롤 플레인 API 서비스 할당량	339
미디어 및 아카이브 미디어 API 서비스 할당량	344
프래그먼트 메타데이터 및 프래그먼트 미디어 할당량	348
프래그먼트 메타데이터 할당량	351
스트림 태그	351
문제 해결	352
일반 문제	352
너무 긴 지연 시간	352
API 문제	353
오류: "알 수 없는 옵션"	353
오류: "Unable to determine service/operation name to be authorized"(권한을 부여할 서비스/ 작업 이름을 확인할 수 없음)	353
오류: "Failed to put a frame in the stream"(스트림에 프레임을 넣지 못했음)	354
오류: "최종 연결이 수신되기 전에 서비스가 연결을 AckEvent 끊었습니다."	354
오류: "STATUS_STORE_OUT_OF_MEMORY"	354
HLS 문제	355
Java 문제	355
Java 로그 활성화	355
프로듀서 라이브러리 문제	356
생산자 SDK를 컴파일할 수 없음	356
비디오 스트림이 콘솔에 나타나지 않음	357
오류: "GStreamer 데모 애플리케이션을 사용하여 데이터를 스트리밍할 때 "요청에 포함된 보 안 토큰이 잘못되었음"	357
오류: "Kinesis 비디오 클라이언트에 프레임을 제출하지 못 함"	357
OS X에 "스트리밍 중지됨, 이유가 협상되지 않음" 메시지와 함께 GStreamer 애플리케이션이 중지됨	358
오류: Raspberry Pi 기반으로 GStreamer 데모에서 Kinesis 비디오 클라이언트를 만들 때 "힙 할당에 실패했음"	358
오류: Raspberry Pi 기반으로 GStreamer 데모를 실행하는 동안 "잘못된 명령"	359
카메라가 Raspberry Pi에서 로드하지 못함	359
macOS High Sierra에서 카메라를 찾을 수 없음	360
macOS High Sierra에서 컴파일 시 jni.h 파일을 찾을 수 없음	360
GStreamer 데모 애플리케이션 실행 시 Curl 오류 발생	360
Raspberry Pi에서 런타임 시 타임스탬프/범위 어설션	360

Raspberry Pi에서 gst_value_set_fraction_range_full에 어설션	360
Android에서 STATUS_MKV_INVALID_ANNEXB_NALU_IN_FRAME_DATA(0x3200000d) 오류 발생	360
최대 프래그먼트 지속 시간에 도달했습니다. 오류	361
IoT 권한 부여를 사용할 때 "Invalid thing name passed(잘못된 이름이 전달되었습니다)" 오류 발생	361
스트림 파서 라이브러리 문제	362
스트림에서 단일 프레임에 액세스할 수 없음	362
조각 디코딩 오류	362
네트워크 문제	363
문서 기록	364
API 참조	368
작업	368
Amazon Kinesis Video Streams	369
Amazon Kinesis Video Streams	492
Amazon Kinesis Video Streams 아카이브	508
Amazon Kinesis Video Kinesis Video Kin	554
Amazon Kinesis Video Streams	562
데이터 유형	566
Amazon Kinesis Video Streams	567
Amazon Kinesis Video Streams	607
Amazon Kinesis Video Streams	610
Amazon Kinesis Video	628
Amazon Kinesis Video Streams	630
일반적인 오류	630
공통 파라미터	632
.....	dcxxxv

Kinesis Video Streams란 무엇입니까?

완전 AWS 서비스 관리형 Amazon Kinesis Video Streams를 사용하여 디바이스에서 디바이스로 라이브 비디오를 스트리밍하거나 실시간 비디오 처리 또는 배치 지향 비디오 분석을 위한 애플리케이션을 구축할 수 있습니다. AWS 클라우드

Kinesis Video Streams는 비디오 데이터만을 위한 스토리지가 아닙니다. 이것을 사용하여 클라우드에서 비디오 스트림을 수신하여 볼 수 있습니다. 에서 라이브 스트림을 모니터링하거나 Kinesis Video Streams API 라이브러리를 사용하여 라이브 비디오를 표시하는 자체 모니터링 애플리케이션을 개발할 수 있습니다. AWS Management Console

Kinesis Video Streams를 사용하여 스마트폰, 보안 카메라, 웹캠, 자동차에 내장된 카메라, 드론 및 기타 소스를 비롯한 수백만 개의 소스에서 대량의 라이브 비디오 데이터를 캡처할 수 있습니다. 오디오 데이터, 열화상 이미지, 깊이 데이터, RADAR 데이터 등 비디오가 아닌 시계열 데이터도 전송할 수 있습니다. 이러한 소스에서 Kinesis 비디오 스트림으로 라이브 비디오가 스트리밍되면 지연 시간이 짧은 처리를 위해 실시간으로 데이터에 액세스하는 애플리케이션을 구축할 수 있습니다. frame-by-frame Kinesis Video Streams는 소스에 구애받지 않습니다. [스트리머 플러그인 - kvssink](#) 라이브러리를 사용하여 컴퓨터 웹캠에서 비디오를 스트리밍하거나 실시간 스트리밍 프로토콜 (RTSP) 을 사용하여 네트워크의 카메라에서 비디오를 스트리밍할 수 있습니다.

또한 지정된 보존 기간 동안 미디어 데이터를 안정적으로 저장하도록 Kinesis 비디오 스트림을 구성할 수 있습니다. Kinesis Video Streams는 이 데이터를 자동으로 저장하고 저장된 데이터를 암호화합니다. 또한 Kinesis Video Streams는 생산자 타임스탬프와 수집 타임스탬프를 모두 기반으로 저장된 데이터를 타임인덱싱합니다. 정기적으로 비디오 데이터를 일괄 처리하는 애플리케이션을 구축할 수도 있고, 사용 사례별로 기간별 데이터에 한 번 액세스해야 하는 애플리케이션을 만들 수도 있습니다.

사용자 지정 애플리케이션 (실시간 또는 배치 지향) 은 Amazon EC2 인스턴스에서 실행할 수 있습니다. 이러한 애플리케이션은 오픈 소스 딥 러닝 알고리즘을 사용하여 데이터를 처리하거나 Kinesis Video Streams와 통합되는 타사 애플리케이션을 사용할 수 있습니다.

Kinesis Video Streams 사용의 이점은 다음과 같습니다.

- 수백만 개의 디바이스에서 연결 및 스트리밍 — Kinesis Video Streams를 사용하여 소비자 스마트폰, 드론, 대시캠 등 수백만 개의 디바이스에서 비디오, 오디오 및 기타 데이터를 연결하고 스트리밍할 수 있습니다. Kinesis Video Streams 프로듀서 라이브러리를 사용하여 디바이스를 구성하고 실시간 또는 미디어 after-the-fact 업로드로 안정적으로 스트리밍할 수 있습니다.
- 데이터의 안정적인 저장, 암호화 및 색인 생성 - 사용자 지정 보존 기간 동안 미디어 데이터를 안정적으로 저장하도록 Kinesis 비디오 스트림을 구성할 수 있습니다. 또한 Kinesis Video Streams는 생산

자 생성 또는 서비스 측 타임스탬프를 기반으로 저장된 데이터에 대한 인덱스를 생성합니다. 애플리케이션은 타임인덱스를 사용하여 스트림에서 지정된 데이터를 검색할 수 있습니다.

- 인프라 대신 애플리케이션 관리에 집중 — Kinesis Video Streams는 서버리스이므로 인프라를 설정하거나 관리할 필요가 없습니다. 데이터 스트림과 소비 애플리케이션 수가 늘어나거나 줄어들기 때문에 기본 인프라의 배포, 구성 또는 탄력적 확장에 대해 걱정할 필요가 없습니다. Kinesis Video Streams는 스트림 관리에 필요한 모든 관리 및 유지 관리를 자동으로 수행하므로 사용자는 인프라가 아닌 애플리케이션에 집중할 수 있습니다.
- 데이터 스트림에서 실시간 및 배치 애플리케이션 구축 — Kinesis Video Streams를 사용하여 라이브 데이터 스트림에서 작동하는 사용자 지정 실시간 애플리케이션을 구축하고 엄격한 지연 시간 요구 사항 없이 안정적으로 지속되는 데이터에서 작동하는 배치 또는 일회성 애플리케이션을 생성할 수 있습니다. 스트림을 처리하고 분석하는 데 사용하여 오픈 소스 (Apache MXnet, OpenCV), 자체 개발 또는 타사 솔루션 등 사용자 지정 애플리케이션을 구축, 배포 및 관리할 수 있습니다. AWS Marketplace Kinesis Video Get Streams API를 사용하여 실시간 또는 배치 중심으로 데이터를 처리하는 여러 개의 동시 애플리케이션을 구축할 수 있습니다.
- 더 안전한 데이터 스트리밍 - Kinesis Video Streams는 서비스를 통해 전송되는 모든 데이터를 암호화하고 데이터가 지속되는 동안 암호화합니다. Kinesis Video Streams는 디바이스의 데이터 스트리밍에 TLS (전송 계층 보안) 기반 암호화를 적용하고 () 를 사용하여 저장된 모든 데이터를 암호화합니다. AWS Key Management Service AWS KMS 또한 (IAM) 을 사용하여 AWS Identity and Access Management 데이터에 대한 액세스를 관리할 수 있습니다.
- 사용한 만큼 지불 — 자세한 내용은 을 참조하십시오 [AWS Pricing Calculator](#).

리전 가용성

Amazon Kinesis Video Streams는 다음 지역에서 사용할 수 있습니다.

리전 이름	AWS지역 코드
미국 동부(오하이오)	us-east-2
미국 동부(버지니아 북부)	us-east-1
미국 서부(오레곤)	us-west-2
아프리카(케이프타운)	af-south-1
아시아 태평양(홍콩)	ap-east-1

리전 이름	AWS 지역 코드
아시아 태평양(뭄바이)	ap-south-1
아시아 태평양(서울)	ap-northeast-2
아시아 태평양(싱가포르)	ap-southeast-1
아시아 태평양(시드니)	ap-southeast-2
아시아 태평양(도쿄)	ap-northeast-1
캐나다(중부)	ca-central-1
중국(베이징)	cn-north-1
유럽(프랑크푸르트)	eu-central-1
유럽(아일랜드)	eu-west-1
유럽(런던)	eu-west-2
유럽(파리)	eu-west-3
남아메리카(상파울루)	sa-east-1

Kinesis Video Streams를 처음 사용하시나요?

Kinesis Video Streams를 처음 사용하는 경우 다음 섹션을 순서대로 읽는 것이 좋습니다.

1. [Kinesis Video Streams: 작동 방식](#)— Kinesis Video Streams의 개념에 대해 알아보기.
2. [아마존 키네시스 비디오 스트림 시작하기](#)— 계정을 설정하고 Kinesis Video Streams를 테스트하려면
3. [Kinesis 비디오 스트림 프로듀서 라이브러리](#)— Kinesis Video Streams 프로듀서 애플리케이션을 만드는 방법에 대해 알아봅니다.
4. [Kinesis 비디오 스트림 파서 라이브러리](#)— Kinesis Video Streams 소비자 애플리케이션에서 들어오는 데이터 프레임을 처리하는 방법에 대해 알아봅니다.
5. [아마존 키네시스 비디오 스트림 예제](#)— Kinesis Video Streams로 수행할 수 있는 작업에 대한 더 많은 예를 보려면

Kinesis Video Streams 시스템 요구 사항

다음 섹션에는 Amazon Kinesis Video Streams에 대한 하드웨어, 소프트웨어 및 스토리지 요구 사항이 포함되어 있습니다.

주제

- [카메라 요구 사항](#)
- [테스트를 거친 운영 체제](#)
- [SDK 스토리지 요구 사항](#)

카메라 요구 사항

Kinesis Video Streams Producer SDK 및 샘플을 실행하는 데 사용되는 카메라의 메모리 요구 사항은 다음과 같습니다.

- SDK 콘텐츠 보기에는 16MB 메모리가 필요합니다.
- 샘플 애플리케이션 기본 구성은 128MiB 메모리입니다. 이 값은 네트워크 연결이 양호하고 추가 버퍼링 요구 사항이 없는 생산자에게 적합합니다. 네트워크 연결이 약하고 추가 버퍼링이 필요한 경우 초당 프레임 속도에 프레임 메모리 크기를 곱하여 버퍼링의 초당 메모리 요구 사항을 계산할 수 있습니다. 메모리 할당에 대한 자세한 내용은 [StorageInfo](#)를 참조하십시오.

CPU에서 인코딩 워크로드를 제거할 수 있도록 H.264를 사용하여 데이터를 인코딩하는 USB 또는 RTSP(실시간 스트리밍 프로토콜) 카메라를 사용하는 것이 좋습니다.

현재 데모 애플리케이션은 RTSP 스트리밍을 위한 사용자 데이터그램 프로토콜 (UDP) 을 지원하지 않습니다. 향후 이 기능을 추가할 예정입니다.

생산자 SDK는 다음 두 가지 유형의 카메라를 지원합니다.

- 웹 카메라.
- USB 카메라.
- H.264 인코딩을 지원하는 카메라(기본 설정).
- H.264 인코딩을 지원하지 않는 카메라.
- Raspberry Pi 카메라 모듈. 이것은 Raspberry Pi 디바이스에서 기본 설정입니다. 왜냐하면 비디오 데이터 전송을 위해 GPU에 연결되므로 CPU 처리에 오버헤드가 들지 않기 때문입니다.

- RTSP(네트워크) 카메라. 이 카메라는 비디오 스트림이 이미 H.264로 인코딩되어 있으므로 기본 설정입니다.

테스트를 거친 운영 체제

다음 디바이스와 운영 체제로 웹 카메라와 RTSP 카메라를 테스트했습니다.

- Mac mini
 - High Sierra
- MacBook 프로급 노트북
 - Sierra(10.12)
 - El Capitan(10.11)
- Ubuntu 16.04를 실행하는 HP 랩톱
- Ubuntu 17.10(도커 컨테이너)
- Raspberry Pi 3

SDK 스토리지 요구 사항

[Kinesis 비디오 스트림 프로듀서 라이브러리](#) 설치 최소 요구 사항은 170MB이고 권장 스토리지 요구 사항은 512MB입니다.

Kinesis Video Streams: 작동 방식

주제

- [Kinesis Video Streams API 및 프로듀서 라이브러리 지원](#)
- [Kinesis Video Streams 재생](#)
- [Kinesis Video Streams에서 스트리밍 메타데이터 사용](#)
- [Kinesis Video Streams 데이터 모델](#)

완전 AWS 서비스관리형 Amazon Kinesis Video Streams를 사용하여 디바이스에서 디바이스로 라이브 비디오를 스트리밍하고 안정적으로 AWS 클라우드 저장할 수 있습니다. 그리고 실시간 비디오 처리를 위한 자체 애플리케이션을 빌드하거나 배치 중심 비디오 분석을 수행할 수 있습니다.

다음 다이어그램은 Kinesis Video Streams의 작동 방식에 대한 개요를 제공합니다.

다이어그램은 다음 구성 요소 간의 상호 작용을 보여 줍니다.

- 프로듀서 - Kinesis 비디오 스트림에 데이터를 넣는 모든 소스. 프로듀서는 보안 카메라, 신체 착용 카메라, 스마트폰 카메라 또는 대시보드 카메라와 같은 모든 비디오 생성 장치가 될 수 있습니다. 생산자는 오디오 피드, 이미지, RADAR 데이터와 같은 비영상 데이터를 전송할 수도 있습니다.

단일 생산자가 하나 이상의 비디오 스트림을 생성할 수 있습니다. 예를 들어 비디오 카메라는 비디오 데이터를 한 Kinesis 비디오 스트림으로 푸시하고 오디오 데이터를 다른 Kinesis 비디오 스트림으로 푸시할 수 있습니다.

- Kinesis Video Streams 프로듀서 라이브러리 — 디바이스에 설치하고 구성할 수 있는 소프트웨어 및 라이브러리 세트입니다. 이러한 라이브러리를 사용하면 비디오를 안전하게 연결하고 몇 초 동안 버퍼링한 후 실시간으로 또는 미디어 업로드로 after-the-fact 스트리밍하는 등 다양한 방식으로 비디오를 안전하게 연결하고 안정적으로 스트리밍할 수 있습니다.
- Kinesis 비디오 스트림 — 라이브 비디오 데이터를 전송하고, 선택적으로 저장하고, 실시간 및 일괄 또는 일회성으로 데이터를 사용할 수 있도록 하는 데 사용할 수 있는 리소스입니다. 일반적인 구성에서 Kinesis 비디오 스트림에는 데이터를 게시하는 제작자가 한 명만 있습니다.

스트림은 오디오, 비디오 및 타임 인코딩된 유사 데이터 스트림(예: 깊이 센서 피드, RADAR 피드 등)을 전송할 수 있습니다. 를 AWS Management Console 사용하거나 SDK를 사용하여 프로그래밍 방식으로 Kinesis 비디오 스트림을 생성합니다. AWS

여러 개의 독립 애플리케이션이 Kinesis 비디오 스트림을 병렬로 사용할 수 있습니다.

- 소비자 — Kinesis 비디오 스트림에서 프래그먼트 및 프레임과 같은 데이터를 가져와 보고 처리하거나 분석합니다. 일반적으로 이러한 소비자를 Kinesis Video Streams 애플리케이션이라고 합니다. Kinesis Video Streams의 데이터를 실시간으로 소비하고 처리하는 애플리케이션을 작성하거나, 짧은 지연 시간 처리가 필요하지 않은 경우 데이터가 저장되고 타임인덱싱된 후에 작성할 수 있습니다. Amazon EC2 인스턴스에서 실행되도록 이러한 소비자 애플리케이션을 생성할 수 있습니다.
- [Kinesis 비디오 스트림 파서 라이브러리](#) — Kinesis Video Streams 애플리케이션이 지연 시간이 짧은 방식으로 Kinesis 비디오 스트림에서 미디어를 안정적으로 가져올 수 있습니다. 또한 애플리케이션이 프레임 자체의 처리 및 분석에 집중할 수 있도록 미디어에서 프레임 경계를 파싱합니다.

Kinesis Video Streams API 및 프로듀서 라이브러리 지원

Kinesis Video Streams는 스트림을 생성 및 관리하고 스트림에서 미디어 데이터를 읽고 쓸 수 있는 API를 제공합니다. Kinesis Video Streams 콘솔은 관리 기능 외에도 라이브 video-on-demand 및 재생을 지원합니다. 또한 Kinesis Video Streams는 애플리케이션 코드에서 미디어 소스에서 데이터를 추출하고 Kinesis 비디오 스트림에 업로드하는 데 사용할 수 있는 프로듀서 라이브러리 세트를 제공합니다.

주제

- [Kinesis Video Streams API](#)
- [엔드포인트 검색 패턴](#)
- [프로듀서 라이브러리](#)

Kinesis Video Streams API

Kinesis Video Streams는 Kinesis 비디오 스트림을 생성하고 관리하기 위한 API를 제공합니다. 다음과 같이 미디어 데이터를 읽고 스트림에 쓸 수 있는 API도 제공합니다.

- 프로듀서 API — Kinesis Video Streams는 Kinesis 비디오 스트림에 미디어 데이터를 쓰기 위한 API를 제공합니다. PutMedia. PutMedia 요청에서 생산자는 미디어 조각 스트림을 전송합니다. 조각은 독립적인 프레임 시퀀스입니다. 조각에 속한 프레임은 다른 조각의 어떤 프레임에 대한 종속성도 없어야 합니다. 자세한 정보는 [PutMedia](#)을 참조하세요.

프래그먼트가 도착하면 Kinesis Video Streams는 고유한 프래그먼트 번호를 오름차순으로 할당합니다. 또한 각 프래그먼트의 생산자측 및 서버측 타임스탬프를 Kinesis Video Streams별 메타데이터로 저장합니다.

- 소비자 API — 소비자는 다음 API를 사용하여 스트림에서 데이터를 가져올 수 있습니다.
- GetMedia - 이런 API를 사용할 때 소비자는 시작 조각을 식별해야 합니다. 그런 다음 API가 스트림에 추가되는 순서대로 조각을 반환합니다(조각 번호가 증가하는 순서대로). 조각에 있는 미디어 데이터는 [Matroska\(MKV\)](#)와 같은 구조화 형식으로 압축됩니다. 자세한 정보는 [GetMedia](#)을 참조하세요.

Note

GetMedia는 조각이 어디에 있는지 압니다(데이터 스토어에 보관되거나 실시간으로 사용 가능). 예를 들어, GetMedia가 시작 조각이 보관되어 있다고 판단하는 경우 데이터 스토어로부터 조각을 반환하기 시작합니다. 아직 보관되지 않은 새로운 프래그먼트를 반환해야 하는 경우 인메모리 스트림 버퍼에서 프래그먼트를 읽는 것으로 GetMedia 전환합니다.

이는 스트림에 의해 수집되는 순서대로 조각을 처리하는 지속적 소비자의 예입니다.

GetMedia는 비디오 처리 애플리케이션이 실패하거나 뒤편되는 경우 추가적인 작업 없이 따라잡을 수 있도록 합니다. 애플리케이션은 GetMedia를 사용하여 데이터 스토어에 보관되어 있는 데이터를 처리하고, 애플리케이션이 따라잡으면서 GetMedia가 계속하여 미디어 데이터를 데이터가 도착하는 대로 실시간으로 공급합니다.

- GetMediaFromFragmentList (및 ListFragments) - 배치 처리 애플리케이션은 오프라인 소비자로 간주됩니다. 오프라인 소비자는 ListFragments 및 GetMediaFromFragmentList API를 조합하여 특정 미디어 조각이나 비디오 범위를 명시적으로 가져올 수 있습니다. ListFragments 및 GetMediaFromFragmentList는 애플리케이션이 특정 시간 범위 혹은 조각 범위에 대해 비디오 조각을 식별한 다음 순차적으로 혹은 병렬로 가져와 처리할 수 있게 해 줍니다. 이와 같은 접근 방식은 대량의 데이터를 병렬로 신속하게 처리해야 하는 MapReduce 애플리케이션 제품군에 적합합니다.

예를 들어, 소비자가 하루 동안의 비디오 조각을 처리해야 하는 경우를 가정해 보겠습니다. 소비자는 다음을 수행할 것입니다.

1. ListFragmentsAPI를 호출하고 원하는 프래그먼트 컬렉션을 선택할 수 있는 시간 범위를 지정하여 프래그먼트 목록을 가져옵니다.

API는 지정된 범위 안에 있는 모든 조각으로부터 메타데이터를 반환합니다. 메타데이터는 프래그먼트 번호, 제작자 측 및 서버 측 타임스탬프 등과 같은 정보를 제공합니다.

2. 조각 메타데이터 목록을 취하고 임의의 순서대로 조각을 검색합니다. 예를 들어 소비자는 하루 동안의 모든 프래그먼트를 처리하기 위해 목록을 하위 목록으로 분할하고 작업자 (예: 여러 Amazon EC2 인스턴스) 가 를 사용하여 GetMediaFromFragmentList 프래그먼트를 병렬로 가져와서 병렬로 처리하도록 할 수 있습니다.

다음 다이어그램은 이들 API 호출 동안의 조각 및 청크의 데이터 흐름을 보여 줍니다.

생산자가 PutMedia 요청을 전송할 때 페이로드에 있는 미디어 메타데이터를 전송한 다음 미디어 데이터 조각의 시퀀스를 전송합니다. 데이터를 수신하면 Kinesis Video Streams는 들어오는 미디어 데이터를 Kinesis Video Streams 청크로 저장합니다. 각 청크는 다음과 같은 요소로 구성됩니다.

- 미디어 메타데이터의 사본
- 조각
- Kinesis Video Streams별 메타데이터 (예: 프래그먼트 번호, 서버측 및 생산자측 타임스탬프)

소비자가 미디어 메타데이터를 요청하면 Kinesis Video Streams는 요청에서 지정한 프래그먼트 번호부터 시작하여 청크 스트림을 반환합니다.

스트림의 데이터 지속성을 활성화하면 스트림에서 프래그먼트를 수신한 후 Kinesis Video Streams는 프래그먼트의 복사본도 데이터 스토어에 저장합니다.

엔드포인트 검색 패턴

컨트롤 플레인 REST API

[Kinesis Video Streams 컨트롤 플레인 REST API](#)에 액세스하려면 [Kinesis Video Streams](#) 서비스 엔드포인트를 사용하십시오.

데이터 플레인 REST API

Kinesis Video Streams는 더 나은 확장 및 트래픽 격리 속성을 보장하기 위해 [셀룰러](#) 아키텍처를 사용하여 구축되었습니다. 각 스트림은 지역의 특정 셀에 매핑되므로 애플리케이션은 스트림이 매핑된 올바른 셀별 엔드포인트를 사용해야 합니다. Data Plane REST API에 액세스할 때는 올바른 엔드포인트를 직접 관리하고 매핑해야 합니다. 이 프로세스인 엔드포인트 검색 패턴은 다음과 같습니다.

1. 엔드포인트 검색 패턴은 GetEndpoints 작업 중 하나에 대한 호출로 시작됩니다. 이러한 액션은 컨트롤 플레인에 속합니다.

1. [the section called “Amazon Kinesis Video Streams”](#) 또는 [the section called “Amazon Kinesis Video Streams 아카이브”](#) 서비스의 엔드포인트를 검색하는 경우 를 사용하십시오. [the section called “GetDataEndpoint”](#)
 2. [the section called “Amazon Kinesis Video Kinesis Video Kin”](#) the section called “Amazon Kinesis Video Streams”, 또는 [Kinesis 비디오](#) 시그널링에 대한 엔드포인트를 검색하는 경우에는 를 사용하십시오. [the section called “GetSignalingChannelEndpoint”](#)
2. 엔드포인트를 캐싱하고 재사용하십시오.
 3. 캐시된 엔드포인트가 더 이상 작동하지 않는 경우 를 새로 GetEndpoints 호출하여 엔드포인트 를 새로 고치십시오.

프로듀서 라이브러리

Kinesis 비디오 스트림을 생성한 후 스트림으로 데이터 전송을 시작할 수 있습니다. 애플리케이션 코드에서 이러한 라이브러리를 사용하여 미디어 소스에서 데이터를 추출하고 Kinesis 비디오 스트림에 업로드할 수 있습니다. 사용 가능한 생산자 라이브러리에 대한 자세한 내용은 [Kinesis 비디오 스트림 프로듀서 라이브러리](#) 단원을 참조하십시오.

Kinesis Video Streams 재생

다음 방법을 사용하여 Kinesis 비디오 스트림을 볼 수 있습니다.

- **GetMedia**— GetMedia API를 사용하여 Kinesis Video Streams를 처리하는 자체 애플리케이션을 구축할 수 있습니다. GetMedia 지연 시간이 짧은 실시간 API입니다. 를 사용하는 GetMedia 플러그인을 만들려면 직접 빌드해야 합니다. 를 사용하여 GetMedia Kinesis 비디오 스트림을 표시하는 애플리케이션을 개발하는 방법에 대한 자세한 내용은 을 참조하십시오. [스트림 구문 분석기 라이브러리](#)
- **HLS** — [HTTP 라이브 스트리밍 \(HLS\)](#) 은 업계 표준 HTTP 기반 미디어 스트리밍 통신 프로토콜입니다. HLS를 사용하여 Kinesis 비디오 스트림을 볼 수 있습니다 (라이브 재생 또는 아카이브된 비디오 보기).

실시간 재생을 위해 HLS를 사용할 수 있습니다. 지연 시간은 일반적으로 3~5초이지만 사용 사례, 플레이어 및 네트워크 상태에 따라 1~10초 사이일 수 있습니다. HLS 스트리밍 세션 URL을 프로그래밍 방식으로 또는 수동으로 제공하여 타사 플레이어(예: [Video.js](#) 또는 [Google Shaka Player](#))를 사용해 비디오 스트림을 표시할 수 있습니다. [Apple Safari](#) 또는 [Microsoft Edge](#) 브라우저의 위치 표시줄에 HLS 스트리밍 세션 URL을 입력하여 비디오를 재생할 수도 있습니다.

- MPEG-DASH — [MPEG-DASH라고도 하는 HTTP를 통한 동적 적응형 스트리밍 \(DASH\) 은 인터넷을 통해 기존 HTTP 웹 서버에서 제공하는 미디어 콘텐츠의 고품질 스트리밍을 가능하게 하는 적응형 비트레이트 스트리밍 프로토콜입니다.](#)

실시간 재생에 MPEG-DASH를 사용할 수 있습니다. 지연 시간은 일반적으로 3~5초이지만 사용 사례, 플레이어 및 네트워크 상태에 따라 1~10초 사이일 수 있습니다. 타사 플레이어 (예: [dash.js](#) 또는 [Google Shaka Player](#)) 를 사용하여 MPEG-DASH 스트리밍 세션 URL을 프로그래밍 방식으로 또는 수동으로 제공하여 비디오 스트림을 표시할 수 있습니다.

- GetClip— GetClip API를 사용하여 지정된 비디오 스트림에서 보관된 온디맨드 미디어가 포함된 클립 (MP4 파일) 을 지정된 시간 범위 동안 다운로드할 수 있습니다. 자세한 내용은 API 참조를 참조하십시오. [GetClip](#)

주제

- [비디오 재생 트랙 요구 사항](#)
- [HLS를 사용한 비디오 재생](#)
- [MPEG-DASH를 사용한 비디오 재생](#)

비디오 재생 트랙 요구 사항

Amazon Kinesis Video Streams는 여러 형식으로 인코딩된 미디어를 지원합니다. Kinesis 비디오 스트림이 아래 나열된 네 가지 API 중 하나에서 지원하지 않는 형식을 사용하는 경우 트랙 유형 제한이 없는 [GetMedia](#) 또는 [GetMediaForFragmentList](#)를 사용하십시오.

주제

- [GetClip 요구 사항](#)
- [GetDash URL 요구 사항 StreamingSession](#)
- [GetHL의 URL 요구 사항 StreamingSession](#)
- [GetImages 요구 사항](#)

GetClip 요구 사항

이 API에 대한 자세한 내용은 [GetClip](#) 단원을 참조하십시오.

트랙 1 설명	트랙 1 코덱 ID	트랙 2 설명	트랙 2 코덱 ID
H.264 비디오	V_MPEG/ISO/AVC	N/A	N/A
H.264 비디오	V_MPEG/ISO/AVC	AAC 오디오	A_AAC
H.264 비디오	V_MPEG/ISO/AVC	G.711 오디오 (A-Law 전용)	A_MS/ACM
H.265 비디오	V_MPEGH/ISO/HEVC	N/A	N/A
H.265 비디오	V_MPEGH/ISO/HEVC	AC 오디오	A_AAC

⚠ Important

각 프래그먼트에 포함된 코덱 프라이빗 데이터 (CPD)에는 프래그먼트를 올바르게 디코딩하는 데 필요한 프레임 속도, 해상도 및 인코딩 프로파일과 같은 코덱별 초기화 정보가 포함되어 있습니다. 결과 클립의 대상 조각 간에는 CPD 변경이 지원되지 않습니다. CPD는 쿼리된 미디어에서 일관성을 유지해야 합니다. 그렇지 않으면 오류가 반환됩니다.

⚠ Important

트랙 변경은 지원되지 않습니다. 트랙은 쿼리된 미디어 전체에서 일관성을 유지해야 합니다. 스트림의 프래그먼트가 비디오만 포함되다가 오디오와 비디오를 모두 포함하는 것으로 변경되거나 AAC 오디오 트랙이 A-Law 오디오 트랙으로 변경되면 오류가 반환됩니다.

GetDash URL 요구 사항 StreamingSession

이 API에 대한 자세한 내용은 [GetDASHStreamingSessionURL](#) 단원을 참조하세요.

트랙 1 설명	트랙 1 코덱 ID	트랙 2 설명	트랙 2 코덱 ID
H.264 비디오	V_MPEG/ISO/AVC	N/A	N/A
H.264 비디오	V_MPEG/ISO/AVC	AAC 오디오	A_AAC

트랙 1 설명	트랙 1 코덱 ID	트랙 2 설명	트랙 2 코덱 ID
H.264 비디오	V_MPEG/ISO/AVC	G.711 오디오 (A-Law 전용)	A_MS/ACM
H.264 비디오	V_MPEG/ISO/AVC	G.711 오디오 (U-Law 전용)	A_MS/ACM
AAC 오디오	A_AAC	N/A	N/A
H.265 비디오	V_MPEGH/ISO/HEVC	N/A	N/A
H.265 비디오	V_MPEGH/ISO/HEVC	AC 오디오	A_AAC

⚠ Important

각 프래그먼트에 포함된 코덱 프라이빗 데이터 (CPD)에는 프래그먼트를 올바르게 디코딩하는 데 필요한 프레임 속도, 해상도 및 인코딩 프로파일과 같은 코덱별 초기화 정보가 포함되어 있습니다. 스트리밍 세션 중에는 CPD 변경이 지원되지 않습니다. CPD는 쿼리된 미디어를 통해 일관성을 유지해야 합니다.

⚠ Important

트랙 변경은 지원되지 않습니다. 트랙은 쿼리된 미디어 전체에서 일관성을 유지해야 합니다. 스트림의 프래그먼트가 비디오만 포함되다가 오디오와 비디오를 모두 포함하는 것으로 변경되거나 AAC 오디오 트랙이 A-Law 오디오 트랙으로 변경되면 스트리밍이 실패합니다.

GetHL의 URL 요구 사항 StreamingSession

이 API에 대한 자세한 내용은 [GetHLSStreamingSessionURL](#) 단원을 참조하세요.

HLS Mp4

트랙 1 설명	트랙 1 코덱 ID	트랙 2 설명	트랙 2 코덱 ID
H.264 비디오	V_MPEG/ISO/AVC	N/A	N/A

트랙 1 설명	트랙 1 코덱 ID	트랙 2 설명	트랙 2 코덱 ID
H.264 비디오	V_MPEG/ISO/AVC	AAC 오디오	A_AAC
AAC 오디오	A_AAC	N/A	N/A
H.265 비디오	V_MPEGH/ISO/HEVC	N/A	N/A
H.265 비디오	V_MPEGH/ISO/HEVC	AC 오디오	A_AAC

홀스 티에스

트랙 1 설명	트랙 1 코덱 ID	트랙 2 설명	트랙 2 코덱 ID
H.264 비디오	V_MPEG/ISO/AVC	N/A	N/A
H.264 비디오	V_MPEG/ISO/AVC	AAC 오디오	A_AAC
AAC 오디오	A_AAC	N/A	N/A

Note

각 프래그먼트에 포함된 코덱 프라이빗 데이터 (CPD)에는 프래그먼트를 올바르게 디코딩하는 데 필요한 프레임 속도, 해상도 및 인코딩 프로파일과 같은 코덱별 초기화 정보가 포함되어 있습니다. TS와 MP4 모두 스트리밍 세션 중에 CPD 변경이 지원됩니다. 따라서 재생을 중단하지 않고도 세션의 프래그먼트가 CPD에서 다른 정보를 가질 수 있습니다. 각 스트리밍 세션에서 CPD 변경은 500개만 허용됩니다.

Important

트랙 변경은 지원되지 않습니다. 트랙은 쿼리된 미디어 전체에서 일관성을 유지해야 합니다. 스트림의 프래그먼트가 비디오만 포함되다가 오디오와 비디오를 모두 포함하는 것으로 변경되거나 AAC 오디오 트랙이 A-Law 오디오 트랙으로 변경되면 스트리밍이 실패합니다.

GetImages 요구 사항

이 API에 대한 자세한 내용은 [GetImages](#) 단원을 참조하세요.

Note

GetImages 미디어에는 트랙 1의 비디오 트랙이 포함되어야 합니다.

HLS를 사용한 비디오 재생

[HTTP 라이브 스트리밍 \(HLS\)](#) 은 업계 표준 HTTP 기반 미디어 스트리밍 통신 프로토콜입니다. HLS를 사용하여 Kinesis 비디오 스트림을 볼 수 있습니다 (라이브 재생 또는 아카이브된 비디오 보기).

실시간 재생을 위해 HLS를 사용할 수 있습니다. 지연 시간은 일반적으로 3~5초이지만 사용 사례, 플레이어 및 네트워크 상태에 따라 1~10초 사이일 수 있습니다. HLS 스트리밍 세션 URL을 프로그래밍 방식으로 또는 수동으로 제공하여 타사 플레이어(예: [Video.js](#) 또는 [Google Shaka Player](#))를 사용해 비디오 스트림을 표시할 수 있습니다. [Apple Safari](#) 또는 [Microsoft Edge](#) 브라우저의 위치 표시줄에 HLS 스트리밍 세션 URL을 입력하여 비디오를 재생할 수도 있습니다.

[HLS를 사용하여 Kinesis 비디오 스트림을 보려면 먼저 GetHLS URL을 사용하여 스트리밍 세션을 생성하십시오.](#) [StreamingSession](#) 이 작업은 HLS 세션에 액세스하기 위한 URL(세션 토큰을 포함)을 반환합니다. 그런 다음 미디어 플레이어 또는 독립 실행형 애플리케이션에서 URL을 사용하여 스트림을 표시할 수 있습니다.

Important

Kinesis Video Streams로 전송된 모든 미디어를 HLS를 통해 재생할 수 있는 것은 아닙니다. 특정 업로드 [the section called "GetHLSStreamingSessionURL"](#) 요구 사항은 를 참조하십시오.

주제

- [를 AWS CLI 사용하여 HLS 스트리밍 세션 URL을 검색할 수 있습니다.](#)
- [예: HTML에서 HLS를 사용하고 JavaScript](#)
- [HLS 문제 해결](#)

를 AWS CLI 사용하여 HLS 스트리밍 세션 URL을 검색할 수 있습니다.

아래 절차에 따라 를 사용하여 Kinesis 비디오 스트림용 HLS 스트리밍 세션 URL을 생성하십시오.

AWS CLI

설치 지침은 사용 [AWS Command Line Interface 설명서](#)를 참조하십시오. 설치 후 자격 증명 및 지역을 AWS CLI 사용하여 [구성하십시오](#).

또는 AWS CLI 설치 및 구성된 AWS CloudShell 터미널을 여십시오. 자세한 내용은 [AWS CloudShell 사용 설명서](#)를 참조하십시오.

Kinesis 비디오 스트림의 HLS URL 엔드포인트를 검색하십시오.

1. 터미널에 다음을 입력합니다.

```
aws kinesisvideo get-data-endpoint \
  --api-name GET_HLS_STREAMING_SESSION_URL \
  --stream-name YourStreamName
```

다음과 같은 응답을 받게 됩니다.

```
{
  "DataEndpoint": "https://b-1234abcd.kinesisvideo.aws-region.amazonaws.com"
}
```

2. 반환된 엔드포인트에 HLS 스트리밍 세션 URL을 요청합니다.

Live

실시간 재생의 경우 HLS 미디어 재생 목록은 최신 미디어가 제공되는 대로 계속 업데이트됩니다. 미디어 플레이어에서 이러한 유형의 세션을 재생하면 일반적으로 사용자 인터페이스에 “라이브” 알림이 표시되며, 표시할 재생 창에서 위치를 선택할 수 있는 스크러버 컨트롤이 없습니다.

이 명령을 실행할 때 미디어를 이 스트림에 업로드하고 있는지 확인하십시오.

```
aws kinesis-video-archived-media get-hls-streaming-session-url \
  --endpoint-url https://b-1234abcd.kinesisvideo.aws-region.amazonaws.com \
  --stream-name YourStreamName \
  --playback-mode LIVE
```

Live replay

라이브 리플레이의 경우 지정된 시작 시간부터 재생이 시작됩니다. 또한 HLS 미디어 플레이리스트는 최신 미디어가 제공되는 대로 계속 업데이트됩니다. 세션에는 세션이 만료될 때까지 또는 지정된 종료 시간 중 먼저 도래하는 시점까지 새로 수집된 미디어가 계속 포함됩니다. 이 모드는 이벤트가 감지된 시점부터 재생을 시작하고 세션 생성 시점에 아직 인제스트되지 않은 미디어를 계속 라이브 스트리밍할 수 있는 데 유용합니다.

시작 타임스탬프를 결정하십시오.

이 예제에서는 Unix Epoch 시간을 초 형식으로 사용합니다. [타임스탬프](#) 형식에 대한 자세한 내용은 AWS Command Line Interface 사용 설명서의 타임스탬프 섹션을 참조하십시오.

변환 도구에 [UnixTime대해서는.org](#)를 참조하십시오.

- 1708471800은 2024년 2월 20일 오후 3:30:00 GMT-08:00 과 같습니다.

이 예에서는 종료 타임스탬프를 지정하지 않습니다. 즉, 세션이 만료될 때까지 세션에 새로 수집된 미디어가 계속 포함됩니다.

LIVE_REPLAY 재생 모드를 지정하고 [HLS](#) 프래그먼트 선택터를 지정하여 GetHLSStreamingSessionURL API를 호출합니다.

```
aws kinesis-video-archived-media get-hls-streaming-session-url \
  --endpoint-url https://b-1234abcd.kinesisvideo.aws-region.amazonaws.com \
  --stream-name YourStreamName \
  --playback-mode LIVE_REPLAY \
  --hls-fragment-selector \
  "FragmentSelectorType=SERVER_TIMESTAMP, TimestampRange={StartTimestamp=1708471800}"
```

On-demand

온디맨드 재생의 경우 HLS 미디어 플레이리스트에는 HLS 프래그먼트 선택터로 지정된 미디어가 포함됩니다. 미디어 플레이어에서 이러한 유형의 세션을 재생할 때 사용자 인터페이스에는 일반적으로 재생 창에서 표시할 위치를 선택할 수 있는 스크러버 컨트롤이 표시됩니다.

스트림의 특정 섹션에 대한 URL을 만들려면 먼저 시작 타임스탬프와 종료 타임스탬프를 결정해야 합니다.

이 예제에서는 Unix Epoch 시간을 초 형식으로 사용합니다. [타임스탬프](#) 형식에 대한 자세한 내용은 AWS Command Line Interface 사용 설명서의 타임스탬프 섹션을 참조하십시오.

변환 도구에 [UnixTime대해서는.org](#)를 참조하십시오.

- 1708471800은 2024년 2월 20일 오후 3:30:00 GMT-08:00 과 같습니다.
- 1708471860은 2024년 2월 20일 오후 3:31:00 GMT-08:00 과 같습니다.

[재생 모드 및 HLS 프래그먼트 셀렉터를 지정한 상태에서 API를 호출합니다.](#)
[GetHLSStreamingSessionURL_ON_DEMAND](#)

```
aws kinesis-video-archived-media get-hls-streaming-session-url \
  --endpoint-url https://b-1234abcd.kinesisvideo.aws-region.amazonaws.com \
  --stream-name YourStreamName \
  --playback-mode ON_DEMAND \
  --hls-fragment-selector \
  "FragmentSelectorType=SERVER_TIMESTAMP, TimestampRange={StartTimestamp=1708471800, EndTime=1708471860}"
```

Note

설명서에 나와 있는 것처럼 타임스탬프는 서로 24시간 이내에 있어야 합니다. [the section called "HLSTimestampRange"](#)

다음과 같은 응답을 받게 됩니다.

```
{
  "HLSStreamingSessionURL": "https://b-1234abcd.kinesisvideo.aws-region.amazonaws.com/hls/v1/getHLSMasterPlaylist.m3u8?SessionToken=CiAz...DkRE6M~"
}
```

Important

승인되지 않은 주체가 액세스할 수 있는 곳에 이 토큰을 공유하거나 보관하지 마십시오. 토큰은 스트림 콘텐츠에 대한 액세스를 제공합니다. AWS 자격 증명에 사용하는 것과 동일한 방법으로 토큰을 보호하십시오.

이 URL과 모든 HLS 플레이어를 사용하여 HLS 스트림을 볼 수 있습니다.

예를 들어 VLC 미디어 플레이어를 사용하세요.

애플 사파리 또는 마이크로소프트 엣지 브라우저의 위치 표시줄에 HLS 스트리밍 세션 URL을 입력하여 HLS 스트림을 재생할 수도 있습니다.

예: HTML에서 HLS를 사용하고 JavaScript

다음 예제는 JavaScript v2용 AWS SDK를 사용하여 Kinesis 비디오 스트림의 HLS 스트리밍 세션을 검색하고 웹 페이지에서 재생하는 방법을 보여줍니다. 예제에서는 다음 플레이어에서 비디오를 재생하는 방법을 보여줍니다.

- [Video.js](#)
- [Google Shaka Player](#)
- [hls.js](#)

[전체 예제 코드와 호스팅된 웹 페이지](#)는 에서 확인하십시오. GitHub

코드 설명 주제:

- [JavaScript 브라우저용 AWS SDK 가져오기](#)
- [Kinesis Video Streams 클라이언트 설정](#)
- [HLS 재생을 위한 엔드포인트 검색](#)
- [Kinesis Video Streams 아카이브 미디어 클라이언트 설정](#)
- [HLS 스트리밍 세션 URL을 검색하십시오.](#)
- [웹 페이지에 HLS 스트림을 표시합니다.](#)

JavaScript 브라우저용 AWS SDK 가져오기

웹 페이지에 다음 스크립트 태그를 포함하여 JavaScript v2용 AWS SDK를 프로젝트로 가져옵니다.

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/aws-sdk/2.490.0/aws-sdk.min.js"></script>
```

자세한 내용은 [AWS SDK 설명서를 참조하십시오. JavaScript](#)

Kinesis Video Streams 클라이언트 설정

HLS를 사용하여 스트리밍 비디오에 액세스하려면 먼저 Kinesis Video Streams 클라이언트를 생성하고 구성하십시오. 다른 인증 방법은 [웹 브라우저에서 자격 증명 설정을](#) 참조하십시오.

```
const clientConfig = {
  accessKeyId: 'YourAccessKey',
  secretAccessKey: 'YourSecretKey',
  region: 'us-west-2'
};
const kinesisVideoClient = new AWS.KinesisVideo(clientConfig);
```

애플리케이션은 HTML 페이지의 입력 상자에서 필요한 값을 검색합니다.

HLS 재생을 위한 엔드포인트 검색

Kinesis Video Streams 클라이언트를 사용하여 API를 호출하여 [the section called “GetDataEndpoint”](#) 엔드포인트를 검색합니다.

```
const getDataEndpointOptions = {
  StreamName: 'YourStreamName',
  APIName: 'GET_HLS_STREAMING_SESSION_URL'
};
const getDataEndpointResponse = await kinesisVideoClient
  .getDataEndpoint(getDataEndpointOptions)
  .promise();
const hlsDataEndpoint = getDataEndpointResponse.DataEndpoint;
```

이 코드는 변수에 엔드포인트를 저장합니다. `hlsDataEndpoint`

Kinesis Video Streams 아카이브 미디어 클라이언트 설정

Kinesis Video Streams 아카이브 미디어 클라이언트의 클라이언트 구성에서 이전 단계에서 획득한 엔드포인트를 지정합니다.

```
const archivedMediaClientConfig = {
  accessKeyId: 'YourAccessKey',
  secretAccessKey: 'YourSecretKey',
  region: 'us-west-2',
  endpoint: hlsDataEndpoint
};
```

```
const kinesisVideoArchivedMediaClient = new
  AWS.KinesisVideoArchivedMedia(archivedMediaClientConfig);
```

HLS 스트리밍 세션 URL을 검색하십시오.

Kinesis Video Streams 아카이브 미디어 클라이언트를 사용하여 API를 [the section called "GetHLSStreamingSessionURL"](#) 호출하여 HLS 재생 URL을 검색합니다.

```
const getHLSStreamingSessionURLOptions = {
  StreamName: 'YourStreamName',
  PlaybackMode: 'LIVE'
};
const getHLSStreamingSessionURLResponse = await kinesisVideoArchivedMediaClient
  .getHLSStreamingSessionURL(getHLSStreamingSessionURLOptions)
  .promise();
const hlsUrl = getHLSStreamingSessionURLResponse.HLSStreamingSessionURL;
```

웹 페이지에 HLS 스트림을 표시합니다.

HLS 스트리밍 세션 URL이 있으면 비디오 플레이어에 제공합니다. 비디오 플레이어에 URL을 제공하는 방법은 사용되는 플레이어에 따라 다릅니다.

Video.js

[Video.js](#) 및 해당 CSS 클래스를 브라우저 스크립트로 가져오려면 다음과 같이 하십시오.

```
<link rel="stylesheet" href="https://vjs.zencdn.net/6.6.3/video-js.css">
<script src="https://vjs.zencdn.net/6.6.3/video.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/videojs-contrib-hls/5.14.1/
videojs-contrib-hls.js"></script>
```

비디오를 표시할 video HTML 요소를 만드세요.

```
<video id="videojs" class="player video-js vjs-default-skin" controls autoplay></
video>
```

HLS URL을 HTML 비디오 요소 소스로 설정합니다.

```
const playerElement = document.getElementById('videojs');
const player = videojs(playerElement);
player.src({
```

```

    src: hlsUrl,
    type: 'application/x-mpegURL'
  });
  player.play();

```

Shaka

[Google Shaka 플레이어](#)를 브라우저 스크립트로 가져오려면 다음과 같이 하세요.

```

<script src="https://cdnjs.cloudflare.com/ajax/libs/shaka-player/2.4.1/shaka-
player.compiled.js"></script>

```

동영상을 표시할 video HTML 요소를 만드세요.

```

<video id="shaka" class="player" controls autoplay></video>

```

비디오 요소를 지정하는 Shaka 플레이어를 만들고 load 메서드를 호출합니다.

```

const playerElement = document.getElementById('shaka');
const player = new shaka.Player(playerElement);
player.load(hlsUrl);

```

hls.js

[hls.js](#) 를 브라우저 스크립트로 가져오려면 다음과 같이 하십시오.

```

<script src="https://cdn.jsdelivr.net/npm/hls.js@latest"></script>

```

비디오를 표시할 video HTML 요소를 만드세요.

```

<video id="hlsjs" class="player" controls autoplay></video>

```

hls.js 플레이어를 만들고 HLS URL을 지정한 다음 재생하라고 지시하세요.

```

const playerElement = document.getElementById('hlsjs');
const player = new Hls();
player.loadSource(hlsUrl);
player.attachMedia(playerElement);
player.on(Hls.Events.MANIFEST_PARSED, function() {

```



```
video.play();
});
```

HLS 문제 해결

이 섹션에서는 Kinesis Video Streams와 함께 HTTP 라이브 스트리밍 (HLS) 을 사용할 때 발생할 수 있는 문제를 설명합니다.

문제

- [HLS 스트리밍 세션 URL 검색이 성공하지만 비디오 플레이어에서 재생이 실패함](#)
- [생산자와 플레이어 사이에 지연 시간이 너무 김](#)

HLS 스트리밍 세션 URL 검색이 성공하지만 비디오 플레이어에서 재생이 실패함

이 상황은 GetHLSStreamingSessionURL을 사용해 HLS 스트리밍 세션 URL을 성공적으로 검색할 수 있지만 비디오 플레이어에 이 URL이 제공하면 비디오가 재생되지 않는 경우에 발생합니다.

이 상황을 해결하려면 다음과 같이 시도해 보십시오.

- Kinesis Video Streams 콘솔에서 비디오 스트림을 재생할지 여부를 결정합니다. 콘솔에 표시되는 모든 오류를 고려합니다.
- 조각 기간이 1초 미만이라면 1초로 늘립니다. 프래그먼트 지속 시간이 너무 짧으면 서비스가 비디오 프래그먼트를 너무 자주 요청하기 때문에 플레이어가 제한될 수 있습니다.
- 각 HLS 스트리밍 세션 URL을 한 플레이에서만 사용하는지 확인합니다. 2대 이상의 플레이어가 단일 HLS 스트리밍 세션 URL을 사용할 경우 서비스가 너무 많은 요청을 수신하여 스로틀링이 발생할 수 있습니다.
- 플레이어가 HLS 스트리밍 세션에 지정하는 모든 옵션을 지원하는지 확인하십시오. 다음 파라미터 값을 다르게 조합하여 시도해 봅니다.
 - ContainerFormat
 - PlaybackMode
 - FragmentSelectorType
 - DiscontinuityMode
 - MaxMediaPlaylistFragmentResults

(HTML5 및 모바일 플레이어와 같은) 일부 미디어 플레이어에서는 일반적으로 fMP4 컨테이너 형식의 HLS만 지원됩니다. 다른 미디어 플레이어 (예: Flash 및 커스텀 플레이어) 는 MPEG TS 컨테이너

형식의 HLS만 지원할 수 있습니다. ContainerFormat 파라미터를 시험해 보고 문제 해결을 시작하는 것이 좋습니다.

- 각 조각의 트랙 수가 일정한지 확인합니다. 스트림의 프래그먼트가 오디오와 비디오 트랙을 둘 다 포함할지 아니면 비디오 트랙만 포함할지 여부를 결정하세요. 또한 각 트랙의 프래그먼트 간에 인코더 설정 (해상도 및 프레임 속도) 이 변경되지 않는지 확인하십시오.

생산자와 플레이어 사이에 지연 시간이 너무 김

이 상황은 비디오가 캡처되는 시점에서 비디오 플레이어에서 재생되는 시점까지의 지연 시간이 너무 긴 경우입니다.

비디오는 조각 단위로 HLS를 통해 재생됩니다. 따라서 지연 시간은 조각 기간보다 짧을 수 없습니다. 또한 지속 시간에는 데이터 버퍼링 및 전송에 필요한 시간도 포함됩니다. 솔루션에 1초 미만의 지연 시간이 필요할 경우 대신 GetMedia API를 사용할 것을 고려해 보십시오.

다음 파라미터를 조정하여 전체 지연 시간을 단축할 수 있습니다. 그러나 파라미터를 조정하면 비디오 품질이 저하되거나 재버퍼링 비율이 높아질 수도 있습니다.

- 프래그먼트 지속 시간 - 프래그먼트 지속 시간은 비디오 인코더에서 생성되는 키프레임의 빈도에 따라 제어되는 스트림의 분할 간 비디오 양입니다. 권장되는 값은 1초입니다. 조각 기간이 짧으면 서비스에 비디오 데이터를 전송하기 전에 조각이 완료될 때까지 대기해야 하는 시간이 작아집니다. 또한 조각이 짧을수록 서비스 처리 속도가 빨라집니다. 그러나 조각 기간이 너무 짧을 경우 플레이어가 콘텐츠를 떨어져 재생을 멈추고 콘텐츠를 버퍼링할 가능성이 증가합니다. 조각 기간이 500밀리초 미만일 경우 생산자가 너무 많은 요청을 생성하여 서비스에서 스로틀링이 발생할 수 있습니다.
- 비트 전송률 - 비트 전송률이 낮은 비디오 스트림은 읽기, 쓰기 및 전송하는 데 걸리는 시간이 더 짧습니다. 그러나 일반적으로 비디오 스트림의 비트레이트가 낮으면 비디오 품질도 떨어집니다.
- 미디어 재생 목록의 프래그먼트 수 - 지연 시간에 민감한 플레이어는 미디어 재생 목록의 최신 프래그먼트만 로드해야 합니다. 대부분의 플레이어는 대신 가장 빠른 프래그먼트부터 시작합니다. 재생 목록의 프래그먼트 수를 줄이면 이전 프래그먼트와 새 프래그먼트 사이의 시간 간격을 줄일 수 있습니다. 재생 목록 크기가 작으면 재생 목록에 새 조각을 추가하는 데 지연이 발생하거나 플레이어가 업데이트된 재생 목록을 가져오는 데 지연이 발생하는 경우 재생 중에 프래그먼트를 건너뛴 수 있습니다. 3~5개의 프래그먼트를 사용하고 재생 목록에서 최신 프래그먼트만 로드하도록 구성된 플레이어를 사용하는 것이 좋습니다.
- 플레이어 버퍼 크기 — 대부분의 비디오 플레이어에는 구성 가능한 최소 버퍼 지속 시간이 있으며, 일반적으로 기본값은 10초입니다. 지연 시간을 최소화하기 위해 이 값을 0초로 설정할 수 있습니다. 하지만 이렇게 하면 플레이어가 지연을 흡수할 버퍼가 없기 때문에 프래그먼트를 생성하는 데 지연이 발생하는 경우 플레이어가 다시 버퍼링하게 됩니다.

- 플레이어 “catch up” — 비디오 플레이어는 일반적으로 버퍼가 가득 차면 비디오 버퍼 앞쪽까지 자동으로 재생되지 않습니다. 예를 들어 지연된 프래그먼트로 인해 프래그먼트의 백로그가 재생되는 경우입니다. 사용자 지정 플레이어는 프레임 제거하거나 재생 속도를 높여(예: 1.1x) 버퍼의 앞부분까지 캐치업하여 이를 방지할 수 있습니다. 그러면 플레이어가 캐치업하는 과정에서 재생이 중간에 끊기거나 속도가 빨라질 수 있고, 버퍼 크기가 짧게 유지되면 재버퍼링이 더 빈번해질 수 있습니다.

MPEG-DASH를 사용한 비디오 재생

[MPEG-DASH를 사용하여 Kinesis 비디오 스트림을 보려면 먼저 GetDash URL을 사용하여 스트리밍 세션을 생성해야 합니다. `StreamingSession` 이 작업은 MPEG-DASH 세션에 액세스하기 위한 URL\(세션 토큰을 포함\)을 반환합니다. 그런 다음 미디어 플레이어 또는 독립 실행형 애플리케이션에서 URL을 사용하여 스트림을 표시할 수 있습니다.](#)

Amazon Kinesis 비디오 스트림에는 MPEG-DASH를 통해 비디오를 제공하기 위한 다음과 같은 요구 사항이 있습니다.

- 스트리밍 비디오 재생 트랙 요구 사항은 [을 참조하십시오. `the section called “대시 URL 가져오기 StreamingSession”`](#)
- 데이터 보존이 0보다 커야 합니다.
- 각 조각의 비디오 추적에는 H.264용 AVC(Advanced Video Coding) 형식 및 H.265용 HEVC 형식의 코덱 프라이빗 데이터가 포함되어야 합니다. 자세한 내용은 [MPEG-4 사양 ISO/IEC 14496-15](#)를 참조하십시오. 스트림 데이터를 지정된 형식에 맞게 조정하는 방법에 대한 자세한 내용은 [NAL 적응 플러그](#)를 참조하십시오.
- 각 조각의 오디오 조각에는 AAC 형식([AAC 사양 ISO/IEC 13818-7](#)) 또는 [MS Wave 형식](#)의 코덱 프라이빗 데이터가 포함되어야 합니다.

예: HTML에서 MPEG-DASH를 사용하고 JavaScript

다음 예제는 Kinesis 비디오 스트림의 MPEG-DASH 스트리밍 세션을 검색하고 웹 페이지에서 재생하는 방법을 보여줍니다. 예제에서는 다음 플레이어에서 비디오를 재생하는 방법을 보여줍니다.

- [Google Shaka Player](#)
- [dash.js](#)

주제

- [Kinesis Video Streams 클라이언트를 MPEG-DASH 재생을 위한 설정](#)

- [MPEG-DASH 재생을 위한 Kinesis Video Streams 아카이브 콘텐츠 엔드포인트 검색](#)
- [MPEG-DASH 스트리밍 세션 URL 검색](#)
- [MPEG-DASH 재생을 사용하여 스트리밍 비디오 표시](#)
- [완성된 예제](#)

Kinesis Video Streams 클라이언트를 MPEG-DASH 재생을 위한 설정

MPEG-DASH를 사용하여 스트리밍 비디오에 액세스하려면 먼저 Kinesis Video Streams 클라이언트 (서비스 엔드포인트 검색) 와 아카이브된 미디어 클라이언트 (MPEG-DASH 스트리밍 세션 검색) 를 생성하고 구성합니다. 애플리케이션은 HTML 페이지의 입력 상자에서 필요한 값을 검색합니다.

```
var streamName = $('#streamName').val();

// Step 1: Configure SDK Clients
var options = {
  accessKeyId: $('#accessKeyId').val(),
  secretAccessKey: $('#secretAccessKey').val(),
  sessionToken: $('#sessionToken').val() || undefined,
  region: $('#region').val(),
  endpoint: $('#endpoint').val() || undefined
}
var kinesisVideo = new AWS.KinesisVideo(options);
var kinesisVideoArchivedContent = new AWS.KinesisVideoArchivedMedia(options);
```

MPEG-DASH 재생을 위한 Kinesis Video Streams 아카이브 콘텐츠 엔드포인트 검색

클라이언트가 시작된 후 Kinesis Video Streams 아카이브 콘텐츠 엔드포인트를 검색하여 다음과 같이 MPEG-DASH 스트리밍 세션 URL을 검색할 수 있습니다.

```
// Step 2: Get a data endpoint for the stream
console.log('Fetching data endpoint');
kinesisVideo.getDataEndpoint({
  StreamName: streamName,
  APIName: "GET_DASH_STREAMING_SESSION_URL"
}, function(err, response) {
  if (err) { return console.error(err); }
  console.log('Data endpoint: ' + response.DataEndpoint);
  kinesisVideoArchivedContent.endpoint = new AWS.Endpoint(response.DataEndpoint);
```

MPEG-DASH 스트리밍 세션 URL 검색

보관된 콘텐츠 엔드포인트가 있는 경우 다음과 같이 [GetDash StreamingSession URL API를 호출하여 MPEG-DASH 스트리밍 세션 URL을](#) 검색하십시오.

```
// Step 3: Get a Streaming Session URL
var consoleInfo = 'Fetching ' + protocol + ' Streaming Session URL';
console.log(consoleInfo);

if (protocol === 'DASH') {
    kinesiusVideoArchivedContent.getDASHStreamingSessionURL({
        StreamName: streamName,
        PlaybackMode: $('#playbackMode').val(),
        DASHFragmentSelector: {
            FragmentSelectorType: $('#fragmentSelectorType').val(),
            TimestampRange: $('#playbackMode').val() === "LIVE" ? undefined : {
                StartTimestamp: new Date($('#startTimestamp').val()),
                EndTimestamp: new Date($('#endTimestamp').val())
            }
        },
        DisplayFragmentTimestamp: $('#displayFragmentTimestamp').val(),
        DisplayFragmentNumber: $('#displayFragmentNumber').val(),
        MaxManifestFragmentResults: parseInt($('#maxResults').val()),
        Expires: parseInt($('#expires').val())
    }, function(err, response) {
        if (err) { return console.error(err); }
        console.log('DASH Streaming Session URL: ' + response.DASHStreamingSessionURL);
    });
}
```

MPEG-DASH 재생을 사용하여 스트리밍 비디오 표시

MPEG-DASH 스트리밍 세션 URL이 있으면 비디오 플레이어에 제공합니다. 비디오 플레이어에 URL을 제공하는 방식은 사용하는 플레이어에 따라 다릅니다.

다음 코드 예제는 [Google Shaka](#) 플레이어에 스트리밍 세션 URL을 제공하는 방법입니다.

```
// Step 4: Give the URL to the video player.

//Shaka Player elements
<video id="shaka" class="player" controls autoplay></video>
```

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/shaka-player/2.4.1/shaka-
player.compiled.js">
</script>
...

var playerName = $('#player').val();

if (playerName === 'Shaka Player') {
  var playerElement = $('#shaka');
  playerElement.show();

  var player = new shaka.Player(playerElement[0]);
  console.log('Created Shaka Player');

  player.load(response.DASHStreamingSessionURL).then(function() {
    console.log('Starting playback');
  });
  console.log('Set player source');
}
```

다음 코드 예제는 [dash.js](#) 플레이어에 스트리밍 세션 URL을 제공하는 방법을 보여 줍니다.

```
<!-- dash.js Player elements -->
<video id="dashjs" class="player" controls autoplay=""></video>
<script src="https://cdn.dashjs.org/latest/dash.all.min.js"></script>

...

var playerElement = $('#dashjs');
playerElement.show();

var player = dashjs.MediaPlayer().create();
console.log('Created DASH.js Player');

player.initialize(document.querySelector('#dashjs'), response.DASHStreamingSessionURL,
  true);
console.log('Starting playback');
console.log('Set player source');
}
```

완성된 예제

에서 완성된 예제 코드를 [다운로드하거나 볼](#) 수 있습니다. GitHub

Kinesis Video Streams에서 스트리밍 메타데이터 사용

Amazon Kinesis Video Streams 프로듀서 SDK를 사용하여 Kinesis 비디오 스트림의 개별 프래그먼트 수준에서 메타데이터를 포함할 수 있습니다. Kinesis Video Streams의 메타데이터는 변경 가능한 키-값 쌍입니다. 이를 사용하여 프래그먼트의 내용을 설명하거나, 실제 프래그먼트와 함께 전송해야 하는 관련 센서 판독값을 포함하거나, 기타 사용자 지정 요구 사항을 충족할 수 있습니다. 메타데이터는 [the section called “GetMedia”](#) 또는 [the section called “GetMediaForFragmentList”](#) API 작업의 일부로 사용 가능합니다. 스트림의 전체 보존 기간 동안 프래그먼트와 함께 저장됩니다. 사용 중인 애플리케이션을 사용하여 메타데이터를 읽고, 처리하고, 이에 따라 대응할 수 있습니다. [Kinesis 비디오 스트림 파서 라이브러리](#)

다음과 같은 두 가지 모드를 통해 스트림의 조각과 함께 메타데이터를 포함할 수 있습니다.

- 비지속적 — 발생한 비즈니스별 기준에 따라 스트림의 프래그먼트에 일회성 또는 임시로 메타데이터를 첨부할 수 있습니다. Kinesis 비디오 스트림으로 프래그먼트를 전송하기 전에 움직임을 감지하고 모션이 포함된 해당 프래그먼트에 메타데이터를 추가하는 스마트 카메라를 예로 들 수 있습니다. 다음 형식으로 메타데이터를 조각에 적용할 수 있습니다. Motion = true
- 퍼시스턴트 - 지속적인 필요에 따라 스트림의 연속된 연속 프래그먼트에 메타데이터를 첨부할 수 있습니다. Kinesis 비디오 스트림으로 전송하는 모든 조각과 관련된 현재 위도 및 경도 좌표를 전송하는 스마트 카메라를 예로 들 수 있습니다. Lat = 47.608013N , Long = -122.335167W 형식으로 메타데이터를 모든 조각에 적용할 수 있습니다.

애플리케이션의 요구 사항에 따라 이 두 가지 모드의 메타데이터를 동일한 조각에 동시에 부착할 수 있습니다. 포함된 메타데이터에는 감지된 객체, 추적된 활동, GPS 좌표 또는 스트림의 조각에 연결하려는 기타 사용자 지정 데이터가 포함될 수 있습니다. 메타데이터는 키-값 문자열 페어로 인코딩됩니다.

주제

- [Kinesis 비디오 스트림에 메타데이터 추가](#)
- [Kinesis 비디오 스트림에 내장된 메타데이터 사용](#)
- [스트리밍 메타데이터 제한](#)

Kinesis 비디오 스트림에 메타데이터 추가

Kinesis 비디오 스트림에 추가하는 메타데이터는 키-값 쌍으로 구현되는 MKV 태그로 모델링됩니다.

메타데이터는 스트림 내에서 이벤트를 표시하는 것처럼 일시적일 수도 있고 특정 이벤트가 발생하는 조각을 식별하는 것처럼 지속적일 수도 있습니다. 영구 메타데이터 항목은 취소될 때까지 남아 있으며 각 연속 프래그먼트에 적용됩니다.

Note

[생산자 라이브러리](#)을(를) 사용하여 추가되는 메타데이터 항목은 [the section called "TagStream"](#), [the section called "UntagStream"](#) 및 [the section called "ListTagsForStream"](#)와 (과) 함께 실행되는 스트림 수준 태깅 API와는 구별됩니다.

스트리밍 메타데이터 API

생산자 SDK에서 다음 작업을 사용하여 메타데이터 스트리밍을 실행할 수 있습니다.

PIC

```
PUBLIC_API STATUS putKinesisVideoFragmentMetadata(STREAM_HANDLE streamHandle,
    PCHAR name,
    PCHAR value,
    BOOL persistent);
```

C++ 생산자 SDK

```
/**
 * Appends a "tag" or metadata - a key/value string pair into the stream.
 */
bool putFragmentMetadata(const std::string& name, const std::string& value, bool
    persistent = true);
```

Java 생산자 SDK

Java Producer SDK를 사용하여 다음을 `MediaSource` 사용하여 `MediaSourceSink.onCodecPrivateData` 메타데이터를 추가할 수 있습니다.

```
void onFragmentMetadata(final @NonNull String metadataName, final @NonNull String
    metadataValue, final boolean persistent)
```



```
throws KinesisVideoException;
```

영구 및 비영구 메타데이터

비지속성 메타데이터의 경우, 여러 메타데이터 항목을 동일한 이름으로 추가할 수 있습니다. 생산자 SDK는 메타데이터 항목이 다음 조각에 추가될 때까지 메타데이터 대기열에서 메타데이터 항목을 수집합니다. 메타데이터 항목이 스트림에 적용됨에 따라 메타데이터 대기열은 삭제됩니다. 메타데이터를 반복하려면 `putKinesisVideoFragmentMetadata` 또는 `putFragmentMetadata`를 다시 호출합니다.

지속성 메타데이터의 경우, 생산자 SDK가 비지속성 메타데이터의 경우와 동일한 방식으로 메타데이터 대기열에서 메타데이터 항목을 수집합니다. 하지만 메타데이터 항목이 다음 프래그먼트 앞에 추가되어도 대기열에서 제거되지 않습니다.

`true`로 설정된 `persistent`로 `putKinesisVideoFragmentMetadata` 또는 `putFragmentMetadata`를 호출하면 다음과 같은 동작으로 이어집니다.

- API를 호출하면 메타데이터 항목을 대기열로 가져옵니다. 메타데이터는 해당 항목이 대기열에 있는 동안 MKV 태그로 모든 조각에 추가됩니다.
- 같은 이름과 다른 값으로 API를 이전에 추가된 메타데이터 항목으로서 호출하면 해당 항목을 덮어쓰기합니다.
- API를 비어 있는 값으로 호출하면 메타데이터 대기열에서 메타데이터 항목을 제거(취소)합니다.

Kinesis 비디오 스트림에 내장된 메타데이터 사용

Kinesis 비디오 스트림의 메타데이터를 사용하려면 다음 구현을 사용하십시오. `MkvTagProcessor`

```
public interface MkvTagProcessor {
    default void process(MkvTag mkvTag, Optional<FragmentMetadata>
currentFragmentMetadata) {
        throw new NotImplementedException("Default
FragmentMetadataVisitor.MkvTagProcessor");
    }
    default void clear() {
        throw new NotImplementedException("Default
FragmentMetadataVisitor.MkvTagProcessor");
    }
}
```

```
}

```

이 인터페이스는 [Kinesis 비디오 스트림 파서 라이브러리](#)의 [FragmentMetadataVisitor](#) 클래스에서 찾을 수 있습니다.

FragmentMetadataVisitor 클래스에는 MkvTagProcessor의 구현이 포함되어 있습니다.

```
public static final class BasicMkvTagProcessor implements
    FragmentMetadataVisitor.MkvTagProcessor {
    @Getter
    private List<MkvTag> tags = new ArrayList<>();

    @Override
    public void process(MkvTag mkvTag, Optional<FragmentMetadata>
currentFragmentMetadata) {
        tags.add(mkvTag);
    }

    @Override
    public void clear() {
        tags.clear();
    }
}

```

KinesisVideoRendererExample 클래스에는 BasicMkvTagProcessor를 사용하는 방법의 예가 포함되어 있습니다. 다음 예에서는 BasicMkvTagProcessor를 애플리케이션의 MediaProcessingArguments에 추가합니다.

```
if (renderFragmentMetadata) {
    getMediaProcessingArguments =
    KinesisVideoRendererExample.GetMediaProcessingArguments.create(
        Optional.of(new FragmentMetadataVisitor.BasicMkvTagProcessor()));
}

```

조각 메타데이터가 도착하면 BasicMkvTagProcessor.process 메서드가 호출됩니다. GetTags로 누적된 메타데이터를 검색할 수 있습니다. 단일 메타데이터 항목을 검색하려면 먼저 clear 호출하여 수집된 메타데이터를 지운 다음 메타데이터 항목을 다시 검색하십시오.

스트리밍 메타데이터 제한

Kinesis 비디오 스트림에 스트리밍 메타데이터를 추가할 때 적용되는 제한에 [the section called “프레그먼트 메타데이터 할당량”](#) 대한 자세한 내용은 을 참조하십시오.

Kinesis Video Streams 데이터 모델

[생산자 라이브러리](#) 및 [스트림 구문 분석기 라이브러리](#)가 비디오 데이터와 함께 정보를 포함하는 것을 지원하는 형식으로 비디오 데이터를 전송하고 수신합니다. 이 형식은 Matroska(MKV) 사양을 기반으로 합니다.

[MKV 형식](#)은 미디어 데이터에 적용되는 공개 사양입니다. Amazon Kinesis Video Streams 개발자 안내서의 모든 라이브러리 및 코드 예제는 MKV 형식으로 데이터를 보내거나 받습니다.

는 StreamDefinition 및 Frame 유형을 [Kinesis 비디오 스트림 프로듀서 라이브러리](#) 사용하여 MKV 스트림 헤더, 프레임 헤더 및 프레임 데이터를 생성합니다.

전체 MKV 사양에 관한 내용은 [Matroska 사양](#)을 참조하십시오.

다음 단원에서는 [C++ 생산자 라이브러리](#)에 의해 생성되는 MKV 형식 데이터의 구성 요소를 설명합니다.

주제

- [스트림 헤더 요소](#)
- [스트림 트랙 데이터](#)
- [프레임 헤더 요소](#)
- [MKV 프레임 데이터](#)

스트림 헤더 요소

다음과 같은 MKV 헤더 요소는 StreamDefinition에 의해 사용됩니다(StreamDefinition.h에 정의되어 있음).

Element	Description	일반적인 값
stream_name	Kinesis 비디오 스트림의 이름에 해당합니다.	my-stream
retention_period	Kinesis Video Streams가 스트림 데이터를 유지하는 기간 (시간)입니다. 데이터를 보존하지 않는 0 스트림에 대해 지정하십시오.	24

Element	Description	일반적인 값
태그	사용자 데이터의 키-값 모음입니다. 이 데이터는 AWS Management Console에 표시되며, 클라이언트 애플리케이션이 읽어 필터링하거나 스트림에 관한 정보를 얻을 수 있습니다.	
kms_key_id	사용자 정의 AWS KMS 키가 있는 경우 스트림의 데이터를 암호화하는 데 사용자 정의 키가 사용됩니다. 없는 경우 Kinesis에서 제공한 키 () 로 데이터가 암호화됩니다. aws/kinesis-video	01234567-89ab-cdef-0123-456789ab
streaming_type	현재 유효한 유일한 스트리밍은 STREAMING_TYPE_REALTIME 입니다.	STREAMING_TYPE_REALTIME
content_type	사용자 정의 콘텐츠 유형입니다. 콘솔에서 재생할 비디오 데이터 스트리밍의 경우 콘텐츠 유형이 video/h264 이어야 합니다.	video/h264
max_latency	이 값은 현재 사용되지 않으므로 0으로 설정해야 합니다.	0
fragment_duration	이는 조각 길이의 추정치로서 최적화에 사용됩니다. 실제 조각 지속 시간은 스트리밍 데이터에 의해 결정됩니다.	2

Element	Description	일반적인 값
timecode_scale	<p>프레임 타임스탬프가 사용하는 스케일을 나타냅니다. 기본값은 1밀리초입니다. 0을 지정하면 기본값 1밀리초도 할당됩니다. 이 값은 100나노초에서 1초 사이일 수 있습니다.</p> <p>자세한 내용은 Matroska TimecodeScale 설명서를 참조하십시오.</p>	
key_frame_fragmentation	true인 경우, 키프레임 수신 시 스트림이 새 클러스터를 시작합니다.	true
frame_timecodes	<p>경우 true Kinesis Video Streams는 수신된 프레임의 프레젠테이션 타임스탬프 (pts) 및 디코딩 타임스탬프 (dts) 값을 사용합니다. 경우 false Kinesis Video Streams는 프레임이 수신될 때 시스템에서 생성한 시간 값과 함께 해당 프레임을 스탬프합니다.</p>	true
absolute_fragment_time	true인 경우 클러스터 타임코드가 절대 시간(예: 생산자 시스템 시계의 값)을 사용하는 것으로 해석됩니다. false인 경우, 클러스터 타임코드가 스트림 시작 시간에 상대적인 것으로 해석됩니다.	true

Element	Description	일반적인 값
fragment_acks	경우 true Kinesis Video Streams가 데이터를 수신할 때 승인 (ACK) 이 전송됩니다. ACK는 KinesisVideoStreamFragmentAck 또는 KinesisVideoStreamParseFragmentAck 콜백을 사용하여 수신할 수 있습니다.	true
restart_on_error	스트림 오류가 발생한 후 스트림이 전송을 다시 시작해야 하는지 여부를 나타냅니다.	true
nal_adaptation_flags	콘텐츠에 NAL(Network Abstraction Layer) 적응 또는 코덱 프라이빗 데이터가 존재하는지를 나타냅니다. 유효한 플래그에는 NAL_ADAPTATION_ANNEXB_NALS 및 NAL_ADAPTATION_ANNEXB_CPD_NALS 가 있습니다.	NAL_ADAPTATION_ANNEXB_NALS
frame_rate	콘텐츠 프레임 속도의 추정치입니다. 이 값은 최적화에 사용되며, 실제 프레임 속도는 수신 데이터의 속도에 의해 결정됩니다. 0을 지정하면 기본값 24가 할당됩니다.	24

Element	Description	일반적인 값
avg_bandwidth_bps	콘텐츠 대역폭의 추정치 (Mbps). 이 값은 최적화에 사용되며, 실제 속도는 수신 데이터의 대역폭에 따라 결정됩니다. 예를 들어, 25FPS로 실행되는 720p 해상도 비디오 스트림의 경우 평균 대역폭을 5Mbps로 예상할 수 있습니다.	5
buffer_duration	콘텐츠가 생산자에서 버퍼링되는 기간입니다. 네트워크 지연 시간이 짧으면 이 값을 줄일 수 있습니다. 네트워크 지연 시간이 긴 경우 이 값을 늘리면 더 작은 버퍼에 프레임을 넣는 할당이 실패하여 프레임이 전송되기 전에 삭제되는 것을 방지할 수 있습니다.	
replay_duration	연결이 끊긴 경우 비디오 데이터 스트림이 “되감기”되는 시간입니다. 연결 끊김으로 인한 프레임 손실이 우려되지 않는 경우 이 값은 0이 될 수 있습니다. 사용 중인 응용 프로그램에서 중복 프레임을 제거할 수 있는 경우 이 값을 높일 수 있습니다. 이 값은 버퍼 지속 시간보다 작아야 합니다. 그렇지 않으면 버퍼 지속 시간이 사용됩니다.	
connection_staleness	수신 데이터가 없을 경우 연결이 유지되는 기간입니다.	

Element	Description	일반적인 값
codec_id	콘텐츠에 사용되는 코덱입니다. 자세한 내용은 Matroska 사양의 CodecID 를 참조하십시오.	V_MPEG2
track_name	사용자가 정의하는 트랙 이름입니다.	my_track
codecPrivateData	프레임 데이터를 디코딩하는 목적으로 사용되며 인코더에 의해 제공되는 데이터 (예: 프레임 픽셀의 높이 및 폭)로서, 다수의 다운스트림 소비자가 필요로 합니다. C++ 생산자 라이브러리 에서는 MkvStatics.cpp 의 gMkvTrackVideoBits 어레이에 프레임의 픽셀 폭과 높이가 포함되어 있습니다.	
codecPrivateData크기	codecPrivateData 파라미터에 있는 데이터의 크기입니다.	
track_type	스트림에 대한 트랙의 유형입니다.	MKV_TRACK_INFO_TYPE_AUDIO 또는 MKV_TRACK_INFO_TYPE_VIDEO
segment_uuid	사용자 정의 세그먼트 uuid(16 바이트)입니다.	
default_track_id	트랙에 대한 고유한 0이 아닌 숫자입니다.	1

스트림 트랙 데이터

다음과 같은 MKV 트랙 요소는 StreamDefinition에 의해 사용됩니다(StreamDefinition.h에 정의되어 있음).

Element	Description	일반적인 값
track_name	사용자 정의 트랙 이름입니다. 예를 들어, 오디오 트랙의 경우 "audio"입니다.	audio
codec_id	트랙에 대한 코덱 ID입니다. 예를 들어, 오디오 트랙의 경우 "A_AAC"입니다.	A_AAC
cpd	프레임 데이터의 디코딩에 사용되는, 인코더에서 제공하는 데이터입니다. 이 데이터에는 다수의 다운스트림 소비자가 필요로 하는 프레임 픽셀의 폭 및 높이가 포함될 수 있습니다. C++ 프로듀서 MkvStatics 라이브러리에서.cpp의 gMkvTrack VideoBits 배열에는 프레임의 픽셀 너비와 높이가 포함됩니다.	
cpd_size	파라미터의 데이터 크기. codecPrivateData	
track_type	트랙의 유형입니다. 예를 들어, 오디오의 경우 MKV_TRACK_INFO_TYPE_AUDIO의 열거 값을 사용할 수 있습니다.	MKV_TRACK_INFO_TYPE_AUDIO

프레임 헤더 요소

Frame은 다음 MKV 헤더 요소를 사용합니다(KinesisVideoPic 패키지의 mkvgen/Include.h에 정의).

- 프레임 인덱스: 단순 증가 값.
- 플래그: 프레임의 유형 유효한 값은 다음과 같습니다.
 - FRAME_FLAGS_NONE
 - FRAME_FLAG_KEY_FRAME: key_frame_fragmentation가 스트림에 설정되어 있는 경우, 키 프레임이 새 조각을 시작합니다.
 - FRAME_FLAG_DISCARDABLE_FRAME: 디코딩이 느린 경우 이 프레임을 무시할 수 있음을 디코더에 알립니다.
 - FRAME_FLAG_INVISIBLE_FRAME: 이 블록의 지속 시간은 0입니다.
- 디코딩 타임스탬프: 이 프레임이 디코딩된 시점의 타임스탬프입니다. 이전 프레임이 이 프레임을 사용하여 디코딩하는 경우 이 타임스탬프가 이전 프레임보다 빠를 수 있습니다. 이 값은 조각의 시작을 기준으로 합니다.
- 프레젠테이션 타임스탬프: 이 프레임이 표시되는 시점의 타임스탬프입니다. 이 값은 조각의 시작을 기준으로 합니다.
- 지속 시간: 프레임의 재생 시간입니다.
- 크기 프레임 데이터의 바이트 단위 크기입니다.

MKV 프레임 데이터

frame.frameData에 있는 데이터는 사용되는 인코딩 스키마에 따라 프레임에 대한 미디어 데이터만 포함할 수도 있고, 더 중첩된 헤더 정보를 포함할 수도 있습니다. 에 표시되려면 데이터가 [H.264](#) 코덱으로 인코딩되어야 AWS Management Console 하지만 Kinesis Video Streams는 어떤 형식으로든 시간 직렬화된 데이터 스트림을 수신할 수 있습니다.

아마존 키네시스 비디오 스트림 시작하기

이 섹션에서는 Amazon Kinesis Video Streams에서 다음 작업을 수행하는 방법을 설명합니다.

- 관리자를 AWS 계정 설정하고 생성하십시오 (아직 생성하지 않은 경우).
- Kinesis 비디오 스트림을 생성합니다.
- 카메라에서 Kinesis 비디오 스트림으로 데이터를 전송하고 콘솔에서 미디어를 확인합니다.

Amazon Kinesis Video Streams를 처음 사용하는 경우 먼저 [Kinesis Video Streams: 작동 방식](#) 읽어보는 것이 좋습니다.

Note

시작하기 샘플을 따라해도 요금이 부과되지 않습니다. AWS 계정해당 지역의 데이터 요금은 [Amazon Kinesis Video Streams](#) 요금을 참조하십시오.

주제

- [계정을 설정합니다.](#)
- [Kinesis 비디오 스트림 생성](#)
- [Amazon Kinesis 비디오 스트림으로 데이터 전송](#)
- [미디어 데이터 사용](#)

계정을 설정합니다.

Amazon Kinesis Video Streams를 처음 사용하기 전에 다음 작업을 완료하십시오.

주제

- [가입하여 AWS 계정](#)
- [관리자 액세스 권한이 있는 사용자 생성](#)
- [키 생성 AWS 계정](#)

가입하여 AWS 계정

계정이 없는 경우 다음 단계를 완료하여 계정을 만드세요. AWS 계정

가입하려면 AWS 계정

1. <https://portal.aws.amazon.com/billing/signup>을 여세요.
2. 온라인 지시 사항을 따르세요.

등록 절차 중에는 전화를 받고 키패드로 인증 코드를 입력하는 과정이 있습니다.

에 AWS 계정가입하면 AWS 계정 루트 사용자a가 생성됩니다. 루트 사용자에게는 계정의 모든 AWS 서비스 및 리소스 액세스 권한이 있습니다. 보안 모범 사례는 사용자에게 관리 액세스 권한을 할당하고, 루트 사용자만 사용하여 [루트 사용자 액세스 권한이 필요한 작업](#)을 수행하는 것입니다.

AWS 가입 절차가 완료된 후 확인 이메일을 보냅니다. 언제든지 <https://aws.amazon.com/>으로 가서 내 계정(My Account)을 선택하여 현재 계정 활동을 보고 계정을 관리할 수 있습니다.

관리자 액세스 권한이 있는 사용자 생성

등록한 AWS 계정후에는 일상적인 작업에 루트 사용자를 사용하지 않도록 관리 사용자를 보호하고 AWS IAM Identity Center활성화하고 생성하십시오 AWS 계정 루트 사용자.

보안을 유지하세요. AWS 계정 루트 사용자

1. 루트 사용자를 선택하고 AWS 계정 이메일 주소를 입력하여 계정 [AWS Management Console](#)소유자로 로그인합니다. 다음 페이지에서 비밀번호를 입력합니다.

루트 사용자를 사용하여 로그인하는 데 도움이 필요하면AWS 로그인 사용 설명서의 [루트 사용자 로 로그인](#)을 참조하세요.

2. 루트 사용자의 다중 인증(MFA)을 활성화합니다.

지침은 IAM [사용 설명서의 AWS 계정 루트 사용자 \(콘솔\)에 대한 가상 MFA 디바이스 활성화](#)를 참조하십시오.

관리자 액세스 권한이 있는 사용자 생성

1. IAM Identity Center를 활성화합니다.

지침은 AWS IAM Identity Center 사용 설명서의 [AWS IAM Identity Center 설정](#)을 참조하세요.

2. IAM Identity Center에서 사용자에게 관리 액세스 권한을 부여합니다.

를 ID 소스로 사용하는 방법에 대한 자습서는 사용 [설명서의 기본값으로 IAM Identity Center 디렉터리 사용자 액세스 구성](#)을 참조하십시오. IAM Identity Center 디렉터리 AWS IAM Identity Center

관리 액세스 권한이 있는 사용자로 로그인

- IAM Identity Center 사용자로 로그인하려면 IAM Identity Center 사용자를 생성할 때 이메일 주소로 전송된 로그인 URL을 사용합니다.

IAM Identity Center 사용자를 사용하여 [로그인하는 데 도움이 필요하면 사용 설명서의 AWS 액세스 포털에 로그인](#)을 참조하십시오. AWS 로그인

추가 사용자에게 액세스 권한 할당

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

지침은 AWS IAM Identity Center 사용 설명서의 [Create a permission set](#)를 참조하세요.

2. 사용자를 그룹에 할당하고, 그룹에 Single Sign-On 액세스 권한을 할당합니다.

지침은 AWS IAM Identity Center 사용 설명서의 [Add groups](#)를 참조하세요.

키 생성 AWS 계정

프로그래밍 방식으로 Amazon Kinesis Video Streams에 액세스하려면 AWS 계정 키가 필요합니다.

AWS 계정 키를 생성하려면 다음과 같이 하십시오.

1. <https://console.aws.amazon.com/iam/>에서 AWS Management Console 로그인하고 IAM 콘솔을 엽니다.
2. 탐색 모음에서 사용자를 선택한 후 관리자 사용자를 선택합니다.
3. 보안 인증 탭을 선택한 후 액세스 키 생성을 선택합니다.
4. 액세스 키 ID를 기록합니다. 보안 액세스 키에서 보기를 선택합니다. 보안 액세스 키를 기록합니다.

Kinesis 비디오 스트림 생성

이 섹션에서는 Kinesis 비디오 스트림을 생성하는 방법을 설명합니다.

이 단원에는 다음 절차가 포함됩니다.

- [the section called “콘솔을 사용하여 비디오 스트림을 생성합니다.”](#)
- [the section called “를 사용하여 비디오 스트림을 생성하십시오. AWS CLI”](#)

콘솔을 사용하여 비디오 스트림을 생성합니다.

1. 에서 콘솔을 엽니다 <https://console.aws.amazon.com/kinesisvideo/home>.
2. 비디오 스트림 페이지에서 비디오 스트림 생성을 선택합니다.
3. 새 비디오 스트림 만들기 페이지에서 스트림 이름을 *YourStreamName* 입력합니다. 기본 구성 버튼을 선택된 상태로 둡니다.
4. 비디오 스트림 생성을 선택합니다.
5. Amazon Kinesis Video Streams에서 스트림을 생성한 후 페이지의 세부 정보를 *YourStreamName* 검토하십시오.

를 사용하여 비디오 스트림을 생성하십시오. AWS CLI

1. AWS CLI 설치 및 구성이 되었는지 확인하십시오. 자세한 내용은 [AWS Command Line Interface](#) 설명서를 참조하세요.
2. AWS CLI에서 다음 Create-Stream 명령을 실행합니다.

```
aws kinesisvideo create-stream --stream-name "YourStreamName" --data-retention-in-hours 24
```

다음과 유사한 응답이 반환될 것입니다.

```
{
  "StreamARN": "arn:aws:kinesisvideo:us-west-2:123456789012:stream/YourStreamName/123456789012"
}
```

Amazon Kinesis 비디오 스트림으로 데이터 전송

이 섹션에서는 카메라의 미디어 데이터를 이전 섹션에서 만든 Kinesis 비디오 스트림으로 보내는 방법을 설명합니다. 이 섹션에서는 [C++ 생산자 라이브러리](#)를 [스트리머 플러그인 - kvssink](#) 플러그인으로 사용합니다.

다양한 운영 체제의 다양한 디바이스에서 미디어를 전송하기 위해 이 자습서에서는 Kinesis Video Streams C++ 프로듀서 라이브러리와 카메라 및 기타 미디어 소스에 대한 액세스를 표준화하는 오픈 소스 미디어 프레임워크인 GStreamer를 사용합니다.

주제

- [SDK와 샘플을 빌드하세요.](#)
- [샘플을 실행하여 Kinesis Video Streams에 미디어를 업로드합니다.](#)
- [승인 객체를 검토하세요.](#)

SDK와 샘플을 빌드하세요.

컴퓨터에서 또는 내부에서 SDK와 샘플을 빌드할 수 있습니다. AWS Cloud9 아래의 적절한 절차를 따르세요.

Build on your computer

[readme 파일의](#) 지침을 사용하여 제작자 라이브러리와 샘플 애플리케이션을 빌드하십시오.

여기에는 다음이 포함됩니다.

- 종속성 설치
- 리포지토리 복제
- CMake를 사용하여 메이크파일 생성
- make를 사용하여 바이너리 파일 빌드하기

Build in AWS Cloud9

다음 절차에 따라 Kinesis Video AWS Cloud9 Streams에 업로드하십시오. 컴퓨터에 아무것도 다운로드할 필요가 없습니다.

1. AWS Management Console에서 엽니다 [AWS Cloud9](#).

환경 만들기를 선택합니다.

2. 환경 만들기 화면에서 다음을 완료하십시오.

- 이름 - 새 환경의 이름을 입력합니다.
- 플랫폼 - 우분투 서버 22.04 LTS를 선택합니다.

다른 필드는 기본 선택 사항을 그대로 둘 수 있습니다.

3. 환경이 생성되면 Cloud9 IDE 열에서 열기를 선택합니다.

화면 하단 중간 영역에 표시됩니다. Admin:~/environment \$ 이 터미널은 AWS Cloud9 (아마존 EC2) 터미널입니다.

Note

실수로 터미널을 닫은 경우 창, 새 터미널을 선택합니다.

터미널에서 다음 명령을 실행하여 볼륨을 20GiB로 변경합니다.

a. 스크립트를 다운로드합니다.

```
wget https://awsj-iot-handson.s3-ap-northeast-1.amazonaws.com/kvs-workshop/resize_volume.sh
```

b. 스크립트에 실행 권한을 부여합니다.

```
chmod +x resize_volume.sh
```

c. 스크립트를 실행합니다.

```
./resize_volume.sh
```

4. 고급 패키징 도구 (APT) 를 통해 설치 또는 업데이트할 수 있는 모든 소프트웨어에 대한 최신 정보를 가져오십시오.

이 명령은 소프트웨어 자체를 업데이트하지는 않지만 사용 가능한 최신 버전이 무엇인지 시스템이 알 수 있도록 합니다.


```
sudo apt-get update
```

5. C++ 프로듀서 SDK 종속 항목을 설치합니다.

```
sudo apt-get install -y cmake m4 git build-essential pkg-config libssl-dev
libcurl4-openssl-dev \
liblog4cplus-dev libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev \
gstreamer1.0-plugins-base-apps gstreamer1.0-plugins-bad gstreamer1.0-plugins-
good \
gstreamer1.0-plugins-ugly gstreamer1.0-tools
```

6. git을 사용하여 C++ 프로듀서 SDK를 복제합니다.

```
git clone https://github.com/aws-labs/amazon-kinesis-video-streams-producer-sdk-
cpp.git
```

7. 빌드 디렉터리를 준비하세요.

```
cd amazon-kinesis-video-streams-producer-sdk-cpp
mkdir build
cd build
```

8. CMake를 사용하여 메이크파일을 생성하세요.

```
cmake .. -DBUILD_GSTREAMER_PLUGIN=TRUE -DBUILD_DEPENDENCIES=OFF
```

예상 출력의 끝은 다음과 같습니다.

```
-- Build files have been written to: /home/ubuntu/environment/amazon-kinesis-
video-streams-producer-sdk-cpp/build
```

9. make를 사용하여 SDK와 샘플 애플리케이션을 컴파일하고 최종 실행 파일을 빌드할 수 있습니다.

```
make
```

예상 출력의 끝은 다음과 같습니다.

```
[100%] Linking CXX executable kvs_gstreamer_file_uploader_sample
```

```
[100%] Built target kvs_gstreamer_file_uploader_sample
```

10. 샘플 파일이 빌드되었는지 확인합니다. 현재 디렉터리에 있는 파일 목록:

```
ls
```

다음 파일이 있는지 확인합니다.

- kvs_gstreamer_sample
- libgstkvssink.so

11. (선택 사항) GST_PLUGIN_PATH 환경 변수 설정을 셸의 시작 스크립트에 추가할 수 있습니다. 이렇게 하면 새 터미널 세션 중에 GST_PLUGIN_PATH가 올바르게 설정됩니다. 에서 셸의 AWS Cloud9시작 스크립트는 다음과 같습니다. ~/.bashrc

다음 명령을 실행하여 셸의 시작 스크립트 끝에 명령을 추가합니다.

```
echo "export GST_PLUGIN_PATH=~/environment/amazon-kinesis-video-streams-producer-sdk-cpp/build" >> ~/.bashrc
```

다음을 입력하여 셸의 시작 스크립트를 실행합니다.

```
source ~/.bashrc
```

GST_PLUGIN_PATH가 설정되었는지 확인합니다.

```
echo $GST_PLUGIN_PATH
```

출력을 올바르게 설정하면 다음과 같은 출력이 표시됩니다. 출력이 비어 있으면 환경 변수가 제대로 설정되지 않은 것입니다.

```
/home/ubuntu/environment/amazon-kinesis-video-streams-producer-sdk-cpp/build
```

샘플을 실행하여 Kinesis Video Streams에 미디어를 업로드합니다.

샘플 애플리케이션은 IMDS 자격 증명을 지원하지 않습니다. 터미널에서 IAM 사용자 또는 역할과 스트림이 위치한 지역의 AWS 자격 증명을 내보내십시오.

```
export AWS_ACCESS_KEY_ID=YourAccessKey
export AWS_SECRET_ACCESS_KEY=YourSecretKey
export AWS_DEFAULT_REGION=YourAWSRegion
```

임시 AWS 자격 증명을 사용하는 경우 세션 토큰도 보내주세요.

```
export AWS_SESSION_TOKEN=YourSessionToken
```

.mp4 files

샘플 .mp4 비디오를 다운로드하여 Kinesis Video Streams에 업로드하십시오.

```
wget https://awsj-iot-handson.s3-ap-northeast-1.amazonaws.com/kvs-workshop/sample.mp4
```

비디오 사양:

- 해상도 - 1280 x 720 픽셀
- 프레임 속도 - 초당 30프레임
- 지속 시간 - 14.0초
- 비디오 인코딩 - H.264, 트랙 1에 있음
- 키프레임 - 3초마다 조각 지속 시간 (사진 그룹 (GoP) 크기라고도 함) 이 3초이고 최종 조각의 길이는 2초입니다.

이전에 만든 스트림의 이름을 사용하여 다음 명령을 실행합니다. 아직 스트림을 만들지 않았다면 [참조하십시오](#) [the section called “Kinesis 비디오 스트림 생성”](#).

```
./kvs_gstreamer_sample YourStreamName ./sample.mp4
```

Sample video from GStreamer

다음 명령어를 사용하여 GStreamer를 사용하여 비디오를 생성하십시오.

GStreamer에게 GStreamer 플러그인을 어디에 설치해야 하는지 알려주세요. kvssink 빌드 디렉터리에서 파일이 들어 있는 폴더의 경로를 지정합니다. libgstkvssink.so

빌드 디렉터리에서 다음 명령을 실행합니다.

```
export GST_PLUGIN_PATH=`pwd`
```

이 GStreamer 파이프라인은 640x480 픽셀의 해상도로 초당 10프레임으로 실행되는 표준 테스트 패턴을 사용하여 라이브 테스트 비디오 스트림을 생성합니다. 현재 시스템 시간 및 날짜를 표시하는 오버레이가 추가되었습니다. 그러면 비디오가 H.264 형식으로 인코딩되고 최대 10프레임마다 키프레임이 생성되므로 프레임 지속 시간 (사진 그룹 (GoP) 크기라고도 함)은 1초가 됩니다. kvssinkH.264로 인코딩된 비디오 스트림을 가져와 Matroska (MKV) 컨테이너 형식으로 패키징한 다음 Kinesis 비디오 스트림에 업로드합니다.

다음 명령을 실행합니다:

```
gst-launch-1.0 -v videotestsrc is-live=true \
! video/x-raw,framerate=10/1,width=640,height=480 \
! clockoverlay time-format="%a %B %d, %Y %I:%M:%S %p" \
! x264enc bframes=0 key-int-max=10 \
! h264parse \
! kvssink stream-name="YourStreamName"
```

GStreamer 파이프라인을 중지하려면 터미널 창을 선택하고 Ctrl+C를 누릅니다.

Note

GStreamer 플러그인을 사용하여 카메라 또는 USB 카메라의 RTSP 스트림에서 비디오를 스트리밍하는 방법에 대한 자세한 내용은 [을 참조하십시오. 예: Kinesis Video Streams 프로듀서 SDK GStreamer 플러그인 - kvssink](#)

승인 객체를 검토하세요.

업로드 중에 Kinesis Video Streams는 업로드를 수행하는 클라이언트에 확인 객체를 다시 보냅니다. 명령 출력에 이러한 내용이 인쇄되어 있어야 합니다. 예는 다음과 같습니다.

```
{"EventType":"PERSISTED","FragmentTimecode":1711124585823,"FragmentNumber":"1234567890123456789"}
```

확인이 사실이라면 Kinesis Video Streams가 검색PERSISTED, 분석 및 장기 저장을 위해 이 미디어 청크를 안정적으로 저장하고 암호화했음을 의미합니다. EventType

승인에 대한 자세한 내용은 [을 참조하십시오. the section called "PutMedia"](#)

미디어 데이터 사용

콘솔에서 보거나 하이퍼텍스트 라이브 스트리밍 (HLS) 을 사용하여 스트림에서 미디어 데이터를 읽는 응용 프로그램을 만들어 미디어 데이터를 사용할 수 있습니다.

콘솔에서 미디어 보기

다른 브라우저 탭에서 를 엽니다 AWS Management Console. Kinesis Video Streams 대시보드에서 비디오 스트림을 [선택합니다](#).

스트림 목록에서 스트림 이름을 선택합니다. 필요한 경우 검색 창을 사용하세요.

미디어 재생 섹션을 확장하세요. 동영상이 아직 업로드 중이면 동영상이 표시됩니다. 업로드가 완료되면 왼쪽 이중 화살표를 선택합니다.

HLS를 사용하여 미디어 데이터를 소비하십시오.

HLS를 사용하여 Kinesis 비디오 스트림의 데이터를 사용하는 클라이언트 애플리케이션을 생성할 수 있습니다. HLS로 미디어 데이터를 사용하는 애플리케이션을 생성하는 방법에 대한 자세한 내용은 [the section called “비디오 재생”](#) 단원을 참조하십시오.

아마존 Kinesis Video Streams 엣지 에이전트

Amazon Kinesis Video Streams는 고객 온프레미스의 IP 카메라에 연결할 수 있는 효율적이고 비용 효율적인 방법을 제공합니다. Amazon Kinesis Video Streams Edge Agent를 사용하면 카메라의 비디오를 로컬로 녹화 및 저장하고 장기 저장, 재생 및 분석 처리를 위해 고객이 정의한 일정에 따라 클라우드로 비디오를 스트리밍할 수 있습니다.

Note

Amazon Kinesis Video Streams Edge 에이전트에 액세스하려면 [이 간단한 양식](#)을 작성하십시오.

Amazon Kinesis Video Streams Edge 에이전트를 다운로드하여 온프레미스 엣지 컴퓨팅 디바이스에 배포할 수 있습니다. Amazon EC2 인스턴스에서 실행되는 Docker 컨테이너에 쉽게 배포할 수도 있습니다. 배포 후에는 Amazon Kinesis Video Streams API를 사용하여 비디오 녹화 및 클라우드 업로드 구성을 업데이트할 수 있습니다. 이 기능은 RTSP 프로토콜을 통해 스트리밍할 수 있는 모든 IP 카메라에서 작동합니다. 카메라에 펌웨어를 추가로 배포할 필요는 없습니다.

Amazon Kinesis Video Streams Edge 에이전트에는 다음과 같은 설치 서비스가 제공됩니다.

- AWS IoT Greengrass V2 구성 요소로: Amazon Kinesis Video Streams Edge 에이전트를 AWS IoT Greengrass 모든 인증된 디바이스에 구성 AWS IoT Greengrass 요소로 설치할 수 있습니다. 자세한 AWS IoT Greengrass 알아보려면 [AWS IoT Greengrass Version 2 개발자 안내서](#)를 참조하십시오.
- 켜기 AWS Snowball Edge: Snowball Edge 디바이스에서 Amazon Kinesis Video Streams Edge 에이전트를 실행할 수 있습니다. 자세한 내용은 [AWS Snowball Edge 개발자 안내서](#)를 참조하십시오.
- 네이티브 AWS IoT 배포 시: Amazon Kinesis Video Streams Edge 에이전트를 모든 컴퓨팅 인스턴스에 기본적으로 설치할 수 있습니다. Edge SDK는 엣지를 통해 관리하는 [AWS IoT Core](#)에 사용됩니다. [the section called "Amazon Kinesis Video Streams"](#)

Amazon Kinesis Video Streams Edge 에이전트를 시작하려면 아래의 적절한 절차를 계속 진행하십시오.

주제

- [아마존 키네시스 비디오 스트림 엣지 에이전트 API 오퍼레이션](#)
- [아마존 Kinesis Video Streams 엣지 에이전트 모니터링](#)

- [Amazon Kinesis Video Streams 엣지 에이전트를 AWS IoT Greengrass 비모드에서 실행](#)
- [Amazon Kinesis Video Streams 엣지 에이전트를 다음 위치에 배포하십시오. AWS IoT Greengrass](#)
- [아마존 Kinesis Video Streams 엣지 에이전트 FAQ](#)

아마존 키네시스 비디오 스트림 엣지 에이전트 API 오퍼레이션

다음 API 작업을 사용하여 Amazon Kinesis Video Streams Edge 에이전트를 구성하십시오.

- [the section called “StartEdgeConfigurationUpdate”](#)
- [the section called “DescribeEdgeConfiguration”](#)
- [the section called “DeleteEdgeConfiguration”](#)
- [the section called “ListEdgeAgentConfigurations”](#)

아마존 Kinesis Video Streams 엣지 에이전트 모니터링

Amazon Kinesis Video Streams Edge 에이전트를 모니터링하려면 을 참조하십시오. [the section called “를 사용하여 Amazon Kinesis Video Streams 엣지 에이전트를 모니터링하기 CloudWatch”](#)

Amazon Kinesis Video Streams 엣지 에이전트를 AWS IoT Greengrass 비모드에서 실행

다음 단계에 따라 MQTT를 사용하여 Amazon Kinesis Video Streams Edge 에이전트를 AWS IoT 독립 실행형 배포로 실행하십시오.

주제

- [1단계: 디바이스에 필요한 종속 항목을 설치합니다.](#)
- [2단계: IP 카메라 RTSP URL을 위한 Amazon Kinesis Video Streams AWS Secrets Manager 및 리 소스 생성](#)
- [3단계: IAM 권한 정책 생성](#)
- [4단계: IAM 역할 생성](#)
- [5단계: 역할 별칭 생성 AWS IoT](#)
- [6단계: AWS IoT 정책 생성](#)
- [7단계: AWS IoT 사물 생성 및 자격 증명 획득 AWS IoT Core](#)

- [8단계: Amazon Kinesis Video Streams Edge 에이전트의 구축 및 실행](#)
- [9단계: \(선택 사항\) 디바이스에 CloudWatch 에이전트 설치](#)
- [10단계: \(선택 사항\) Amazon Kinesis Video Streams Edge 에이전트를 기본 프로세스로 실행](#)

1단계: 디바이스에 필요한 종속 항목을 설치합니다.

Note

지원되는 운영 체제 목록은 [이 섹션](#)을 참조하십시오. [the section called “Amazon Kinesis Video Streams Edge 에이전트는 어떤 운영 체제를 지원합니까?”](#)

디바이스에 종속 항목 설치

1. Amazon Kinesis Video Streams Edge 에이전트를 실행하려면 디바이스에 다음과 같은 적절한 라이브러리를 설치하십시오.

Ubuntu

유형:

```
wget -O- https://apt.corretto.aws/corretto.key | sudo apt-key add -
sudo add-apt-repository 'deb https://apt.corretto.aws stable main'
sudo apt-get update

sudo apt-get install -y gcc libssl-dev libcurl4-openssl-dev liblog4cplus-dev \
libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev \
gstreamer1.0-plugins-base-apps gstreamer1.0-plugins-bad \
gstreamer1.0-plugins-good gstreamer1.0-tools \
unzip java-11-amazon-corretto-jdk maven
```

Amazon Linux 2

유형:

```
sudo yum update -y && sudo yum upgrade -y && sudo yum clean all -y
sudo yum install -y gcc-c++ openssl-devel libcurl-devel gstreamer1* wget \
java-11-amazon-corretto tar
```


log4cplus-2.1.0소스에서 설치하십시오.

```
wget https://github.com/log4cplus/log4cplus/releases/download/REL_2_1_0/
log4cplus-2.1.0.tar.gz
tar -xzvf log4cplus-2.1.0.tar.gz
cd log4cplus-2.1.0 && \
mkdir build && \
cd build && \
cmake .. && \
sudo make && \
sudo make install
```

apache-maven-3.9.2소스에서 설치하세요.

```
wget https://dlcdn.apache.org/maven/maven-3/3.9.2/binaries/apache-maven-3.9.2-
bin.tar.gz
RUN tar -xzvf apache-maven-3.9.2-bin.tar.gz -C /opt
```

Important

일부 서비스를 다시 시작해야 한다는 화면이 표시되면 Enter 키를 눌러 확인을 선택합니다.

자세한 내용은 [Amazon Corretto 11](#) 사용 설명서를 참조하십시오.

2. 를 설치하십시오. AWS Command Line InterfaceAWS Command Line Interface 사용 [설명서의 최신 버전 설치 또는 업데이트 AWS CLI](#) 절차를 참조하십시오.

2단계: IP 카메라 RTSP URL을 위한 Amazon Kinesis Video Streams AWS Secrets Manager 및 리소스 생성

다음 절차에 따라 필요한 스트림과 암호를 생성하십시오. AWS Secrets Manager정책에 생성된 리소스의 ARN이 필요하므로 이 단계를 먼저 수행하십시오.

아마존 Kinesis Video Streams 생성

AWS CLI, 또는 API를 사용하여 Amazon Kinesis Video Streams를 AWS Management Console에서 생성하십시오.

에서 [Amazon Kinesis Video Streams 콘솔](#)을 엽니다. AWS Management Console 왼쪽 탐색창에서 비디오 스트림을 선택합니다.

자세한 정보는 [the section called “Kinesis 비디오 스트림 생성”](#)을 참조하세요.

에서 시크릿 생성하기 AWS Secrets Manager

에서 AWS Management Console [AWS Secrets Manager 콘솔](#)을 엽니다. 왼쪽 탐색창에서 시크릿을 선택합니다.

적절한 지역이 선택되었는지 확인하십시오.

1. 새 암호 저장을 선택합니다.

a. 1단계: 암호 유형 선택

- 다른 유형의 보안 암호(Other type of secrets)를 선택합니다.
- 키값 쌍 섹션에서 키-값 쌍을 추가합니다.

키: MediaURI

Note

키는 다음과 같아야 합니다. MediaURI 대소문자를 구분합니다. 잘못 입력하면 응용 프로그램이 작동하지 않습니다.

값: *Your MediaURI*

Example

예: `rtsp://<YourCameraIPAddress>:<YourCameraRTSPPort>/
YourCameraMediaURI.`

- 2단계: 시크릿 구성. 이 비밀번호에 이름을 지정하세요. 원하는 대로 이름을 지어주세요.
- 3단계: 회전 구성 - 선택 사항. 다음을 선택합니다.
- 4단계: 검토. 저장(Store)을 선택합니다.

2. 비밀번호가 즉시 표시되지 않으면 새로 고침 버튼을 선택하세요.

비밀번호의 이름을 선택합니다. 시크릿 ARN을 기록해 둡니다.

3. 스트리밍하려는 각 MediaURI에 대해 이 프로세스를 반복하세요.

Note

AWS 네트워크가 일부 공개 RTSP 소스를 차단합니다. Amazon EC2 인스턴스 내에서 또는 VPN에 연결된 상태에서 관리되지 않는 상태로 실행하는 경우에는 이러한 인스턴스에 액세스할 수 없습니다.

Important

카메라 RTSP URL은 h.264 형식으로 비디오를 스트리밍해야 합니다. 프래그먼트 지속 시간은 에서 언급한 제한을 초과해서는 안 됩니다. [the section called “프로듀서 SDK 한도”](#)

Amazon Kinesis Video Streams Edge 에이전트는 비디오만 지원합니다.

`gst-discoverer-1.0` *Your RtspUrl* 실행하여 디바이스에서 카메라에 접근할 수 있는지 확인하십시오.

생성한 모든 스트림과 시크릿의 ARN을 저장하세요. 다음 단계에 필요합니다.

3단계: IAM 권한 정책 생성

다음 절차에 따라 IAM 정책을 생성하십시오. 이 권한 정책은 리소스에 대한 선택적 액세스 제어 (지원되는 작업의 일부) 를 허용합니다. AWS 이 경우 AWS 리소스는 Amazon Kinesis Video Streams Edge 에이전트가 스트리밍하려는 비디오 스트림입니다. 리소스에는 Amazon Kinesis Video Streams Edge 에이전트가 검색할 수 있는 AWS Secrets Manager 비밀도 포함되어 있습니다. 자세한 내용을 알아보려면 [IAM 정책](#)을 참조하세요.

JSON 정책 편집기를 사용하여 정책을 생성합니다.

1. <https://console.aws.amazon.com/iam/> 에서 AWS Management Console 로그인하고 IAM 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 정책을 선택합니다.

정책을 처음으로 선택하는 경우 관리형 정책 소개 페이지가 나타납니다. 시작하기를 선택합니다.

3. 페이지 상단에서 정책 생성을 선택합니다.
4. 정책 편집기 섹션에서 JSON 옵션을 선택합니다.
5. 다음 JSON 정책 문서를 입력합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:PutMetricData",
        "kinesisvideo:ListStreams",
        "iot:Connect",
        "iot:Publish",
        "iot:Subscribe",
        "iot:Receive"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "kinesisvideo:DescribeStream",
        "kinesisvideo:PutMedia",
        "kinesisvideo:TagStream",
        "kinesisvideo:GetDataEndpoint"
      ],
      "Resource": [
        "arn:aws:kinesisvideo:*:*:stream/streamName1/*",
        "arn:aws:kinesisvideo:*:*:stream/streamName2/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "secretsmanager:GetSecretValue",
      "Resource": [
        "arn:aws:secretsmanager:*:*:secret:*",
        "arn:aws:secretsmanager:*:*:secret:*"
      ]
    }
  ]
}
```

```

    ]
  }
]
}

```

Note

를 비디오 스트림용 `arn:aws:kinesisvideo:*:*:stream/streamName2/*` ARN으로 `arn:aws:kinesisvideo:*:*:stream/streamName1/*` 바꾸고, 에서 생성한 MediaURI 암호가 포함된 `arn:aws:secretsmanager:*:*:secret:*` ARN으로 바꾸십시오. [the section called “2. IP 카메라 RTSP URL을 위한 리소스 생성”](#) Amazon Kinesis Video Streams Edge 에이전트가 액세스하도록 하려는 보안 정보에는 ARN을 사용하십시오.

6. 다음을 선택합니다.

Note

언제든지 시각적 편집기 옵션과 JSON 편집기 옵션 간에 전환할 수 있습니다. 그러나 변경을 적용하거나 시각적 편집기에서 다음을 선택한 경우 IAM은 시각적 편집기에 최적화되도록 정책을 재구성할 수 있습니다. 자세한 내용은 IAM 사용 [설명서의 정책 재구성을](#) 참조하십시오.

7. 검토 및 생성 페이지에서 생성 중인 정책에 대한 정책 이름과 설명 (선택 사항) 을 입력합니다. 이 정책에 정의된 권한을 검토하여 정책이 부여한 권한을 확인합니다.
8. 정책 생성을 선택하고 새로운 정책을 저장합니다.

4단계: IAM 역할 생성

() 에서 임시 자격 증명을 얻기 위해 이 단계에서 생성한 역할을 수입할 수 있습니다. AWS IoT AWS Security Token Service AWS STS이 Amazon Kinesis Video Streams Edge 에이전트에서 자격 증명 권한 요청을 수행할 때 수행됩니다.

아마존 키네시스 비디오 스트림 (IAM 콘솔) 에 대한 서비스 역할 생성

1. AWS Management Console [로그인하고 https://console.aws.amazon.com/iam/](https://console.aws.amazon.com/iam/) 에서 IAM 콘솔을 엽니다.
2. IAM 콘솔의 탐색 창에서 역할을 선택하고 역할 생성을 선택합니다.

3. 사용자 지정 신뢰 정책 역할 유형을 선택하고 다음 정책을 붙여넣습니다.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Principal": {
      "Service": "credentials.iot.amazonaws.com"
    },
    "Action": "sts:AssumeRole"
  }
}
```

4. 에서 [the section called “3. IAM 권한 정책 생성”](#) 생성한 IAM 정책 옆의 상자를 선택합니다.
5. 다음을 선택합니다.
6. 이 역할의 용도를 식별하는 데 도움이 되는 역할 이름 또는 역할 이름 접미사를 입력합니다.

Example

예: KvsEdgeAgentRole

7. (선택 사항) 설명에 새 역할에 대한 설명을 입력합니다.
8. (선택 사항) 태그를 키/값 쌍으로 첨부하여 역할에 메타데이터를 추가합니다.

IAM에서 태그를 사용하는 방법에 대한 자세한 내용은 IAM 사용 설명서의 [IAM 리소스 태깅](#)을 참조하십시오.

9. 역할을 검토한 다음 역할 생성을 선택합니다.

5단계: 역할 별칭 생성 AWS IoT

에서 생성한 IAM AWS IoT 역할의 역할 별칭을 생성하려면 다음 절차를 따르십시오. [the section called “4. IAM 역할 생성”](#) 역할 별칭은 IAM 역할을 가리키는 대체 데이터 모델입니다. AWS IoT 자격 증명 공급자 요청에는 () 에서 임시 자격 증명을 얻기 위해 어떤 IAM 역할을 말아야 하는지를 나타내는 역할 별칭이 포함되어야 합니다. AWS Security Token Service AWS STS 자세한 내용은 [인증서를 사용하여 보안 토큰을 얻는 방법](#)을 참조하십시오.

AWS IoT 역할 별칭 생성

1. 에 AWS Management Console 로그인하고 <https://console.aws.amazon.com/iot/> 에서 AWS IoT Core 콘솔을 엽니다.

2. 적절한 지역이 선택되었는지 확인하십시오.
3. 왼쪽 탐색에서 보안을 선택한 다음 역할 별칭을 선택합니다.
4. 역할 별칭 생성을 선택합니다.
5. 역할 별칭의 이름을 입력합니다.

Example

예: KvsEdgeAgentRoleAlias

6. 역할 드롭다운에서 생성한 IAM 역할을 선택합니다. [the section called “4. IAM 역할 생성”](#)
7. 생성을 선택합니다. 다음 페이지에서 역할 별칭이 성공적으로 생성되었다는 메모를 볼 수 있습니다.
8. 새로 만든 역할 별칭을 검색하여 선택합니다. 역할 별칭 ARN을 기록해 둡니다. 다음 단계의 AWS IoT 정책에 이 정보가 필요합니다.

6단계: AWS IoT 정책 생성

다음 절차에 따라 장치 인증서에 첨부할 AWS IoT 정책을 만드십시오. 이렇게 하면 AWS IoT 기능에 대한 권한이 부여되고 인증서를 사용하여 역할 별칭을 수임할 수 있습니다.

AWS IoT Core 정책을 사용하여 AWS IoT Core 데이터 플레인에 대한 액세스를 제어할 수 있습니다. AWS IoT Core 데이터 플레인은 다음 작업을 수행하는 데 사용할 수 있는 작업으로 구성됩니다.

- AWS IoT Core 메시지 브로커에 연결
- MQTT 메시지 전송 및 수신
- 사물의 디바이스 새도 가져오기 또는 업데이트

자세한 내용은 [AWS IoT Core 정책](#)을 참조하세요.

AWS IoT 정책 편집기를 사용하여 AWS IoT 정책 생성

1. <https://console.aws.amazon.com/iot/>에서 AWS Management Console 로그인하고 AWS IoT Core 콘솔을 엽니다.
2. 왼쪽 탐색 메뉴에서 보안을 선택한 다음 정책을 선택합니다.
3. 정책 생성을 선택합니다.
4. 정책의 이름을 입력합니다.

Example

정책 이름의 예로는 KvsEdgeAccessIotPolicy가 있습니다.

5. (선택 사항) 태그를 키 값 페어로 연결하여 메타데이터를 정책에 추가합니다.

IAM에서 태그를 사용하는 방법에 대한 자세한 내용은 개발자 안내서의 [AWS IoT 리소스 태그 AWS IoT Core 지정](#)을 참조하십시오.

6. JSON 탭을 선택합니다.
7. 다음 JSON 정책 문서를 붙여 넣습니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:Connect",
        "iot:Publish",
        "iot:Subscribe",
        "iot:Receive"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:AssumeRoleWithCertificate"
      ],
      "Resource": "your-role-alias-arn"
    }
  ]
}
```

Note

에서 생성한 역할 별칭의 *your-role-alias-arn* ARN으로 대체합니다. [the section called “5. AWS IoT 역할 별칭 생성”](#)

8. [Create] 를 선택하여 작업 내용을 저장합니다.

7단계: AWS IoT 사물 생성 및 자격 증명 획득 AWS IoT Core

이제 다음을 생성했습니다.

- IAM 권한 정책. [the section called “3. IAM 권한 정책 생성”](#) 섹션을 참조하십시오.
- 권한 정책이 첨부된 IAM 역할 [the section called “4. IAM 역할 생성”](#) 섹션을 참조하십시오.
- IAM AWS IoT 역할의 역할 별칭. [the section called “5. AWS IoT 역할 별칭 생성”](#) 섹션을 참조하십시오.
- 현재 어떤 리소스에도 연결되지 않은 AWS IoT 정책입니다. AWS [the section called “6. AWS IoT 정책 생성”](#) 섹션을 참조하십시오.

사물을 생성 및 등록하고 AWS IoT Core 액세스 자격 증명을 얻으려면

1. 디바이스를 사물로 AWS IoT 등록하고 디바이스의 X.509 인증서를 생성합니다.
 - a. <https://console.aws.amazon.com/iot/> 에서 AWS Management Console 로그인하고 AWS IoT Core 콘솔을 엽니다.
 - b. 적절한 지역을 선택합니다.
 - c. 왼쪽 내비게이션에서 모든 기기를 선택한 다음 사물을 선택합니다.
 - d. 사물 만들기를 선택합니다.
 - e. 단일 사물 생성을 선택한 후 다음을 선택합니다.
 1. 단계 1. 사물 속성을 지정합니다.

사물의 이름을 입력하고 다음을 선택합니다.
 2. 단계 2. 장치 인증서 구성

새 인증서 자동 생성 (권장) 을 선택하고 다음을 선택합니다.
 3. 단계 3. 인증서에 정책 연결

에서 생성한 권한 정책을 [the section called “6. AWS IoT 정책 생성”](#) 검색하십시오.

정책 옆의 확인란을 선택하고 Create thing (항목 만들기) 을 선택합니다.
 - f. 표시되는 창에서 다음 파일을 다운로드합니다.

- 디바이스 인증서. X.509 인증서입니다.
- 퍼블릭 키 파일
- 프라이빗 키 파일
- 아마존 트러스트 서비스 엔드포인트 (RSA 2048비트 키: 아마존 루트 CA 1)

이후 단계를 위해 각 파일의 위치를 기록해 두십시오.

- 완료를 선택합니다. 다음 페이지에서 사물이 성공적으로 생성되었다는 메모를 볼 수 있습니다.
 - 위에서 다운로드한 파일을 사물에 옮기십시오 AWS IoT (아직 없는 경우).
- AWS 계정의 자격 증명 제공업체 엔드포인트를 확보하세요.

AWS CLI

다음 명령을 실행합니다:

```
aws iot describe-endpoint --endpoint-type iot:CredentialProvider
```

AWS Management Console

에서 [AWS CloudShell](#) 다음 명령을 실행합니다.

```
aws iot describe-endpoint --endpoint-type iot:CredentialProvider
```

이후 단계를 위해 이 정보를 기록해 두십시오.

- AWS 계정의 디바이스 데이터 엔드포인트를 확보하세요.

AWS CLI

다음 명령을 실행합니다:

```
aws iot describe-endpoint --endpoint-type iot:Data-ATS
```

AWS Management Console

다음에 따릅니다.

1. <https://console.aws.amazon.com/iot/> 에서 AWS Management Console 로그인하고 AWS IoT Core 콘솔을 엽니다.
2. 왼쪽 탐색창에서 설정을 선택합니다.
3. 장치 데이터 엔드포인트를 찾습니다.

이후 단계를 위해 이 정보를 기록해 두십시오.

4. (선택 사항) 인증서가 올바르게 생성되었는지 확인하십시오.

다음 명령을 실행하여 항목이 올바르게 생성되었는지 확인합니다.

```
curl --header "x-amzn-iot-thingname:your-thing-name" \
  --cert /path/to/certificateID-certificate.pem.crt \
  --key /path/to/certificateID-private.pem.key \
  --cacert /path/to/AmazonRootCA1.pem \
  https://your-credential-provider-endpoint/role-aliases/your-role-alias-name/
credentials
```

자세한 내용은 [인증서를 사용하여 보안 토큰을 얻는 방법을 참조하십시오](#).

8단계: Amazon Kinesis Video Streams Edge 에이전트의 구축 및 실행

Amazon Kinesis Video Streams Edge 에이전트의 구축 및 실행

1. 제공된 링크를 사용하여 tar 파일을 다운로드하십시오.

Amazon Kinesis Video Streams Edge 에이전트 관심 양식을 작성한 경우 이메일에서 다운로드 링크를 확인하십시오. 양식을 아직 작성하지 않은 경우 [여기에서](#) 작성하십시오.

2. 체크섬을 확인하세요.
3. 디바이스에서 바이너리와 jar를 추출합니다.

유형: `tar -xvf kvs-edge-agent.tar.gz`.

압축을 풀고 나면 폴더 구조가 다음과 같이 표시됩니다.

```
kvs-edge-agent/LICENSE
kvs-edge-agent/THIRD-PARTY-LICENSES
kvs-edge-agent/pom.xml
```

```

kvs-edge-agent/KvsEdgeComponent
kvs-edge-agent/KvsEdgeComponent/recipes
kvs-edge-agent/KvsEdgeComponent/recipes/recipe.yaml
kvs-edge-agent/KvsEdgeComponent/artifacts
kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/edge_log_config
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/kvs-edge-agent.jar
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/libgstkvssink.so
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/libIngestorPipelineJNI.so
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/lib
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/lib/libcproducer.so
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/lib/libKinesisVideoProducer.so

```

Note

릴리스 폴더 이름은 최신 바이너리 릴리스 번호를 반영하는 방식으로 설정해야 합니다. 예를 들어 1.0.0 릴리스에서는 폴더 이름이 1.0.0으로 설정됩니다.

4. 종속성 항아리를 빌드하세요.

Note

에 포함된 jar에는 종속성이 kvs-edge-agent.tar.gz 없습니다. 다음 단계를 사용하여 해당 라이브러리를 빌드하십시오.

포함된 kvs-edge-agent 폴더로 이동합니다pom.xml.

mvn clean package를 입력합니다.

그러면 Amazon Kinesis Video Streams Edge 에이전트에 필요한 종속성이 포함된 jar 파일이 생성됩니다. kvs-edge-agent/target/libs.jar

5. 구성 요소의 `libs.jar` 아티팩트가 들어 있는 폴더에 파일을 배치합니다.

```
mv ./target/libs.jar ./KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/를 입력합니다.
```

6. 이전 단계의 값을 사용하여 환경 변수를 설정합니다. 다음 표에는 변수에 대한 설명이 나와 있습니다.

환경 변수 이름	필수	설명
AWS_REGION	예	사용되는 지역. 예: us-west-2
AWS_IOT_CA_CERT	예	TLS를 통해 백엔드 서비스와의 신뢰를 구축하는 데 사용되는 CA 인증서의 파일 경로. 예: <i>/file/path/to/AmazonRootCA1.pem</i>
AWS_IOT_CORE_CERT	예	X.509 인증서의 파일 경로. 예: <i>/file/path/to/certificateID-certificate.pem.crt</i>
AWS_IOT_CORE_CREDENTIAL_ENDPOINT	예	계정의 AWS IoT Core 자격 증명 엔드포인트 공급자 엔드포인트 . AWS 예: <i>credential-account-specific-prefix.credentials.iot.region.amazonaws.com</i>
AWS_IOT_CORE_DATA_ATS_ENDPOINT	예	AWS 계정의 AWS IoT Core 데이터 플레인 엔드포인트 .

환경 변수 이름	필수	설명
		<p>예: <code>data-account-specific-prefix .iot.aws-region .amazonaws.com</code></p>
AWS_IOT_CORE_PRIVATE_KEY	예	<p>공개/개인 키 쌍에서 사용되는 개인 키의 파일 경로입니다. 자세한 내용은 의 키 관리를 참조하십시오. AWS IoT</p> <p>예: <code>/file/path/to/certificateID-private.pem.key</code></p>
AWS_IOT_CORE_ROLE_ALIAS	예	<p>연결할 때 사용할 AWS IAM 역할을 가리키는 역할 별칭의 이름. AWS IoT Core</p> <p>예: <code>kvs-edge-role-alias</code></p>
AWS_IOT_CORE_THING_NAME	예	<p>애플리케이션이 실행되는 AWS IoT 대상 객체의 이름.</p> <p>예: <code>my-edge-device-thing</code></p>

환경 변수 이름	필수	설명
GST_PLUGIN_PATH	예	<p>gstkvssink 및 IngestorPipelineJNI 플랫폼 종속 라이브러리가 들어 있는 폴더를 가리키는 파일 경로입니다. GStreamer가 이러한 플러그인을 로드할 수 있도록 합니다. 자세한 내용은 GStreamer 요소 다운로드, 빌드 및 구성을 참조하십시오.</p> <p>예: <i>/download-location /kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/ EdgeAgent Version /</i></p>
LD_LIBRARY_PATH	예	<p>cproducer 및 KinesisVideoProducer 플랫폼 종속 라이브러리가 포함된 디렉터리를 가리키는 파일 경로입니다.</p> <p>예: <i>/download-location /kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/ EdgeAgent Version /lib/</i></p>

환경 변수 이름	필수	설명
AWS_KVS_EDGE_CLOUD_WATCH_ENABLED	아니요	<p>Amazon Kinesis Video Streams Edge 에이전트가 작업 상태 지표를 Amazon CloudWatch에 게시할지 여부를 결정합니다.</p> <p>허용되는 값: TRUE/FALSE(대소문자 구분 안 함). 제공되지 않은 FALSE 경우 기본값은 입니다.</p> <p>예: 거짓</p>
AWS_KVS_EDGE_LOG_LEVEL	아니요	<p>Amazon Kinesis Video Streams Edge Agent 출력의 로깅 수준입니다.</p> <p>허용되는 값:</p> <ul style="list-style-type: none"> • OFF • ALL • 치명적입니다 • ERROR • WARN • 정보, 기본값 (제공되지 않은 경우) • DEBUG • TRACE <p>예: INFO</p>

환경 변수 이름	필수	설명
AWS_KVS_EDGE_LOG_M AX_FILE_SIZE	아니요	<p>로그 파일이 이 크기에 도달하면 롤오버가 발생합니다.</p> <ul style="list-style-type: none"> • 최소: 0 • 최대: 1만 • 기본값: 20 (제공되지 않은 경우) • 단위: 메가바이트 (MB) <p>예: 5</p>
AWS_KVS_EDGE_LOG_O UTPUT_DIRECTORY	아니요	<p>Amazon Kinesis Video Streams Edge 에이전트 로그가 출력되는 디렉토리를 가리키는 파일 경로입니다. 지정하지 않은 <code>./log</code> 경우 기본값은 <code>./log</code>입니다.</p> <p>예: <code>/file/path/</code></p>
AWS_KVS_EDGE_LOG_R OLLOVER_COUNT	아니요	<p>삭제하기 전에 보관할 롤오버된 로그의 수입니다.</p> <ul style="list-style-type: none"> • 최소: 1 • 최대: 100개 • 기본값: 10 (제공되지 않은 경우) <p>예: 20</p>

환경 변수 이름	필수	설명
AWS_KVS_EDGE_RECORDING_DIRECTORY	아니요	기록된 미디어가 기록될 디렉토리를 가리키는 파일 경로. 지정하지 않은 경우 현재 디렉토리가 기본값입니다. 예: <i>/file/path/</i>
GST_DEBUG	아니요	출력할 GStreamer 로그의 수준을 지정합니다. 자세한 내용은 GStreamer 설명서를 참조하십시오. 예: 0
GST_DEBUG_FILE	아니요	GStreamer 디버그 로그의 출력 파일을 지정합니다. 설정하지 않으면 디버그 로그가 표준 오류로 출력됩니다. 자세한 내용은 GStreamer 설명서를 참조하십시오. 예: <i>/tmp/gstreamer-logging.log</i>

7. GStreamer 캐시를 지우세요. 유형:

```
rm ~/.cache/gstreamer-1.0/registry.your-os-architecture.bin
```

자세한 내용은 [GStreamer](#) 레지스트리 설명서를 참조하십시오.

8. java 명령을 준비하고 실행합니다. Amazon Kinesis Video Streams 엣지 에이전트는 다음과 같은 인수를 받아들입니다.


Java 속성 이름	필수	설명
java.library.path	아니요	gstkvssink 및 IngestorPipelineJNI

Java 속성 이름

필수

설명

종속 라이브러리가 들어 있는 폴더를 가리키는 파일 경로. 제공하지 않으면 Amazon Kinesis Video Streams Edge 에이전트는 현재 디렉터리에서 해당 항목을 검색합니다.

 Important

Amazon Kinesis Video Streams Edge 에이전트는 이러한 파일을 찾을 수 없는 경우 제대로 작동하지 않습니다.

예: */file/path/*

이를 `-Djava-property-name=value` 설정하려면 jar 실행에 사용되는 Java 명령을 추가하십시오.

예:

```
java -Djava.library.path=download-location/kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion \
  --add-opens java.base/jdk.internal.misc=ALL-UNNAMED \
  -Dio.netty.tryReflectionSetAccessible=true \
  -cp kvs-edge-agent.jar:libs.jar \
  com.amazonaws.kinesisvideo.edge.controller.ControllerApp
```

⚠ Important

와 같은 디렉토리에서 위의 java 명령을 실행합니다/*download-location*/kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/*EdgeAgentVersion*.

9. 를 사용하여 응용 프로그램에 구성을 전송합니다 AWS CLI.

a. 새 파일 생성,*example-edge-configuration.json*.

다음 코드를 파일에 붙여 넣습니다. 디바이스의 시스템 시간에 따라 매일 오전 9:00:00 부터 오후 4:59:59 까지 기록하는 샘플 구성입니다. AWS IoT 또한 매일 오후 7:00:00 부터 9:59:59 까지 녹화된 미디어를 업로드합니다.

자세한 정보는 [the section called "StartEdgeConfigurationUpdate"](#)을 참조하세요.

```
{
  "StreamARN": "arn:aws:kinesisvideo:your-region:your-account-id:stream/your-stream/0123456789012",
  "EdgeConfig": {
    "HubDeviceArn": "arn:aws:iot:your-region:your-account-id:thing/kvs-edge-agent-demo",
    "RecorderConfig": {
      "MediaSourceConfig": {
        "MediaUriSecretArn": "arn:aws:secretsmanager:your-region:your-account-id:secret:your-secret-dRbHJQ",
        "MediaUriType": "RTSP_URI"
      },
      "ScheduleConfig": {
        "ScheduleExpression": "0 0 9,10,11,12,13,14,15,16 ? * * **",
        "DurationInSeconds": 3599
      }
    },
    "UploaderConfig": {
      "ScheduleConfig": {
        "ScheduleExpression": "0 0 19,20,21 ? * * **",
        "DurationInSeconds": 3599
      }
    },
    "DeletionConfig": {
      "EdgeRetentionInHours": 15,

```

```

        "LocalSizeConfig": {
            "MaxLocalMediaSizeInMB": 2800,
            "StrategyOnFullSize": "DELETE_OLDEST_MEDIA"
        },
        "DeleteAfterUpload": true
    }
}

```

- b. 파일을 Amazon Kinesis Video Streams Edge 에이전트로 보내려면 필드에 AWS CLI 다음을 입력합니다.

```

aws kinesishvideo start-edge-configuration-update --cli-input-json
"file://example-edge-configuration.json"

```

10. Amazon Kinesis Video Streams Edge 에이전트의 각 스트림에 대해 이전 단계를 반복합니다.

9단계: (선택 사항) 디바이스에 CloudWatch 에이전트 설치

Note

[CloudWatch 할당량](#)을 숙지하세요.

다음 절차에 따라 Amazon Kinesis Video Streams Edge CloudWatch Agent에서 생성된 로그를 자동으로 업로드하도록 에이전트를 CloudWatch 설치하고 구성합니다.

디바이스에 CloudWatch 에이전트를 설치하는 [절차](#)는 Amazon CloudWatch 사용 설명서를 참조하십시오.

구성을 입력하라는 메시지가 표시되면 다음 구성 중 하나를 선택합니다.

Important

다음 `file_path` 구성에서는 기본 로깅 출력 위치가 사용된다고 가정합니다. 사용된 파일 경로는 다음 위치에서 Amazon Kinesis Video Streams Edge 에이전트를 실행하고 있다고 가정합니다. `download-location/kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/version`

- 로그를 업로드하고 디바이스 RAM 및 CPU 지표를 게시하도록 CloudWatch 에이전트를 구성하려면 구성 파일에 다음을 붙여넣습니다.

```
{
  "agent": {
    "run_as_user": "ubuntu",
    "metrics_collection_interval": 60
  },
  "metrics": {
    "metrics_collected": {
      "mem": {
        "measurement": [
          "mem_used_percent"
        ],
        "append_dimensions": {
          "IotThing": "YourIotThingName"
        }
      },
      "cpu": {
        "resources": [
          "*"
        ],
        "measurement": [
          "usage_active"
        ],
        "totalcpu": true,
        "append_dimensions": {
          "IotThing": "YourIotThingName"
        }
      }
    }
  },
  "logs": {
    "logs_collected": {
      "files": {
        "collect_list": [
          {
            "file_path": "download-location/kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/version/log/java_kvs.log",
            "log_group_name": "/aws/kinesisvideo/EdgeRuntimeAgent",
            "log_stream_name": "YourIotThingName-java_kvs.log"
          },
          {

```



```
[Service]
Type=simple
Restart=on-failure
RestartSec=10
WorkingDirectory=/download-location/kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion
Environment="GST_PLUGIN_PATH=/download-location/kvs-edge-agent/KvsEdgeComponent/
artifacts/aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion"
Environment="LD_LIBRARY_PATH=/download-location/kvs-edge-agent/KvsEdgeComponent/
artifacts/aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/lib"
...
Environment="AWS_IOT_CORE_DATA_ATS_ENDPOINT=data-account-specific-prefix.iot.aws-
region.amazonaws.com"
ExecStart=/usr/lib/jvm/java-11-amazon-corretto/bin/java --add-opens java.base/
jdk.internal.misc=ALL-UNNAMED -Dio.netty.tryReflectionSetAccessible=true -cp kvs-
edge-agent.jar:libs.jar com.amazonaws.kinesisvideo.edge.controller.ControllerApp

[Install]
WantedBy=multi-user.target
```

systemd서비스 구성 파일에서 허용하는 매개변수에 대한 자세한 내용은 [설명서를](#) 참조하십시오.

Note

에 지정된 ... 위치에 필수 환경 변수를 추가합니다 [the section called “8. Amazon Kinesis Video Streams Edge 에이전트의 구축 및 실행”](#).

2. 새 서비스를 포함하도록 서비스 파일을 다시 로드합니다.

`sudo systemctl daemon-reload`를 입력합니다.

3. 서비스를 시작합니다.

`sudo systemctl start aws.kinesisvideo.edge-runtime-agent.service`를 입력합니다.

4. Amazon Kinesis Video Streams Edge 에이전트 서비스의 상태를 확인하여 서비스가 실행 중인지 확인하십시오.

`sudo systemctl status aws.kinesisvideo.edge-runtime-agent.service`를 입력합니다.

다음은 표시되는 출력의 예입니다.

```
aws.kinesisvideo.edge-runtime-agent.service - AWS Kinesis Video Streams edge agent
  Loaded: loaded (/etc/systemd/system/aws.kinesisvideo.edge-runtime-agent.service; disabled; vendor preset: enabled)
  Active: active (running) since Thu 2023-06-08 19:15:02 UTC; 6s ago
  Main PID: 506483 (java)
  Tasks: 23 (limit: 9518)
  Memory: 77.5M
  CPU: 4.214s
  CGroup: /system.slice/aws.kinesisvideo.edge-runtime-agent.service
          ##506483 /usr/lib/jvm/java-11-amazon-corretto/bin/java -cp kvs-edge-agent.jar:libs.jar com.amazonaws.kinesisvideo.edge.controller.ControllerApp
```

5. 로그에 오류가 있는지 검사하십시오.

`journalctl -e -u aws.kinesisvideo.edge-runtime-agent.service`를 입력합니다.

6. `systemctl --help`를 사용하여 프로세스를 관리하는 전체 옵션 목록을 입력하십시오 `systemctl`.

Amazon Kinesis Video Streams Edge 에이전트를 관리하기 위한 몇 가지 일반적인 명령은 다음과 같습니다.

- 다시 시작하려면 `sudo systemctl restart aws.kinesisvideo.edge-runtime-agent.service`.
- 중지하려면 `sudo systemctl stop aws.kinesisvideo.edge-runtime-agent.service`.
- 장치를 재부팅할 때마다 자동으로 시작하려면 `sudo systemctl enable aws.kinesisvideo.edge-runtime-agent.service`.

Amazon Kinesis Video Streams 엣지 에이전트를 다음 위치에 배포하십시오. AWS IoT Greengrass

다음 단계에 따라 Amazon Kinesis Video Streams Edge 에이전트를 AWS IoT Greengrass 배포하여 IP 카메라에서 미디어를 녹화하고 업로드할 수 있습니다.

주제

- [1단계: 우분투 아마존 EC2 인스턴스 생성](#)
- [2단계: 디바이스에 AWS IoT Greengrass V2 코어 디바이스 설정](#)

- [3단계: IP 카메라 RTSP URL을 위한 Amazon Kinesis Video Streams AWS Secrets Manager 및 리소스 생성](#)
- [4단계: 토큰 교환 서비스 \(TES\) 역할에 권한 추가](#)
- [5단계: 디바이스에 AWS IoT Greengrass 시크릿 매니저 컴포넌트 설치](#)
- [6단계: Amazon Kinesis Video Streams Edge AWS IoT Greengrass Agent 구성 요소를 디바이스에 배포합니다.](#)
- [7단계: \(선택 사항\) 디바이스에 AWS IoT Greengrass 로그 관리자 구성 요소 설치](#)

1단계: 우분투 아마존 EC2 인스턴스 생성

우분투 Amazon EC2 인스턴스를 생성하려면 다음과 같이 하십시오.

우분투 아마존 EC2 인스턴스 생성

1. AWS Management Console [로그인하고 https://console.aws.amazon.com/ec2/](https://console.aws.amazon.com/ec2/) 에서 Amazon EC2 콘솔을 엽니다.

적절한 지역이 선택되었는지 확인하십시오.

2. 인스턴스 시작을 선택합니다.

다음 작업을 완료합니다.

- 이름 - 인스턴스의 이름을 입력합니다.
- 애플리케이션 및 OS 이미지 (Amazon 머신 이미지) — Ubuntu를 선택합니다.
- 인스턴스 유형 — t2.large를 선택합니다.
- 키 페어 로그인 — 고유한 키 페어를 생성합니다.
- 네트워크 설정 — 기본값을 유지합니다.
- 스토리지 구성 - 볼륨을 256GiB로 늘립니다.
- 고급 설정 — 기본값을 유지합니다.

3. 인스턴스를 시작하고 해당 인스턴스에서 SSH를 실행합니다.

다음은 따릅니다.

1. 왼쪽 탐색창에서 인스턴스를 선택한 다음 인스턴스 ID를 선택합니다.
2. 오른쪽 상단에서 Connect를 선택합니다.
3. SSH 클라이언트를 선택하고 화면의 지침을 따릅니다.

4. 터미널을 열고 다운로드한 .pem 파일 (에 있을 수 ~/Downloads 있음) 을 탐색합니다.
 5. 이 절차를 처음 수행하면 “호스트 (...) 의 인증을 확인할 수 없습니다.” 라는 메시지가 표시됩니다. yes를 입력합니다.
4. 시스템 라이브러리를 설치하여 Amazon Kinesis Video Streams Edge 에이전트를 인스턴스에 빌드합니다.

```
wget -O- https://apt.corretto.aws/corretto.key | sudo apt-key add -
sudo add-apt-repository 'deb https://apt.corretto.aws stable main'

sudo apt-get update

sudo apt-get install -y gcc libssl-dev libcurl4-openssl-dev liblog4cplus-dev \
libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev \
gstreamer1.0-plugins-base-apps gstreamer1.0-plugins-bad \
gstreamer1.0-plugins-good gstreamer1.0-tools \
unzip java-11-amazon-corretto-jdk maven
```

Important

일부 서비스를 다시 시작해야 한다는 화면이 표시되면 Enter 키를 눌러 확인을 선택합니다.

자세한 내용은 [Amazon Corretto 11](#) 사용 설명서를 참조하십시오.

2단계: 디바이스에 AWS IoT Greengrass V2 코어 디바이스 설정

다음 절차에 따라 Amazon EC2 인스턴스에 AWS IoT Greengrass 코어 핵 소프트웨어를 설치합니다.

코어 디바이스를 AWS IoT Greengrass 설정합니다.

1. AWS Management Console, <https://console.aws.amazon.com/iot/> 에 로그인합니다.

적절한 지역이 선택되었는지 확인하십시오.

2. 왼쪽 탐색창에서 Greengrass 장치, Core 장치를 선택합니다.
3. 코어 디바이스 1개 설정을 선택합니다.
4. 화면의 단계를 완료하세요.

- 1단계: 그린그래스 코어 디바이스를 등록합니다. 장치 이름을 입력합니다.
- 2단계: 사물 그룹에 추가하여 지속적 배포를 적용합니다. 그룹 없음을 선택합니다.
- 3단계: 그린그래스 코어 소프트웨어를 설치합니다. 리눅스를 선택합니다.
 - 3.1단계: 디바이스에 Java 설치

Java는 일부로 설치됩니다 [the section called “1. 우분투 인스턴스 생성”](#). Java가 아직 설치되지 않은 경우 해당 단계로 돌아가십시오.

- 3.2단계: AWS 자격 증명을 디바이스에 복사

bash/zsh 옵션을 열고 Amazon EC2 인스턴스에 내보내기 명령을 붙여넣습니다.

- 3.3단계: 설치 프로그램 실행
 1. Ubuntu Amazon EC2 인스턴스에서 설치 프로그램 다운로드 및 설치 프로그램 실행 명령을 복사하여 실행합니다.

Note

설치 프로그램 실행 명령은 이전 단계에서 선택한 이름을 기반으로 자동으로 업데이트됩니다.

2. 생성된 토큰 교환 서비스 (TES) 역할을 기록해 둡니다. 나중에 필요합니다.

Note

기본적으로 생성되는 역할은 TokenExchangeRoleGreenGrassV2라고 합니다.

3단계: IP 카메라 RTSP URL을 위한 Amazon Kinesis Video Streams AWS Secrets Manager 및 리소스 생성

다음 절차에 따라 에서 필요한 스트림과 시크릿을 AWS Secrets Manager 생성하십시오. 정책에 생성된 리소스의 ARN이 필요하므로 이 단계를 먼저 수행하십시오.

아마존 Kinesis Video Streams 생성

AWS CLI, 또는 API를 사용하여 Amazon Kinesis Video Streams를 AWS Management Console 생성하십시오.

에서 [Amazon Kinesis Video Streams](#) 콘솔을 엽니다. AWS Management Console 왼쪽 탐색창에서 비디오 스트림을 선택합니다.

자세한 정보는 [the section called “Kinesis 비디오 스트림 생성”](#)을 참조하세요.

에서 시크릿 생성하기 AWS Secrets Manager

에서 AWS Management Console [AWS Secrets Manager 콘솔](#)을 엽니다. 왼쪽 탐색창에서 시크릿을 선택합니다.

적절한 지역이 선택되었는지 확인하십시오.

1. 새 암호 저장을 선택합니다.

a. 1단계: 암호 유형 선택

- 다른 유형의 보안 암호(Other type of secrets)를 선택합니다.
- 키값 쌍 섹션에서 키값 쌍을 추가합니다.

키: MediaURI

Note

키는 다음과 같아야 합니다. MediaURI 대소문자를 구분합니다. 잘못 입력하면 응용 프로그램이 작동하지 않습니다.

값: *Your MediaURI*

Example

예:rtsp://<YourCameraIPAddress>:<YourCameraRTSPPort>/
YourCameraMediaURI.

- 2단계: 시크릿 구성. 이 비밀번호에 이름을 지정하세요. 원하는 대로 이름을 지어주세요.
 - 3단계: 회전 구성 - 선택 사항. 다음을 선택합니다.
 - 4단계: 검토. 저장(Store)을 선택합니다.
2. 비밀번호가 즉시 표시되지 않으면 새로 고침 버튼을 선택하세요.

비밀번호의 이름을 선택합니다. 시크릿 ARN을 기록해 둡니다.

3. 스트리밍하려는 각 MediaURI에 대해 이 프로세스를 반복하세요.

Note

AWS 네트워크가 일부 공개 RTSP 소스를 차단합니다. Amazon EC2 인스턴스 내에서 또는 VPN에 연결된 상태에서 관리되지 않는 상태로 실행하는 경우에는 이러한 인스턴스에 액세스할 수 없습니다.

Important

카메라 RTSP URL은 h.264 형식으로 비디오를 스트리밍해야 합니다. 프래그먼트 지속 시간은 에서 언급한 제한을 초과해서는 안 됩니다. [the section called “프로듀서 SDK 한도”](#)

Amazon Kinesis Video Streams Edge 에이전트는 비디오만 지원합니다.

`gst-discoverer-1.0` *Your RtspUrl* 실행하여 디바이스에서 카메라에 접근할 수 있는지 확인하십시오.

생성한 모든 스트림과 시크릿의 ARN을 저장하세요. 다음 단계에 필요합니다.

4단계: 토큰 교환 서비스 (TES) 역할에 권한 추가

암호를 볼 권한이 있는 디바이스에 토큰 교환 서비스 (TES) 역할을 부여하십시오. 이는 AWS Secrets Manager AWS IoT Greengrass 구성 요소가 제대로 작동하는 데 필요합니다.

TES 역할에 권한 추가

1. 에 AWS Management Console 로그인하고 <https://console.aws.amazon.com/iam/> 에서 IAM 콘솔을 엽니다.
2. 왼쪽 탐색창에서 역할을 선택하고 프로세스 초기에 생성한 TES 역할을 검색합니다.
3. 권한 추가 드롭다운에서 정책 연결을 선택합니다.
4. 정책 생성을 선택합니다.
5. 아래로 스크롤하여 편집을 선택합니다.
6. 정책 편집기에서 JSON을 선택하고 정책을 편집합니다.

정책을 다음과 같이 대체합니다.

Note

arn:aws:kinesisvideo:*:*:stream/streamName1/* 및 를 이전 단계에서 생성한 스트림의 ARN으로 arn:aws:kinesisvideo:*:*:stream/streamName2/* 바꾸십시오.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kinesisvideo:ListStreams"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "kinesisvideo:DescribeStream",
        "kinesisvideo:PutMedia",
        "kinesisvideo:TagStream",
        "kinesisvideo:GetDataEndpoint"
      ],
      "Resource": [
        "arn:aws:kinesisvideo:*:*:stream/streamName1/*",
        "arn:aws:kinesisvideo:*:*:stream/streamName2/*"
      ]
    }
  ]
}
```

7. Add tags(태그 추가) 페이지에서 Next: Review(다음: 검토)를 선택합니다.
8. 정책 이름을 지정한 다음 Create policy (정책 생성) 를 선택합니다.

정책 이름의 예는 다음과 같습니다 KvsEdgeAccessPolicy.

9. 탭을 닫고 정책을 TES 역할에 연결했던 탭으로 돌아가십시오.

새로 고침 버튼을 선택한 다음 새로 만든 정책을 검색하십시오.

확인란을 선택하고 정책 연결을 선택합니다.

다음 화면에는 정책이 역할에 성공적으로 연결되었다는 메모가 표시됩니다.

10. 다른 정책을 만들어 첨부하세요. 이번에는 비밀을 지키기 위한 것입니다.

정책을 다음과 같이 대체하십시오.

Note

에서 생성한 MediaURI 비밀이 포함된 `arn:aws:secretsmanager:*:*:secret:*` ARN으로 대체하십시오. [the section called “3. IP 카메라 RTSP URL을 위한 리소스 생성”](#)

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "secretsmanager:GetSecretValue",
      "Resource": [
        "arn:aws:secretsmanager:*:*:secret:*",
        "arn:aws:secretsmanager:*:*:secret:*"
      ]
    }
  ]
}
```

11. 이번에는 메트릭에 대한 정책을 하나 더 만들어 첨부하세요. Amazon CloudWatch 정책을 다음과 같이 대체하십시오.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:PutMetricData"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

```

    ]
  }
]
}

```

5단계: 디바이스에 AWS IoT Greengrass 시크릿 매니저 컴포넌트 설치

Amazon Kinesis Video Streams Edge 에이전트를 사용하려면 AWS IoT Greengrass 먼저 시크릿 관리자 구성 요소를 디바이스에 설치해야 합니다.

시크릿 매니저 구성 요소를 설치합니다.

1. <https://console.aws.amazon.com/iot/> 에서 AWS Management Console 로그인하고 AWS IoT Core 콘솔을 엽니다. 적절한 지역이 선택되었는지 확인하십시오.
2. 왼쪽 탐색창에서 Greengrass 디바이스, 배포를 선택합니다.

에서 만든 것과 동일한 타겟을 가진 디플로이먼트를 선택하세요. [the section called “2. AWS IoT Greengrass 코어 디바이스 설정”](#)

3. 오른쪽 상단의 작업 드롭다운에서 수정을 선택합니다.

표시되는 팝업에서 배포 수정을 선택합니다.

4. 다음 섹션을 완료하세요.

- 1단계: 대상을 지정합니다. 다음을 선택합니다.
- 2단계: 구성 요소 선택
 - AWS.Greengrass.cli 구성 요소가 선택되었는지 확인합니다. 이 구성 요소를 제거하지 마십시오.
 - 선택한 구성 요소만 표시 스위치를 토글하고 aws.greengrass를 검색하십시오. SecretManager.
 - aws.greengrass 옆의 체크박스를 체크하세요. SecretManager그런 다음 다음을 선택합니다.
- 3단계: 구성 요소 구성 AWS IoT Greengrass 환경 내에서 암호를 다운로드하도록 AWS IoT Greengrass Secret Manager 구성 요소를 구성합니다.

aws.greengrass를 선택합니다. SecretManager구성 요소를 선택한 다음 구성 요소 구성을 선택합니다.

표시되는 화면에서 병합할 구성 상자에서 AWS Secrets Manager ARN을 업데이트합니다.

Note

에서 생성한 비밀의 `arn:aws:secretsmanager:*:*:secret:*` ARN으로 바꾸십시오. [the section called “3. IP 카메라 RTSP URL을 위한 리소스 생성”](#)

```
{
  "cloudSecrets": [
    {
      "arn": "arn:aws:secretsmanager:*:*:secret:*"
    },
    {
      "arn": "arn:aws:secretsmanager:*:*:secret:*"
    }
  ]
}
```

Note

`cloudSecrets`키가 `arn` 있는 개체의 목록입니다. 자세한 내용은 AWS IoT Greengrass Version 2 개발자 안내서의 [Secret Manager 구성](#) 섹션을 참조하십시오.

완료하면 확인을 선택한 후 다음을 선택합니다.

- 4단계: 고급 설정 구성. 다음을 선택합니다.
 - 5단계: 검토. 배포를 선택합니다.
5. AWS Secrets Manager 구성 요소 및 권한이 올바르게 설치되었는지 확인합니다.

Ubuntu Amazon EC2 인스턴스에서 `sudo /greengrass/v2/bin/greengrass-cli component details --name aws.greengrass.SecretManager` 입력하여 구성 요소가 업데이트된 구성을 수신했는지 확인합니다.

6. 코어 로그를 검사하십시오. AWS IoT Greengrass

`sudo less /greengrass/v2/logs/greengrass.log`를 입력합니다.

배포 오류를 검토하세요.

오류가 발생한 경우 배포를 수정하여 `aws.greengrass.SecretManager` 구성 요소를 제거하세요.

AWS IoT Greengrass 코어 서비스를 다시 `sudo service greengrass restart` 시작하려면 입력합니다.

배포 오류가 권한 누락과 관련된 것이라면 [the section called “4. TES 역할에 권한 추가”](#) 섹션을 검토하여 TES 역할에 적절한 권한이 있는지 확인하십시오. 그런 다음 이 섹션을 반복하세요.

AWS IoT Greengrass 시크릿 매니저 컴포넌트의 시크릿을 업데이트하세요.

⚠ Important

AWS IoT Greengrass Secret Manager 구성 요소는 배포가 업데이트된 경우에만 비밀을 가져와 캐시합니다.

AWS IoT Greengrass Secret Manager 구성 요소의 암호를 업데이트하려면 이전 단계 1~6을 따르고 다음과 같이 변경하십시오.

3단계: 구성 요소 구성 AWS IoT Greengrass 환경 내에서 암호를 다운로드하도록 AWS IoT Greengrass Secret Manager 구성 요소를 구성합니다.

`aws.greengrass`를 선택합니다. SecretManager 구성 요소를 선택한 다음 구성 요소 구성을 선택합니다.

표시되는 화면에서 경로 재설정 [""] 상자에 붙여넣고 병합할 구성 상자에서 AWS Secrets Manager ARN을 업데이트합니다.

자세한 내용은 [업데이트 재설정을](#) 참조하십시오.

6단계: Amazon Kinesis Video Streams Edge AWS IoT Greengrass Agent 구성 요소를 디바이스에 배포합니다.

Amazon Kinesis Video Streams Edge AWS IoT Greengrass Agent 구성 요소를 디바이스에 배포합니다.

1. 제공된 링크를 사용하여 tar 파일을 다운로드합니다.

Amazon Kinesis Video Streams Edge 에이전트 관심 양식을 작성한 경우 이메일에서 다운로드 링크를 확인하십시오. 양식을 아직 작성하지 않은 경우 [여기에서](#) 작성하십시오.

2. 체크섬을 확인하세요.
3. 디바이스에서 바이너리와 jar를 추출합니다.

유형: `tar -xvf kvs-edge-agent.tar.gz`.

압축을 풀고 나면 폴더 구조가 다음과 같이 표시됩니다.

```
kvs-edge-agent/LICENSE
kvs-edge-agent/THIRD-PARTY-LICENSES
kvs-edge-agent/pom.xml
kvs-edge-agent/KvsEdgeComponent
kvs-edge-agent/KvsEdgeComponent/recipes
kvs-edge-agent/KvsEdgeComponent/recipes/recipe.yaml
kvs-edge-agent/KvsEdgeComponent/artifacts
kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/edge_log_config

kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/kvs-edge-agent.jar
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/libgstkvssink.so
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/libIngestorPipelineJNI.so
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/lib
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/lib/libcproducer.so
```

```
kvs-edge-agent/KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/EdgeAgentVersion/lib/libKinesisVideoProducer.so
```

Note

릴리스 폴더 이름은 최신 바이너리 릴리스 번호를 반영하는 방식으로 설정해야 합니다. 예를 들어 1.0.0 릴리스에서는 폴더 이름이 1.0.0으로 설정됩니다.

4. 종속성 항아리를 빌드하세요.

Note

kvs-edge-agent.tar.gz에 포함된 jar에는 종속성이 없습니다. 다음 단계를 사용하여 해당 라이브러리를 빌드하십시오.

포함된 kvs-edge-agent 폴더로 이동합니다pom.xml.

mvn clean package를 입력합니다.

그러면 Amazon Kinesis Video Streams Edge 에이전트에 필요한 종속성이 포함된 jar 파일이 생성됩니다. kvs-edge-agent/target/libs.jar

5. 구성 요소의 아티팩트가 들어 있는 폴더에 libs.jar 파일을 배치합니다.

mv ./target/libs.jar ./KvsEdgeComponent/artifacts/
aws.kinesisvideo.KvsEdgeComponent/*EdgeAgentVersion*/를 입력합니다.

6. 선택 사항. 속성을 설정합니다. Amazon Kinesis Video Streams Edge 에이전트는 모드에서 다음과 같은 환경 변수를 받아들입니다 AWS IoT Greengrass .

환경 변수 이름	필수	설명
AWS_REGION	예	사용되는 지역. 예: us-west-2 AWS IoT Greengrass Core 소프트웨어가 자동으로 이 값을 설정합니다. 자세한 내

환경 변수 이름	필수	설명
GST_PLUGIN_PATH	예	<p>용은 AWS IoT Greengrass Version 2 개발자 안내서의 구성 요소 환경 변수 참조 항목을 참조하십시오.</p> <p>gstkvssink 및 IngestorPipelineJNI 플랫폼 종속 라이브러리가 들어 있는 폴더를 가리키는 파일 경로입니다. 이렇게 하면 GStreamer가 이러한 플러그인을 로드할 수 있습니다. 자세한 내용은 GStreamer 요소 다운로드, 빌드 및 구성을 참조하십시오.</p> <p>예: <i>/download-location /kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesis.video.KvsEdgeComponent/ EdgeAgent Version /</i></p>

환경 변수 이름	필수	설명
LD_LIBRARY_PATH	예	<p>cproducer 및 KinesisVideoProducer 플랫폼 종속 라이브러리가 포함된 디렉터리를 가리키는 파일 경로입니다.</p> <p>예: <i>/download-location /kvs-edge-agent/KvsEdgeComponent/artifacts/aws.kinesisvideo.KvsEdgeComponent/ EdgeAgent Version /lib/</i></p>
AWS_KVS_EDGE_CLOUD_WATCH_ENABLED	아니요	<p>Amazon Kinesis Video Streams Edge 에이전트가 작업 상태 지표를 Amazon CloudWatch에 게시할지 여부를 결정합니다.</p> <p>허용되는 값: TRUE/FALSE(대소문자 구분 안 함). 제공되지 않은 FALSE 경우 기본값은 입니다.</p> <p>예: 거짓</p>

환경 변수 이름	필수	설명
AWS_KVS_EDGE_LOG_LEVEL	아니요	<p>Amazon Kinesis Video Streams Edge Agent 출력의 로깅 수준입니다.</p> <p>허용되는 값:</p> <ul style="list-style-type: none"> • OFF • ALL • 치명적입니다 • ERROR • WARN • 정보, 기본값 (제공되지 않은 경우) • DEBUG • TRACE <p>예: INFO</p>
AWS_KVS_EDGE_LOG_MAX_FILE_SIZE	아니요	<p>로그 파일이 이 크기에 도달하면 롤오버가 발생합니다.</p> <ul style="list-style-type: none"> • 최소: 1 • 최대: 100개 • 기본값: 20 (제공되지 않은 경우) • 단위: 메가바이트 (MB) <p>예: 5</p>

환경 변수 이름	필수	설명
AWS_KVS_EDGE_LOG_OUTPUT_DIRECTORY	아니요	<p>Amazon Kinesis Video Streams Edge 에이전트 로그가 출력되는 디렉토리를 가리키는 파일 경로입니다. 지정하지 않은 <code>./log</code> 경우 기본값은 <code>./log</code> 입니다.</p> <p>예: <code>/file/path/</code></p>
AWS_KVS_EDGE_LOG_ROLLOVER_COUNT	아니요	<p>삭제하기 전에 보관할 롤오버된 로그의 수입니다.</p> <ul style="list-style-type: none"> • 최소: 1 • 최대: 100개 • 기본값: 10 (제공되지 않은 경우) <p>예: 20</p>
AWS_KVS_EDGE_RECORDING_DIRECTORY	아니요	<p>기록된 미디어가 기록될 디렉토리를 가리키는 파일 경로. 지정하지 않은 경우 현재 디렉토리가 기본값입니다.</p> <p>예: <code>/file/path/</code></p>
GREENGRASS_ROOT_DIRECTORY	아니요	<p>AWS IoT Greengrass 루트 디렉토리의 파일 경로.</p> <p>지정하지 않은 <code>/greengrass/v2/</code> 경우 기본값은 <code>./greengrass/v2/</code> 입니다.</p> <p>예: <code>/file/path/</code></p>

환경 변수 이름	필수	설명
GST_DEBUG	아니요	출력할 GStreamer 로그의 수준을 지정합니다. 자세한 내용은 GStreamer 설명서를 참조하십시오. 예: 0
GST_DEBUG_FILE	아니요	GStreamer 디버그 로그의 출력 파일을 지정합니다. 설정하지 않으면 디버그 로그가 표준 오류로 출력됩니다. 자세한 내용은 GStreamer 설명서를 참조하십시오. 예: <code>/tmp/gstreamer-logging.log</code>

실행 스크립트를 `kvs-edge-agent/KvsEdgeComponent/recipes/recipe.yaml` 열고 수정하여 위의 환경 변수를 추가합니다.

Important

수정된 실행 스크립트에 탭 문자가 포함되어 있지 않은지 확인하십시오. AWS IoT Greengrass 핵심 소프트웨어는 레시피를 읽을 수 없습니다.

7. Amazon Kinesis Video Streams Edge AWS IoT Greengrass Agent 구성 요소를 배포하십시오.

유형:

```
sudo /greengrass/v2/bin/greengrass-cli deployment create \
  --recipeDir <download location>/kvs-edge-agent/KvsEdgeComponent/recipes/ \
  --artifactDir <download location>/kvs-edge-agent/KvsEdgeComponent/artifacts/ \
  --merge "aws.kinesisvideo.KvsEdgeComponent=EdgeAgentVersion"
```

자세한 내용은 AWS IoT Greengrass Version 2 개발자 안내서의 다음 섹션을 참조하십시오.

- [AWS IoT Greengrass CLI 명령](#)
- [디바이스에 AWS IoT Greengrass 구성 요소 배포](#)

8. 를 사용하여 애플리케이션에 구성을 전송합니다 AWS CLI.

- a. 새 파일 생성, *example-edge-configuration.json*.

다음 코드를 파일에 붙여 넣습니다. 디바이스의 시스템 시간에 따라 매일 오전 9:00:00 부터 오후 4:59:59 까지 기록하는 샘플 구성입니다. AWS IoT 또한 매일 오후 7:00:00 부터 9:59:59 까지 녹화된 미디어를 업로드합니다.

자세한 정보는 [the section called "StartEdgeConfigurationUpdate"](#)을 참조하세요.

```
{
  "StreamARN": "arn:aws:kinesisvideo:your-region:your-account-id:stream/your-stream/0123456789012",
  "EdgeConfig": {
    "HubDeviceArn": "arn:aws:iot:your-region:your-account-id:thing/kvs-edge-agent-demo",
    "RecorderConfig": {
      "MediaSourceConfig": {
        "MediaUriSecretArn": "arn:aws:secretsmanager:your-region:your-account-id:secret:your-secret-dRbHJQ",
        "MediaUriType": "RTSP_URI"
      },
      "ScheduleConfig": {
        "ScheduleExpression": "0 0 9,10,11,12,13,14,15,16 ? * * **",
        "DurationInSeconds": 3599
      }
    },
    "UploaderConfig": {
      "ScheduleConfig": {
        "ScheduleExpression": "0 0 19,20,21 ? * * **",
        "DurationInSeconds": 3599
      }
    },
    "DeletionConfig": {
      "EdgeRetentionInHours": 15,
      "LocalSizeConfig": {
        "MaxLocalMediaSizeInMB": 2800,
        "StrategyOnFullSize": "DELETE_OLDEST_MEDIA"
      }
    }
  }
}
```

```

        "DeleteAfterUpload": true
    }
}
}

```

- b. 파일을 Amazon Kinesis Video Streams Edge 에이전트로 AWS CLI 보내려면 에 다음을 입력합니다.

```

aws kinesishvideo start-edge-configuration-update --cli-input-json
"file://example-edge-configuration.json"

```

9. Amazon Kinesis Video Streams Edge 에이전트의 각 스트림에 대해 이전 단계를 반복합니다.

7단계: (선택 사항) 디바이스에 AWS IoT Greengrass 로그 관리자 구성 요소 설치

Note

[CloudWatch 할당량](#)을 잘 알고 있어야 합니다.

다음 절차에 따라 Amazon Kinesis Video Streams Edge Agent 로그를 로그 관리자 구성 요소를 사용하여 AWS IoT Greengrass 자동으로 CloudWatch 업로드하도록 구성하십시오.

AWS IoT Greengrass 로그 관리자 구성 요소를 설치합니다.

1. AWS IoT Greengrass 장치 역할에 [적절한 권한이 있는지](#) 확인합니다.
 - a. <https://console.aws.amazon.com/iam/> 에서 AWS Management Console 로그인하고 IAM 콘솔을 엽니다.
 - b. 왼쪽 탐색창에서 역할을 클릭합니다.
 - c. 에서 생성한 TES 역할의 이름을 선택합니다 [the section called "2. AWS IoT Greengrass 코어 디바이스 설정"](#). 필요한 경우 검색 창을 사용하세요.
 - d. GreengrassV2TokenExchangeRoleAccess 정책을 선택합니다.
 - e. JSON 탭을 선택하고 정책이 다음과 같은지 확인합니다.

```

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

```

```

    "Statement": [
      {
        "Effect": "Allow",
        "Action": [
          "logs:CreateLogGroup",
          "logs:CreateLogStream",
          "logs:PutLogEvents",
          "logs:DescribeLogStreams",
          "s3:GetBucketLocation"
        ],
        "Resource": "*"
      }
    ]
  }
}

```

- f. GreengrassV2TokenExchangeRoleAccess정책이 존재하지 않거나 일부 필수 권한이 누락된 경우 이러한 권한으로 새 IAM 정책을 생성하고 에서 생성한 TES 역할에 연결하십시오. [the section called “2. AWS IoT Greengrass 코어 디바이스 설정”](#)
2. AWS Management Console 로그인하고 <https://console.aws.amazon.com/iot/> 에서 AWS IoT Core 콘솔을 엽니다. 적절한 지역이 선택되었는지 확인하십시오.
3. 왼쪽 탐색창에서 Greengrass 디바이스, 배포를 선택합니다.

에서 생성한 것과 동일한 타겟을 가진 디플로이먼트를 선택하세요. [the section called “2. AWS IoT Greengrass 코어 디바이스 설정”](#)
4. 오른쪽 상단에서 작업을 선택한 다음 수정을 선택합니다.

표시되는 팝업에서 배포 수정을 선택합니다.
5. 다음 섹션을 완료하세요.
 - a. 1단계: 대상을 지정합니다. 다음을 선택합니다.
 - b. 2단계: 구성 요소 선택
 - i. AWS.Greengrass.CLI 구성 요소와 AWS.greengrass를 확인하십시오. SecretManager 구성 요소는 아직 선택되어 있습니다.

⚠ Important

이러한 구성 요소는 제거하지 마세요.

- ii. 선택한 구성 요소만 표시 스위치를 토글하고 `aws.greengrass`를 검색하십시오. `LogManager`.
- iii. `aws.greengrass` 옆의 상자를 선택합니다. `LogManager` 그런 다음 다음을 선택합니다.
- c. 3단계: 구성 요소 구성 Amazon Kinesis Video Streams Edge 에이전트에서 생성된 로그를 업로드하도록 AWS IoT Greengrass 로그 관리자 구성 요소를 구성합니다.

`aws.greengrass`를 선택합니다. `LogManager` 구성 요소를 선택한 다음 구성 요소 구성을 선택합니다.

표시되는 화면에서 병합할 구성 상자에 다음 로그 관리자 구성을 붙여넣습니다.

```
{
  "logsUploaderConfiguration": {
    "componentLogsConfigurationMap": {
      "aws.kinesisvideo.KvsEdgeComponent/java_kvs.log": {
        "diskSpaceLimit": "100",
        "diskSpaceLimitUnit": "MB",
        "logFileDirectoryPath": "/greengrass/v2/work/
aws.kinesisvideo.KvsEdgeComponent/log",
        "logFileRegex": "java_kvs.log\\w*"
      },
      "aws.kinesisvideo.KvsEdgeComponent/cpp_kvs_edge.log": {
        "diskSpaceLimit": "100",
        "diskSpaceLimitUnit": "MB",
        "logFileDirectoryPath": "/greengrass/v2/work/
aws.kinesisvideo.KvsEdgeComponent/log",
        "logFileRegex": "cpp_kvs_edge.log\\w*"
      },
      "aws.kinesisvideo.KvsEdgeComponent/cpp_kvssink.log": {
        "diskSpaceLimit": "100",
        "diskSpaceLimitUnit": "MB",
        "logFileDirectoryPath": "/greengrass/v2/work/
aws.kinesisvideo.KvsEdgeComponent/log",
        "logFileRegex": "cpp_kvssink.log\\w*"
      },
      "aws.kinesisvideo.KvsEdgeComponent/cpp_kvs_streams.log": {
        "diskSpaceLimit": "100",
        "diskSpaceLimitUnit": "MB",
        "logFileDirectoryPath": "/greengrass/v2/work/
aws.kinesisvideo.KvsEdgeComponent/log",
        "logFileRegex": "cpp_kvs_streams.log\\w*"
      }
    }
  }
}
```

```

    }
  },
  "periodicUploadIntervalSec": "1"
}

```

⚠ Important

위 `logFileDirectoryPath` 구성에서는 기본 로깅 출력 위치가 사용된다고 가정합니다.

ℹ Note

로그 관리자 구성의 각 매개 변수에 대한 자세한 내용은 AWS IoT Greengrass Version 2 개발자 안내서의 [로그 관리자](#) 섹션을 참조하십시오.

작업을 마치면 확인을 선택하고 다음을 선택합니다.

- d. 4단계: 고급 설정을 구성합니다. 다음을 선택합니다.
 - e. 5단계: 검토. 배포를 선택합니다.
6. AWS 로그 관리자 구성 요소 및 권한이 올바르게 설치되었는지 확인합니다.
 7. Ubuntu Amazon EC2 인스턴스에서 `sudo /greengrass/v2/bin/greengrass-cli component details --name aws.greengrass.LogManager` 입력하여 구성 요소가 업데이트된 구성을 수신했는지 확인합니다.
 8. 코어 로그를 검사하십시오. AWS IoT Greengrass

`sudo less /greengrass/v2/logs/greengrass.log`를 입력합니다.

배포 오류를 검토하세요.

오류가 발생한 경우 배포를 수정하여 `aws.greengrass.LogManager` 구성 요소를 제거하세요.

AWS IoT Greengrass 코어 서비스를 다시 `sudo service greengrass restart` 시작하려면 입력합니다.

배포 오류가 권한 누락과 관련된 것이라면 TES 역할에 적절한 권한이 있는지 [the section called "4. TES 역할에 권한 추가"](#) 검토하십시오. 그런 다음 이 섹션을 반복하세요.

아마존 Kinesis Video Streams 엣지 에이전트 FAQ

다음은 Amazon Kinesis Video Streams Edge 에이전트 서비스에 대한 몇 가지 일반적인 질문입니다.

Amazon Kinesis Video Streams Edge 에이전트는 어떤 운영 체제를 지원합니까?

Amazon Kinesis Video Streams Edge Agent는 현재 다음 운영 체제를 지원합니다.

Ubuntu

- 22.x
 - AMD64
- 18.x
 - ARM

AL2

- amzn2
 - AMD64 아마존 리눅스:2.0.20210219.0-amd64 (Snowball)

Amazon Kinesis Video Streams Edge 에이전트는 H.265 미디어를 지원합니까?

Amazon Kinesis Video Streams Edge 에이전트는 H.264 기본 스트림만 지원합니다.

Amazon Kinesis Video Streams Edge 에이전트는 AL2에서 작동합니까?

예.

사물 또는 장치 내에서 여러 스트림을 실행하려면 어떻게 AWS IoT 해야 합니까?

HubDeviceArn 같지만 다른 [the section called "StartEdgeConfigurationUpdate"](#) Amazon Kinesis 비디오 AWS Secrets Manager 스트림/ARN으로 다른 파일을 전송하십시오.

전송된 **StartEdgeConfigurationUpdate** 후 편집하려면 어떻게 해야 합니까?

동일한 Amazon Kinesis Video Streams ARN을 HubDeviceArn 사용하여 동일한 계정으로 업데이트를 [the section called "StartEdgeConfigurationUpdate"](#) 전송하십시오. 애플리케이션이 Amazon Kinesis Video Streams로부터 메시지를 수신하면 해당 스트림의 이전 구성을 재정의합니다. 그러면 변경 사항이 적용됩니다.

공통적인 예가 **ScheduleConfigs** 있으신가요?

Amazon Kinesis Video Streams Edge 에이전트는 에이전트가 실행되는 디바이스의 시스템 시간을 사용합니다.

설명	ScheduleExpression	DurationInSeconds
연중무휴 녹화, 시간별 업로드	(null ScheduleConfig)	
매일 오전 9시~오후 4시 59분 59분	0 0 9-16 * * ? *	3599
평일 오전 9:00:00 - 오후 4:59:59	0 0 9-16 ? * 2-6 *	3599
	0 0 9-16 ? * 2,3,4,5,6 *	3599
	0 0 9-16 ? * MON-FRI *	3599
	0 0 9-16 ? * MON,TUE,WED,THU,FRI *	3599
주말 오전 9:00:00 - 오후 4:59:59	0 0 9-16 ? * SAT,SUN *	3599
평일 오후 10:00:00 - 오후 11:59:59	0 0 22,23 ? * MON-FRI *	3599
매일 오전 9시~오전 10시	0 0 9 * * ? *	3600
매일 오후 4시~오후 5시 59분 59분	0 0 16-17 * * ? *	3599

더 많은 예는 [Quartz 설명서를](#) 참조하십시오.

최대 스트리밍 한도가 있나요?

Amazon Kinesis Video Streams Edge 에이전트는 현재 디바이스당 스트림 16개로 엄격하게 제한되어 있습니다. [the section called “DeleteEdgeConfiguration”](#) API를 사용하여 디바이스에서 스트림을 삭제하십시오. 를 사용하여 동일한 스트림의 구성을 업데이트해도 기기의 스트림 수는 증가하지 [the section called “StartEdgeConfigurationUpdate”](#) 않습니다.

오류가 발생한 작업을 다시 시작하려면 어떻게 해야 합니까?

오류가 발생하는 경우 Amazon Kinesis Video Streams Edge 에이전트는 작업 재시작을 시도합니다. 하지만 일부 오류 (예: 구성 오류) 가 있는 경우 작업을 수동으로 다시 시작해야 합니다.

수동으로 다시 시작해야 하는 작업을 결정하려면 의 FatalError [the section called “를 사용하여 Amazon Kinesis Video Streams 엣지 에이전트를 모니터링하기 CloudWatch”](#) 지표를 참조하십시오.

를 다시 [the section called “StartEdgeConfigurationUpdate”](#) 전송하여 스트림에 대한 작업을 다시 시작하십시오.

Amazon Kinesis Video Streams Edge 에이전트의 상태를 모니터링하려면 어떻게 해야 합니까?

자세한 내용은 [the section called “를 사용하여 Amazon Kinesis Video Streams 엣지 에이전트를 모니터링하기 CloudWatch”](#)을(를) 참조하세요.

VPC를 통한 비디오 스트리밍

이 베타 버전은 유럽 (파리) 지역, eu-west-3에서 프리뷰로 제공됩니다. [이러한 구성 요소 및 시작 안내서에 액세스하려면 이메일을 보내주세요.](#)

Amazon Kinesis Video Streams VPC 엔드포인트 서비스를 사용하면 데이터가 퍼블릭 인터넷을 거치지 않고도 Amazon 네트워크를 통해 비디오를 스트리밍하고 사용할 수 있습니다.

액세스를 요청하려면 다음 정보를 [이메일로 보내주세요.](#)

- 계정 ID
- 스트림 ARN
- VPC ID

Note

서비스에 사용자를 추가하는 데 최대 1주일이 걸릴 수 있습니다.

과거에 VPC 엔드포인트를 사용해 본 적이 없다면 다음 정보를 검토하여 개념을 익히십시오.

- [AWS PrivateLink 배경](#)
- [VPC 시작 가이드](#)

추가 정보

베타에 추가되면 이 기능에 대한 추가 정보 링크를 이메일로 보내드립니다.

VPC 엔드포인트 프로시저

할당량

주요 할당량 차이는 다음과 같습니다.

- 모든 대역폭 API의 낮은 할당량 (2mbps):

- PutMedia
- GetMedia
- GetMediaForFragmentList
- 고객당 스트림 10개 허용

엔드포인트 생성

허용 목록에 등록되면 Amazon Kinesis Video Streams의 VPC 엔드포인트 서비스 이름을 받게 됩니다. 다음과 같이 보일 것입니다. `com.amazonaws.region.kinesisvideo`

Amazon [VPC 콘솔 또는 \(\)](#) 를 사용하여 [Amazon Kinesis Video Streams용 인터페이스 VPC 엔드포인트](#)를 생성합니다. AWS Command Line Interface AWS CLI

에 AWS CLI다음을 입력합니다.

```
aws ec2 create-vpc-endpoint \
--vpc-id customer-provided-vpc-id\
--service-name com.amazonaws.eu-west-2.kinesisvideo \
--private-dns-enabled
```

Important

VPC 내의 트래픽은 프라이빗 DNS를 사용하여 엔드포인트를 통해 라우팅합니다. 이 기능을 활성화하지 않으면 자체 DNS 로직을 구현해야 합니다. 프라이빗 DNS에 대한 자세한 내용은 [AWS PrivateLink 설명서](#)를 참조하십시오.

AWS CLI 옵션에 대한 자세한 내용은 을 참조하십시오 [create-vpc-endpoint](#).

엔드포인트에 대한 액세스 제어

Amazon Kinesis Video Streams에 대한 액세스를 제어하는 엔드포인트 정책을 VPC 엔드포인트에 연결할 수 있습니다. 이 정책은 다음 정보를 지정합니다.

- 작업을 수행할 수 있는 보안 주체는
- 수행할 수 있는 작업 및
- 작업을 수행할 수 있는 리소스.

자세한 내용은 가이드의 엔드포인트 [정책을 사용하여 VPC 엔드포인트로 서비스에 대한 액세스 제어를 참조하십시오](#). AWS PrivateLink

다음은 Amazon Kinesis Video Streams에 대한 엔드포인트 정책의 예입니다. 엔드포인트에 연결된 경우 이 정책은 모든 리소스의 모든 보안 주체에 대해 나열된 PutMedia 작업에 대한 액세스를 거부합니다.

```
{
  "Statement": [
    {
      "Principal": "*",
      "Effect": "Deny",
      "Action": [
        "kinesisvideo:PutMedia"
      ],
      "Resource": "*"
    }
  ]
}
```

Kinesis 비디오 스트림의 이미지

Amazon Kinesis 비디오 스트림 API 및 SDK를 사용하여 비디오 스트림에서 이미지를 추출하는 데 도움을 줄 수 있습니다. 썸네일이나 향상된 스크러빙과 같은 향상된 재생 애플리케이션에 이러한 이미지를 사용하거나 기계 학습 파이프라인에서 사용할 수 있습니다. Kinesis Video Streams는 API를 통한 온디맨드 이미지 추출 또는 인제스트된 비디오의 메타데이터 태그에서 자동 이미지 추출 기능을 제공합니다.

Kinesis Video Streams 관리형 이미지 지원 사용에 대한 자세한 내용은 다음을 참조하십시오.

- [온디맨드 이미지 생성 \(GetImages\)](#)- 이 API를 통해 고객은 Kinesis Video Streams에 저장된 비디오에서 단일 이미지 또는 여러 이미지를 추출할 수 있습니다.
- [자동 이미지 생성 \(S3 전송\)](#) - 업로드된 비디오의 태그를 기반으로 실시간으로 비디오 데이터에서 이미지를 자동으로 추출하여 고객이 지정한 S3 버킷으로 이미지를 전송하도록 Kinesis Video Streams를 구성합니다.

주제

- [GetImages 시작하기](#)
- [Amazon S3 전송 시작하기](#)

GetImages 시작하기

이미지 관리형 지원은 Kinesis Video Streams에서 스트리밍되고 저장된 비디오 데이터에서 이미지를 가져올 수 있는 완전 관리형 방법을 제공합니다. 이미지를 사용하여 사람, 애완동물 또는 차량 감지와 같은 기계 학습 (ML) 워크로드를 실행할 수 있습니다. 이미지를 사용하여 모션 이벤트에 대한 이미지 미리 보기 및 비디오 클립의 스크러빙과 같은 인터랙티브 요소를 재생에 추가할 수도 있습니다.

에 대한 자세한 내용은 [GetImages](#) 기능, 참조 [GetImages](#)에서 아마존 키네시스 비디오 스트리밍 아카이브 미디어 API 참조 가이드.

Amazon S3 전송 시작하기

현재 고객은 자체 이미지 트랜스코딩 파이프라인을 실행 및 관리하여 스크러빙, 이미지 미리 보기, 이미지에서 ML 모델 실행 등과 같은 다양한 목적으로 이미지를 생성합니다. Kinesis Video Streams는 이미지를 트랜스코딩하여 전송하는 기능을 제공합니다. Kinesis Video Streams는 태그를 기반으로 실시

간으로 비디오 데이터에서 이미지를 자동으로 추출하여 고객이 지정한 S3 버킷으로 이미지를 전송합니다.

UpdateImageGenerationConfiguration

Amazon S3에 이미지를 생성할 수 있도록 Kinesis 비디오 스트림을 설정하려면:

1. 생성하기S3 버킷새 API를 사용하여 SDK에 추가된 태그를 기반으로 이미지를 생성할 수 있습니다. 참고 사항S3 URI이는 다음 단계에서 스트림의 이미지 생성 구성을 업데이트할 때 필요합니다.
2. 라는 JSON 파일을 생성합니다.update-image-generation-input.json다음 내용을 입력합니다.

```
{
  "StreamName": "TestStream",
  "ImageGenerationConfiguration": {
    {
      "Status": "ENABLED",
      "DestinationConfig": {
        {
          "DestinationRegion": "us-east-1",
          "Uri": "s3://bucket-name"
        },
        "SamplingInterval": 200,
        "ImageSelectorType": "PRODUCER_TIMESTAMP",
        "Format": "JPEG",
        "FormatConfig": {
          "JPEGQuality": "80"
        },
        "WidthPixels": 320,
        "HeightPixels": 240
      }
    }
  }
}
```

사용할 수 있습니다AWS CLI호출하려면[UpdateImageGenerationConfiguration](#)이전에 생성한 Amazon S3 ARN을 추가하고 상태를 다음과 같이 변경하는 API 작업ENABLED.

```
aws kinesisisvideo update-image-generation-configuration \
--cli-input-json file://./update-image-generation-input.json \
```

요청:

UpdateImageGenerationConfiguration HTTP/1.1

```

Method: 'POST'
Path: '/updateImageGenerationConfiguration'
Body: {
  StreamName: 'String', // Optional. Either stream name or arn should be passed
  StreamArn: 'String', // Optional. Either stream name or arn should be passed
  ImageGenerationConfiguration : {
    // required
    Status: 'Enum', // ENABLED | DISABLED,
    ImageSelectorType: 'Enum', // SERVER_TIMESTAMP | PRODUCER_TIMESTAMP..
    DestinationConfig: {
      DestinationRegion: 'String',
      Uri: string,
    },
    SamplingInterval: 'Number'//
    Format: 'Enum', // JPEG | PNG
    // Optional parameters
    FormatConfig: {
      'String': 'String',
    },
    WidthPixels: 'Number', // 1 - 3840 (4k).
    HeightPixels: 'Number' // 1 - 2160 (4k).
  }
}


```

응답:

```

HTTP/1.1 200
Content-type: application/json
Body: {
}

```

 Note

이미지 생성 구성을 업데이트한 후 이미지 생성 워크플로를 시작하는 데 최소 1분이 걸립니다. 호출하기 전에 최소 1분 정도 기다리세요. PutMedia 업데이트 호출 후

DescribeImageGenerationConfiguration

스트림에 대해 이미 설정된 이미지 생성 구성을 보려면 고객이 다음을 수행할 수 있습니다. DescribeImageGenerationConfiguration 다음과 같이 요청하십시오.

요청:

```
DescribeImageGenerationConfiguration HTTP/1.1
```

```
Method: 'POST'
Path: '/describeImageGenerationConfiguration'
Body: {
  StreamName: 'String', // Optional. Either stream name or arn should be passed
  StreamArn: 'String', // Optional. Either stream name or arn should be passed
}
```

응답:

```
HTTP/1.1 200
Content-type: application/json
Body: {
  ImageGenerationConfiguration : {
    Status: 'Enum',
    ImageSelectorType: 'Enum', // SERVER_TIMESTAMP | PRODUCER_TIMESTAMP
    DestinationConfig: {
      DestinationRegion: 'String'
      Uri: 'string',
    },
    SamplingInterval: 'Number',
    Format: 'Enum',
    FormatConfig: {
      'String': 'String',
    },
    WidthPixels: 'Number',
    HeightPixels: 'Number'
  }
}
```

에 대해 자세히 알아보려면 `DescribeImageGenerationConfiguration` 기능, 참조 [DescribeImageGenerationConfiguration](#) 에서 아마존 키네시스 비디오 스트림 개발자 가이드.

프로듀서 MKV 태그

Kinesis Video Streams Producer SDK를 사용하면 SDK에서 API 작업을 호출하여 관심 있는 특정 프래그먼트에 태그를 지정할 수 있습니다. 태그의 예는 다음을 참조하십시오. [이 코드](#). 이 API를 호출하면 SDK는 미리 정의된 MKV 태그 세트를 프래그먼트 데이터와 함께 추가합니다. Kinesis Video Streams는 이러한 특수 MKV 태그를 인식하고 해당 스트림의 이미지 처리 구성을 기반으로 이미지 생성 워크플로를 시작합니다.

Amazon S3 이미지 생성 태그와 함께 제공된 모든 프래그먼트 메타데이터는 Amazon S3 메타데이터로 저장됩니다.

프로듀서 MKV 태그의 구문

```
|+ Tags
| + Tag
| // MANDATORY: Predefined MKV tag to trigger image generation for the fragment
| + Simple
| + Name: AWS_KINESISVIDEO_IMAGE_GENERATION

| // OPTIONAL: S3 prefix which will be set as prefix for generated image.
| + Simple
| + Name: AWS_KINESISVIDEO_IMAGE_PREFIX
| + String: image_prefix_in_s3 // 256 bytes max m

| // OPTIONAL: Key value pairs that will be persisted as S3 Image object metadata.
| + Simple
| + Name: CUSTOM_KEY_1 // Max 128 bytes
| + String: CUSTOM_VALUE_1 // Max 256 bytes
| + Simple
| + Name: CUSTOM_KEY_2 // Max 128 bytes
| + String: CUSTOM_VALUE_2 // Max 256 bytes
```

다음을 사용하여 Producer SDK에 메타데이터 태그 추가 `PutEventMetaData`

`PutEventMetaData` 함수는 이벤트와 관련된 MKV 파일을 추가합니다. `PutEventMetaData` 두 개의 매개 변수를 사용합니다. 첫 번째 파라미터는 값이 다음에서 오는 이벤트입니다.

다.STREAM_EVENT_TYPE열거형. 두 번째 파라미터,[pStreamEventMetadata](#)는 선택 사항이며 추가 메타데이터를 키-값 쌍으로 포함하는 데 사용할 수 있습니다. 추가할 수 있는 메타데이터의 키-값 쌍은 5개로 제한됩니다.

제한

다음 표에는 메타데이터 태그와 관련된 제한 사항이 나와 있습니다. 메타데이터 태그 제한을 조정할 수 있는 경우 계정 관리자를 통해 증가를 요청할 수 있습니다.

한도	최대 값	조정 가능
이미지 접두사 길이	256	아니요
선택적 메타데이터 키 길이	128	아니요
선택적 메타데이터 값 길이	256	아니요
선택적 메타데이터의 최대 수	10	예

S3 오브젝트 메타데이터

기본적으로 Kinesis 비디오 스트림은 다음을 설정합니다.프래그먼트 넘버,생산자, 및서버 타임스탬프Amazon S3 객체 메타데이터로 생성된 이미지 MKV 태그에 프래그먼트 데이터가 추가로 지정되면 해당 태그도 Amazon S3 객체 메타데이터에 추가됩니다. 다음 예제는 Amazon S3 객체 메타데이터의 올바른 구문을 보여줍니다.

```
{
  // KVS S3 object metadata
  x-amz-meta-aws_kinesisvideo_fragment_number : 'string',
  x-amz-meta-aws_kinesisvideo_producer_timestamp: 'number',
  x-amz-meta-aws_kinesisvideo_server_timestamp: 'number',

  // Optional key value pair sent as part of the MKV tags
  custom_key_1: custom_value_1,
  custom_key_2: custom_value_2,
}
```

S3 오브젝트 경로 (이미지)

다음 목록은 개체 경로의 올바른 형식을 보여 주고 경로 내의 각 요소를 설명합니다.

형식:

ImagePrefix_## ID_StreamName_ImageTimecode_## ###.## ###

1. ImagePrefix - 가치 AWS_KINESISVIDEO_IMAGE_PREFIX.
2. AccountID - 스트림이 생성되는 계정 ID
3. StreamName - 이미지가 생성되는 스트림의 이름입니다.
4. ImageTimecode - 이미지가 생성되는 프래그먼트의 에포크 타임코드
5. RandomID - 랜덤 GUID.
6. file-extension - 요청된 이미지 형식에 따른 JPG 또는 PNG

스로틀링 방지를 위한 Amazon S3 URI 권장 사항

Amazon S3에 수천 개의 이미지를 쓸 경우 병목 현상이 발생할 위험이 있습니다. 자세한 내용은 [이 링크](#)를 참조하십시오. [S3 프리픽스 풋 요청 제한](#).

Amazon S3 접두사는 초당 3,500개의 PUT 요청의 PUT 한도로 시작하며 시간이 지남에 따라 고유한 접두사에 대한 제한이 점차 늘어납니다. 날짜와 시간을 Amazon S3 접두사로 사용하지 마십시오. 타임 코딩된 데이터는 한 번에 하나의 접두사에 영향을 미치며 정기적으로 변경되므로 이전의 접두사 확장이 무효화됩니다. 더 빠르고 일관된 Amazon S3 확장을 가능하게 하려면 Amazon S3 대상 URI에 16진수 코드나 UUID와 같은 임의의 접두사를 추가하는 것이 좋습니다. 예를 들어 16진수 코드 접두사는 요청을 16개의 서로 다른 접두사 (각 고유한 16진수 문자의 접두사) 로 자연스럽게 분할합니다. 그러면 Amazon S3가 자동 조정된 후 초당 56,000건의 PUT 요청이 허용됩니다.

Kinesis Video Streams의 알림

미디어 프래그먼트를 사용할 수 있게 되면 Kinesis Video Streams는 (Amazon Simple Notification Service Amazon SNS) 알림을 사용하여 고객에게 알립니다. 다음 주제에서는 알림을 시작하는 방법을 설명합니다.

UpdateNotificationConfiguration

이 API 작업을 사용하여 스트림의 알림 정보를 업데이트할 수 있습니다. 이 UpdateNotificationConfiguration 기능에 대한 자세한 내용은 Amazon Kinesis Video Streams 개발자 안내서를 참조하십시오 [UpdateNotificationConfiguration](#).

Note

알림 구성을 업데이트한 후 알림을 시작하는 데 최소 1분이 걸립니다. 업데이트 호출 PutMedia 후 호출하려면 1분 이상 기다리십시오.

DescribeNotificationConfiguration

이 API를 사용하여 스트림에 첨부된 알림 구성을 설명하십시오. 이 DescribeNotificationConfiguration 기능에 대한 자세한 내용은 Amazon Kinesis Video Streams 개발자 안내서를 참조하십시오 [DescribeNotificationConfiguration](#).

프로듀서 MKV 태그

Kinesis Video Streams Producer SDK를 사용하면 SDK에서 API 작업을 호출하여 관심 있는 특정 프래그먼트에 태그를 지정할 수 있습니다. 이 코드 [섹션에서](#) 작동 방식의 샘플을 참조하십시오. 이 API를 호출하면 SDK는 프래그먼트 데이터와 함께 사전 정의된 MKV 태그 세트를 추가합니다. Kinesis Video Streams는 이러한 특수 MKV 태그를 인식하고 태그가 지정된 프래그먼트에 대한 알림을 시작합니다.

알림 MKV 태그와 함께 제공되는 모든 프래그먼트 메타데이터는 Amazon SNS 주제 페이로드의 일부로 게시됩니다.

프로듀서 MKV 태그의 구문

```

|+ Tags
| + Tag
| // MANDATORY: Predefined MKV tag to trigger the notification for the fragment
| + Simple
| + Name: AWS_KINESISVIDEO_NOTIFICATION
| + String
| // OPTIONAL: Key value pairs that will be sent as part of the Notification payload
| + Simple
| + Name: CUSTOM_KEY_1 // Max 128 bytes
| + String:CUSTOM_VALUE_1 // Max 256 bytes
| + Simple
| + Name: CUSTOM_KEY_2 // Max 128 bytes
| + String: CUSTOM_VALUE_2 // Max 256 bytes

```

MKV 태그 제한

다음 표에는 메타데이터 태그와 관련된 제한 사항이 나열되어 있습니다. 메타데이터 태그 한도를 조정할 수 있는 경우 계정 관리자를 통해 증가를 요청할 수 있습니다.

Limit	최대값	조정 가능
선택적 메타데이터 키 길이	128	아니요
선택적 메타데이터 값 길이	256	아니요
선택적 메타데이터의 최대 수	10	예

Amazon SNS 주제 페이로드

이전 워크플로를 통해 시작된 모든 알림은 다음 예와 같이 Amazon SNS 주제 페이로드를 전달합니다. 이 예는 Amazon Simple Queue Service (Amazon SQS) 대기열의 알림 데이터를 사용한 후 발생하는 Amazon SNS 메시지입니다.

```

{
  "Type" : "Notification",
  "MessageId" : Message ID,
  "TopicArn" : SNS ARN,

```

```

"Subject" : "Kinesis Video Streams Notification",
"Message" : "{\"StreamArn\": \"Stream Arn, \"FragmentNumber\": \"Fragment Number,
\"FragmentStartProducerTimestamp\": \"Fragment Start Producer Timestamp,
          \"FragmentStartServerTimestamp\": \"Fragment Start Server Timestamp,
\"NotificationType\": \"PERSISTED\", \"NotificationPayload\": {\" CUSTOM_KEY_1:
\"CUSTOM_VALUE_1,
          \"CUSTOM_KEY_2: \"CUSTOM_VALUE_2}}\",
"Timestamp" : "2022-04-25T18:36:29.194Z",
"SignatureVersion" : Signature Version,
"Signature" : Signature,
"SigningCertURL" : Signing Cert URL,
"UnsubscribeURL" : Unsubscribe URL
}

```

```

Subject: "Kinesis Video Streams Notification"
Message:
{
  "StreamArn": Stream Arn,
  "FragmentNumber": Fragment Number,
  "FragmentStartProducerTimestamp": Fragment Start Producer Timestamp,
  "FragmentStartServerTimestamp": Fragment Start Server Timestamp,
  "NotificationType": "PERSISTED",
  "NotificationPayload": {
    CUSTOM_KEY_1: CUSTOM_VALUE_1,
    CUSTOM_KEY_2: CUSTOM_VALUE_2
  }
}

```

Amazon SNS 메시지 보기

Amazon SNS 주제에서 직접 메시지를 읽을 수는 없습니다. 이를 위한 API가 없기 때문입니다. 메시지를 보려면 SQS 대기열에서 해당 SNS 주제를 구독하거나 다른 [Amazon SNS 지원 대상을](#) 선택하십시오. 하지만 메시지를 보는 데 가장 효율적인 옵션은 Amazon SQS를 사용하는 것입니다.

Amazon SQS를 사용하여 Amazon SNS 메시지를 보려면

1. [Amazon SQS](#) 대기열을 생성합니다.
2. 에서 AWS Management Console 대상으로 설정된 Amazon SNS 주제를 아래에서 엽니다 NotificationConfiguration.
3. [구독 생성] 을 선택한 다음 첫 단계에서 만든 Amazon SQS 대기열을 선택합니다.

4. 알림 구성을 활성화하고 프래그먼트에 알림 MKV 태그를 추가한 상태로 PutMedia 세션을 실행합니다.
5. Amazon SQS 콘솔에서 Amazon SQS 대기열을 선택한 다음 Amazon SQS 대기열에 대한 메시지 전송 및 수신을 선택합니다.
6. 메시지 폴링. 이 명령어는 PutMedia 세션에서 생성된 모든 알림을 표시해야 합니다. 폴링에 대한 자세한 내용은 [Amazon SQS 단기 및 장기 폴링](#)을 참조하십시오.

아마존 키네시스 비디오 스트림의 보안

클라우드 AWS 보안이 최우선 과제입니다. AWS 고객은 가장 보안에 민감한 조직의 요구 사항을 충족하도록 구축된 데이터 센터 및 네트워크 아키텍처를 활용할 수 있습니다.

보안은 두 사람 사이의 AWS 공동 책임입니다. [공동 책임 모델](#)은 이 사항을 클라우드 내 보안 및 클라우드의 보안으로 설명합니다.

- 클라우드 보안 - AWS 클라우드에서 AWS 서비스를 실행하는 인프라를 보호하는 역할을 합니다. AWS 또한 안전하게 사용할 수 있는 서비스를 제공합니다. 서드 파티 감사자는 정기적으로 [AWS 규정 준수 프로그램](#)의 일환으로 보안 효과를 테스트하고 검증합니다. Kinesis Video Streams에 적용되는 규정 준수 프로그램에 대한 자세한 내용은 [규정 준수 프로그램 제공 범위 내의 AWS 서비스를 참조하세요](#).
- 클라우드에서의 보안 — 귀하의 책임은 사용하는 AWS 서비스에 따라 결정됩니다. 또한 데이터의 민감도, 조직의 요건 및 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다.

이 설명서는 Kinesis Video Streams를 사용할 때 공동 책임 모델을 적용하는 방법을 이해하는 데 도움이 됩니다. 다음 항목에서는 보안 및 규정 준수 목표를 충족하도록 Kinesis Video Streams를 구성하는 방법을 보여줍니다. 또한 Kinesis Video Streams 리소스를 모니터링하고 보호하는 데 도움이 될 수 있는 다른 AWS 서비스를 사용하는 방법도 알아봅니다.

주제

- [Kinesis Video Streams의 데이터 보호](#)
- [IAM을 사용하여 Kinesis Video Streams 리소스에 대한 액세스 제어](#)
- [를 사용하여 Kinesis Video Streams 리소스에 대한 액세스 제어 AWS IoT](#)
- [아마존 키네시스 비디오 스트림 모니터링](#)
- [Amazon Kinesis Video Streams에 대한 규정 준수 검증](#)
- [Amazon Kinesis Video Streams의 레질리언스](#)
- [Kinesis Video Streams의 인프라 보안](#)
- [Kinesis Video Streams의 보안 모범 사례](#)

Kinesis Video Streams의 데이터 보호

Amazon Kinesis Video Streams에 저장된 데이터를 암호화하여 엄격한 데이터 관리 요구 사항을 충족하기 위해 AWS Key Management Service (AWS KMS) 키를 사용하는 서버 측 암호화 (SSE) 를 사용할 수 있습니다.

주제

- [Kinesis Video Streams의 서버 측 암호화란 무엇입니까?](#)
- [비용, 지역 및 성능 고려 사항](#)
- [서버 측 암호화를 시작하려면 어떻게 해야 하나요?](#)
- [고객 관리 키 생성 및 사용](#)
- [고객 관리 키를 사용할 수 있는 권한](#)

Kinesis Video Streams의 서버 측 암호화란 무엇입니까?

서버 측 암호화는 사용자가 지정한 키를 사용하여 데이터를 저장하기 전에 데이터를 자동으로 암호화하는 Kinesis Video Streams의 기능입니다. AWS KMS 데이터는 Kinesis Video Streams 스트림 스토리지 계층에 기록되기 전에 암호화되고 스토리지에서 검색된 후에 복호화됩니다. 따라서 Kinesis Video Streams 서비스 내에서 저장된 데이터는 항상 암호화됩니다.

서버 측 암호화를 사용하면 Kinesis 비디오 스트림 제작자와 소비자가 KMS 키 또는 암호화 작업을 관리할 필요가 없습니다. 데이터 보존이 활성화된 경우 Kinesis Video Streams에 들어오고 나가는 데이터가 자동으로 암호화되므로 저장된 데이터가 암호화됩니다. AWS KMS 서버 측 암호화 기능에 사용되는 모든 키를 제공합니다. AWS KMS 서비스로 가져온 사용자 지정 키인 Kinesis Video Streams에서 KMS AWS KMS 키를 효율적으로 사용할 수 있도록 합니다. AWS AWS KMS

비용, 지역 및 성능 고려 사항

서버 측 암호화를 적용할 때는 AWS KMS API 사용 및 키 비용이 적용됩니다. 사용자 지정 AWS KMS 키와 달리 (Default) aws/kinesis-video KMS 키는 무료로 제공됩니다. 하지만 Kinesis Video Streams에서 발생하는 API 사용 비용은 여전히 사용자 대신 지불해야 합니다.

API 사용 비용은 사용자 지정 키를 포함한 모든 KMS 키에 적용됩니다. 각 사용자 자격 증명에는 고유한 API 호출이 필요하기 때문에 데이터 생산자와 소비자에게 사용하는 사용자 자격 증명의 수에 따라 AWS KMS 비용이 조정됩니다. AWS KMS

다음은 리소스별 비용에 대한 설명입니다.

키

- (별칭 `aws/kinesis-video` =) 에서 관리하는 AWS Kinesis Video Streams의 KMS 키에는 요금이 부과되지 않습니다.
- 사용자 생성 KMS 키에는 비용이 부과됩니다. AWS KMS key 자세한 내용은 [AWS Key Management Service 요금](#)을 참조하십시오.

AWS KMS API 사용

새 데이터 암호화 키를 생성하거나 기존 암호화 키를 검색하기 위한 API 요청은 트래픽이 증가함에 따라 증가하며 AWS KMS 사용 비용이 부과됩니다. 자세한 내용은 [AWS Key Management Service 요금: 사용량을](#) 참조하십시오.

Kinesis Video Streams는 보존이 0 (보존 없음) 으로 설정된 경우에도 주요 요청을 생성합니다.

지역별 서버 측 암호화 사용 가능 여부

Kinesis 비디오 스트림의 서버 측 암호화는 Kinesis Video Streams가 제공되는 모든 곳에서 사용할 수 있습니다. AWS 리전

서버 측 암호화를 시작하려면 어떻게 해야 하나요?

Kinesis Video Streams에서는 서버 측 암호화가 항상 활성화되어 있습니다. 스트림이 생성될 때 사용자 제공 키가 지정되지 않은 경우 AWS 관리형 키 (Kinesis Video Streams에서 제공) 가 사용됩니다.

사용자가 제공한 KMS 키는 Kinesis 비디오 스트림이 생성될 때 해당 스트림에 할당되어야 합니다. 나중에 [UpdateStream](#)API를 사용하여 스트림에 다른 키를 할당할 수 없습니다.

다음 두 가지 방법으로 사용자가 제공한 KMS 키를 Kinesis 비디오 스트림에 할당할 수 있습니다.

- 에서 Kinesis 비디오 스트림을 생성할 때는 새 비디오 스트림 생성 페이지의 암호화 탭에서 KMS 키를 지정합니다. AWS Management Console
- [CreateStream](#)API를 사용하여 Kinesis 비디오 스트림을 생성할 때는 파라미터에 키 ID를 지정하십시오. `KmsKeyId`

고객 관리 키 생성 및 사용

이 섹션에서는 Amazon Kinesis Video Streams에서 관리하는 키를 사용하는 대신 자체 KMS 키를 생성하고 사용하는 방법을 설명합니다.

고객 관리형 키 생성

고유한 키를 생성하는 방법에 대한 자세한 내용은 개발자 안내서의 [키 생성](#)을 참조하십시오. AWS Key Management Service 계정으로 키를 생성하면 Kinesis Video Streams 서비스가 고객 관리 키 목록에 이러한 키를 반환합니다.

고객 관리형 키 사용

소비자, 생산자 및 관리자에게 올바른 권한이 적용된 후에는 사용자 지정 KMS 키를 자체 AWS 계정 또는 다른 곳에서 사용할 수 있습니다. AWS 계정계정의 모든 KMS 키는 콘솔의 고객 관리 키 목록에 표시됩니다.

다른 계정에 있는 사용자 지정 KMS 키를 사용하려면 해당 키를 사용할 권한이 있어야 합니다. 또한 CreateStream API를 사용하여 스트림을 생성해야 합니다. 콘솔에서 만든 스트림에서는 다른 계정의 KMS 키를 사용할 수 없습니다.

Note

PutMedia 또는 GetMedia 작업을 수행할 때까지는 KMS 키에 액세스할 수 없습니다. 결과는 다음과 같습니다.

- 지정한 키가 존재하지 않는 경우 CreateStream 작업은 성공하지만 PutMedia 스트림에서의 GetMedia 작업은 실패합니다.
- 제공된 키 (aws/kinesis-video) 를 사용하는 경우 첫 번째 PutMedia 또는 GetMedia 작업이 수행될 때까지 키가 계정에 표시되지 않습니다.

고객 관리 키를 사용할 수 있는 권한

고객 관리 키로 서버 측 암호화를 사용하려면 먼저 스트림의 암호화와 스트림 레코드의 암호화 및 암호 해독을 허용하도록 KMS 키 정책을 구성해야 합니다. 권한에 대한 예와 자세한 내용은 [AWS KMS API AWS KMS 권한](#): 작업 및 리소스 참조를 참조하십시오.

Note

암호화에 기본 서비스 키를 사용할 때는 사용자 지정 IAM 권한을 적용할 필요가 없습니다.

고객 관리 키를 사용하기 전에 Kinesis 비디오 스트림 제작자 및 소비자 (IAM 주체) 가 기본 키 정책의 사용자인지 확인하십시오. AWS KMS 그렇지 않으면 스트림에서 읽기 및 쓰기가 실패하여 궁극적으로 데이터 손실, 처리 지연 또는 애플리케이션 중단이 발생할 수 있습니다. IAM 정책을 사용하여 KMS 키 권한을 관리할 수 있습니다. 자세한 내용은 IAM 정책 [사용을](#) 참조하십시오. AWS KMS

프로듀서 권한 예시

Kinesis 비디오 스트림 제작자는 다음과 같은 허가를 받아야 합니다. kms:GenerateDataKey

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKey"
      ],
      "Resource": "arn:aws:kms:us-west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    {
      "Effect": "Allow",
      "Action": [
        "kinesis-video:PutMedia",
      ],
      "Resource": "arn:aws:kinesis-video:*:123456789012:MyStream"
    }
  ]
}
```

소비자 권한 예시

Kinesis 비디오 스트림 소비자는 다음과 같은 권한을 가져야 합니다. kms:Decrypt

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
    }
  ]
}
```

```

    "Resource": "arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  },
  {
    "Effect": "Allow",
    "Action": [
      "kinesis-video:GetMedia",
    ],
    "Resource": "arn:aws:kinesis-video:*:123456789012:MyStream"
  }
]
}

```

IAM을 사용하여 Kinesis Video Streams 리소스에 대한 액세스 제어

Amazon Kinesis Video Streams와 함께 AWS Identity and Access Management (IAM) 을 사용하여 조직의 사용자가 특정 Kinesis Video Streams API 작업을 사용하여 작업을 수행할 수 있는지 여부와 특정 리소스를 사용할 수 있는지 여부를 제어할 수 있습니다. AWS

IAM에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Identity and Access Management \(IAM\)](#)
- [시작하기](#)
- [IAM 사용 설명서](#)

내용

- [정책 구문](#)
- [Kinesis Video Streams를 위한 작업](#)
- [Kinesis Video Streams용 Amazon 리소스 이름\(ARN\)](#)
- [다른 IAM 계정에 Kinesis 비디오 스트림에 대한 액세스 권한 부여](#)
- [Kinesis Video Streams에 대한 예제 정책](#)

정책 구문

IAM 정책은 하나 이상의 문으로 구성된 JSON 문서입니다. 각 명령문의 구조는 다음과 같습니다.

```
{
```

```

"Statement": [{
  "Effect": "effect",
  "Action": "action",
  "Resource": "arn",
  "Condition": {
    "condition": {
      "key": "value"
    }
  }
}
]
}

```

명령문을 이루는 요소는 다양합니다.

- 효과 — 효과는 Allow 또는 Deny 일 수 있습니다. 기본적으로 사용자에게는 리소스 및 API 작업을 사용할 권한이 없으므로 모든 요청이 거부됩니다. 명시적 허용은 기본 설정을 무시합니다. 명시적 거부는 모든 허용을 무시합니다.
- 조치 — 조치는 권한을 부여하거나 거부하는 특정 API 작업입니다.
- 리소스 — 작업의 영향을 받는 리소스입니다. 명령문에서 리소스를 지정하려면 Amazon 리소스 이름 (ARN)을 사용해야 합니다.
- 조건: 조건은 선택 사항으로서 정책이 적용되는 시점을 제어하는 데 사용할 수 있습니다.

[IAM 정책을 생성하고 관리할 때는 IAM 정책 생성기와 IAM 정책 시뮬레이터를 사용하는 것이 좋습니다.](#)

Kinesis Video Streams를 위한 작업

IAM 정책 설명에는 IAM을 지원하는 모든 서비스의 모든 API 작업을 지정할 수 있습니다. Kinesis Video Streams의 경우 API 작업 이름과 함께 다음 접두사를 사용하십시오.

kinesisvideo: 예를 들어, kinesisvideo:CreateStream, kinesisvideo:ListStreams 및 kinesisvideo:DescribeStream입니다.

문 하나에 여러 작업을 지정하려면 다음과 같이 쉼표로 구분합니다.

```
"Action": ["kinesisvideo:action1", "kinesisvideo:action2"]
```

와일드카드를 사용하여 여러 작업을 지정할 수도 있습니다. 예를 들어 다음과 같이 이름이 "Get"으로 시작되는 모든 작업을 지정할 수 있습니다.


```
"Action": "kinesisvideo:Get*"
```

모든 Kinesis Video Streams 작업을 지정하려면 다음과 같이 별표(*) 와일드카드를 사용합니다.

```
"Action": "kinesisvideo:*"
```

Kinesis Video Streams API 작업의 전체 목록은 [Kinesis Video Streams API 참조](#)를 참조하세요.

Kinesis Video Streams용 Amazon 리소스 이름(ARN)

각 IAM 정책 명령문은 ARN을 사용하여 지정한 리소스에 적용됩니다.

Kinesis Video Streams에는 다음 ARN 리소스 형식을 사용합니다.

```
arn:aws:kinesisvideo:region:account-id:stream/stream-name/code
```

예:

```
"Resource": arn:aws:kinesisvideo:*:111122223333:stream/my-stream/0123456789012
```

를 사용하여 스트림의 ARN을 가져올 수 있습니다. [DescribeStream](#)

다른 IAM 계정에 Kinesis 비디오 스트림에 대한 액세스 권한 부여

Kinesis Video Streams의 스트림에서 작업을 수행하려면 다른 IAM 계정에 권한을 부여해야 할 수 있습니다. 다음의 개요는 여러 계정에 걸쳐 비디오 스트림 액세스 권한을 부여하는 일반적인 단계를 설명한 것입니다.

1. 계정에서 생성된 스트림 리소스에서 작업을 수행할 수 있는 권한을 부여하려는 계정의 12자리 계정 ID를 가져옵니다.

예: 다음 단계에서는 권한을 부여하려는 계정의 계정 ID로 1111111111을 사용하고 Kinesis Video Streams의 ID로 9999999999를 사용합니다.

2. 스트림을 소유한 계정 (9999999999) 에서 부여하려는 액세스 수준을 허용하는 IAM 관리형 정책을 생성하십시오.

샘플 정책:

```
{
```


⚠ Important

역할 ARN을 기록해 두십시오. 다음 단계에서 필요합니다.

4. 다른 계정인 111111111111에서 관리형 정책을 생성하십시오. 그러면 이전 단계에서 999999999999 계정에서 생성한 역할에 대한 AssumeRole 작업을 수행할 수 있습니다. 이전 단계의 역할 ARN을 언급해야 합니다.

샘플 정책:

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "sts:AssumeRole",
    "Resource": "arn:aws:iam::999999999999:role/CustomRoleName"
  }
}
```

5. 이전 단계에서 생성한 정책을 111111111111 계정의 역할 또는 사용자와 같은 IAM 엔티티에 연결합니다. 이제 이 사용자는 999999999999 계정에서 역할을 수입할 수 있는 권한을 갖게 되었습니다. CustomRoleName

이 사용자의 자격 증명은 AWS STS AssumeRole API를 호출하여 세션 자격 증명을 가져오고, 이후 999999999999 계정에서 생성된 스트림에서 Kinesis Video Streams API를 호출하는 데 사용됩니다.

```
aws sts assume-role --role-arn "arn:aws:iam::999999999999:role/CustomRoleName" --
role-session-name "kvs-cross-account-assume-role"
{
  "Credentials": {
    "AccessKeyId": "",
    "SecretAccessKey": "",
    "SessionToken": "",
    "Expiration": ""
  },
  "AssumedRoleUser": {
    "AssumedRoleId": "",
    "Arn": ""
  }
}
```

```
}

```

6. 환경의 이전 설정을 기반으로 액세스 키, 비밀 키 및 세션 자격 증명을 설정합니다.

```
set AWS_ACCESS_KEY_ID=
set AWS_SECRET_ACCESS_KEY=
set AWS_SESSION_TOKEN=

```

7. Kinesis Video Streams API를 실행하여 999999999999 계정의 스트림에 대한 데이터 엔드포인트를 설명하고 가져올 수 있습니다.

```
aws kinesisvideo describe-stream --stream-arn "arn:aws:kinesisvideo:us-west-2:999999999999:stream/custom-stream-name/1613732218179"
{
  "StreamInfo": {
    "StreamName": "custom-stream-name",
    "StreamARN": "arn:aws:kinesisvideo:us-west-2:999999999999:stream/custom-stream-name/1613732218179",
    "KmsKeyId": "arn:aws:kms:us-west-2:999999999999:alias/aws/kinesisvideo",
    "Version": "abcd",
    "Status": "ACTIVE",
    "CreationTime": "2018-02-19T10:56:58.179000+00:00",
    "DataRetentionInHours": 24
  }
}

aws kinesisvideo get-data-endpoint --stream-arn "arn:aws:kinesisvideo:us-west-2:999999999999:stream/custom-stream-name/1613732218179" --api-name "PUT_MEDIA"
{
  "DataEndpoint": "https://s-b12345.kinesisvideo.us-west-2.amazonaws.com"
}

```

[계정 간 액세스 권한 부여에 대한 일반적인 step-by-step 지침은 IAM 역할 사용에 대한 액세스 위임을 참조하십시오. AWS 계정](#)

Kinesis Video Streams에 대한 예제 정책

다음 예제 정책은 Kinesis Video Streams에 대한 사용자 액세스를 제어하는 방법을 보여줍니다.

Example 1: 사용자가 모든 Kinesis 비디오 스트림에서 데이터를 가져오도록 허용

이 정책은 사용자 또는 그룹이 모든 Kinesis 비디오 스트림에서 DescribeStream, GetDataEndpoint, GetMedia, ListStreams, 및 ListTagsForStream 작업을 수행할 수 있도록 허용합니다. 이 정책은 임의의 비디오 스트림에서 데이터를 가져올 수 있는 사용자에게 해당됩니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kinesisvideo:Describe*",
        "kinesisvideo:Get*",
        "kinesisvideo:List*"
      ],
      "Resource": "*"
    }
  ]
}
```

Example 2: 사용자가 Kinesis 비디오 스트림을 생성하고 여기에 데이터를 쓰도록 허용

이 정책을 사용하면 사용자 또는 그룹이 CreateStream 및 PutMedia 작업을 수행할 수 있습니다. 이 정책은 비디오 스트림을 생성하고 데이터를 생성된 스트림에 전송할 수 있는 보안 카메라에 해당됩니다.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kinesisvideo:CreateStream",
        "kinesisvideo:PutMedia"
      ],
      "Resource": "*"
    }
  ]
}
```

Example 3: 사용자에게 모든 Kinesis Video Streams 리소스에 대한 전체 액세스 허용

이 정책을 통해 사용자 또는 그룹은 모든 리소스에서 모든 Kinesis Video Streams 작업을 수행할 수 있습니다. 이 정책은 관리자에 해당됩니다.

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

Example 4: 사용자가 특정 Kinesis 비디오 스트림에 데이터를 쓸 수 있도록 허용

이 정책을 사용하면 사용자 또는 그룹이 특정 비디오 스트림에 데이터를 작성할 수 있습니다. 이 정책은 단일 스트림에 데이터를 전송할 수 있는 디바이스에 해당됩니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "kinesisvideo:PutMedia",
      "Resource": "arn:aws:kinesisvideo:us-west-2:123456789012:stream/your_stream/0123456789012"
    }
  ]
}
```

를 사용하여 Kinesis Video Streams 리소스에 대한 액세스 제어 AWS IoT

이 섹션에서는 장치 (예: 카메라) 가 오디오 및 비디오 데이터를 특정 Kinesis 비디오 스트림 하나에만 전송하도록 설정하는 방법을 설명합니다. AWS IoT 자격 증명 제공자와 AWS Identity and Access Management (IAM) 역할을 사용하여 이 작업을 수행할 수 있습니다.

디바이스는 X.509 인증서를 AWS IoT 사용하여 TLS 상호 인증 프로토콜을 사용하여 연결할 수 있습니다. 기타 AWS 서비스 (예: Kinesis Video Streams) 는 인증서 기반 인증을 지원하지 않지만 서명 버전 4 AWS 형식의 자격 증명을 AWS 사용하여 호출할 수 있습니다. 서명 버전 4 알고리즘에서는 일반적으로 호출자에게 액세스 키 ID와 보안 액세스 키가 있어야 합니다. AWS IoT 에는 내장 X.509 인증서를 고유한 디바이스 ID로 사용하여 AWS 요청 (예: Kinesis Video Streams에 대한 요청) 을 인증할 수 있는 자격 증명 공급자가 있습니다. 이렇게 하면 디바이스에 액세스 키 ID와 보안 액세스 키를 저장할 필요가 없습니다.

자격 증명 공급자는 X.509 인증서를 사용하여 클라이언트 (이 경우 카메라에서 실행 중인 Kinesis Video Streams SDK, 비디오 스트림으로 데이터를 전송하려는 Kinesis Video Streams SDK) 를 인증하고 권한이 제한된 임시 보안 토큰을 발급합니다. 토큰을 사용하여 모든 AWS 요청 (이 경우에는 Kinesis Video Streams에 대한 호출) 에 서명하고 인증할 수 있습니다. 자세한 내용은 서비스에 대한 [다이렉트 콜 승인을](#) 참조하십시오. AWS

Kinesis Video Streams에 대한 카메라의 요청을 인증하려면 IAM 역할을 생성 및 구성하고 자격 증명 제공자가 사용자를 대신하여 역할을 맡을 수 있도록 AWS IoT 적절한 IAM 정책을 역할에 연결해야 합니다.

[에 대한 자세한 내용은 설명서를 참조하십시오. AWS IoT AWS IoT Core IAM 사용에 관한 자세한 내용은 AWS Identity and Access Management \(IAM\)를 참조하십시오.](#)

주제

- [AWS IoT ThingName 스트림 이름으로](#)
- [AWS IoT CertificateId 스트림 이름으로](#)
- [AWS IoT 자격 증명을 사용하여 하드 코딩된 스트림 이름으로 스트리밍할 수 있습니다.](#)

AWS IoT ThingName 스트림 이름으로

주제

- [1단계: AWS IoT 사물 유형 및 AWS IoT 사물 생성](#)
- [2단계: 다음과 같이 위임할 IAM 역할 생성 AWS IoT](#)
- [3단계: X.509 인증서 생성 및 구성](#)
- [4단계: Kinesis 비디오 스트림으로 AWS IoT 자격 증명 테스트](#)
- [5단계: 카메라의 파일 시스템에 AWS IoT 인증서 및 자격 증명을 배포하고 비디오 스트림으로 데이터를 스트리밍합니다.](#)

1단계: AWS IoT 사물 유형 및 AWS IoT 사물 생성

에서 AWS IoT 사물은 특정 장치 또는 논리적 개체의 표현입니다. 이 경우 리소스 수준 AWS IoT 액세스 제어를 구성하려는 Kinesis 비디오 스트림을 나타내는 항목이 있습니다. 사물을 생성하려면 먼저 사물 유형을 AWS IoT 생성해야 합니다. 사물 유형을 사용하여 AWS IoT 동일한 사물 유형과 연관된 모든 사물에 공통적인 설명 및 구성 정보를 저장할 수 있습니다.

1. 다음 예제 명령은 `kvs_example_camera` 사물 유형을 생성합니다.

```
aws --profile default iot create-thing-type --thing-type-name kvs_example_camera >
iot-thing-type.json
```

2. 이 예제 명령은 `kvs_example_camera_stream` `kvs_example_camera` 사물 유형의 사물을 생성합니다.

```
aws --profile default iot create-thing --thing-name kvs_example_camera_stream --
thing-type-name kvs_example_camera > iot-thing.json
```

2단계: 다음과 같이 위임할 IAM 역할 생성 AWS IoT

IAM 역할은 AWS 자격 증명이 수행할 수 있는 작업과 수행할 수 없는 작업을 결정하는 권한 정책이 있는 자격 증명이라는 점에서 사용자와 유사합니다. AWS 역할은 역할이 필요한 사용자라면 누구나 맡을 수 있습니다. 역할을 수입한 사람에게는 해당 역할 세션을 위한 임시 보안 자격 증명이 제공됩니다.

이 단계에서 생성하는 역할은 클라이언트로부터 자격 증명 권한 부여 요청을 수행할 때 보안 토큰 서비스 (STS) 에서 임시 자격 증명을 받는 것으로 간주할 수 있습니다. AWS IoT 이 경우 클라이언트는 카메라에서 실행되는 Kinesis Video Streams SDK입니다.

다음 단계를 수행하여 이 IAM 역할을 생성하고 구성합니다.

1. IAM 역할을 생성합니다.

다음 예제 명령은 `KVSCameraCertificateBasedIAMRole` 라는 IAM 역할을 생성합니다.

```
aws --profile default iam create-role --role-name KVSCameraCertificateBasedIAMRole
--assume-role-policy-document 'file://iam-policy-document.json' > iam-role.json
```

다음 신뢰 정책 JSON을 `iam-policy-document.json`에 사용할 수 있습니다.

```
{
```



```

    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Principal": {
          "Service": "credentials.iot.amazonaws.com"
        },
        "Action": "sts:AssumeRole"
      }
    ]
  }
}

```

- 다음으로, 이전에 생성한 IAM 역할에 권한 정책을 연결합니다. 이 권한 정책은 리소스에 대한 선택적 액세스 제어 (지원되는 작업의 일부) 를 허용합니다. AWS 이 경우 AWS 리소스는 카메라가 데이터를 보내도록 하려는 비디오 스트림입니다. 다시 말해, 모든 구성 단계가 완료되면 이 카메라는 이 비디오 스트림에만 데이터를 전송할 수 있습니다.

```

aws --profile default iam put-role-policy --role-name
KVSCameraCertificateBasedIAMRole --policy-name KVSCameraIAMPolicy --policy-
document 'file://iam-permission-document.json'

```

.json에 대해 다음과 같은 IAM 정책 JSON을 사용할 수 있습니다 iam-permission-document.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kinesisvideo:DescribeStream",
        "kinesisvideo:PutMedia",
        "kinesisvideo:TagStream",
        "kinesisvideo:GetDataEndpoint"
      ],
      "Resource": "arn:aws:kinesisvideo:*:*:stream/${credentials-
iot:ThingName}/*"
    }
  ]
}

```

이 정책은 자리 표시자 (`{credentials-iot:}`) 로 지정된 비디오 스트림 (AWS 리소스) 에서만 지정된 작업을 승인한다는 점에 유의하십시오. ThingName 이 자리 표시자는 AWS IoT 자격 증명 제공자가 요청에서 비디오 스트림 이름을 보낼 ThingName 때 AWS IoT 사물 속성의 값을 사용합니다.

- 다음으로, IAM 역할의 역할 별칭을 생성합니다. 역할 별칭은 IAM 역할을 가리키는 대체 데이터 모델입니다. AWS IoT 자격 증명 공급자 요청에는 STS에서 임시 자격 증명을 받기 위해 맡을 IAM 역할을 지정하는 역할 별칭이 포함되어야 합니다.

다음 샘플 명령은 KvsCameraIoTRoleAlias라는 역할 별칭을 생성합니다.

```
aws --profile default iot create-role-alias --role-alias KvsCameraIoTRoleAlias --role-arn $(jq --raw-output '.Role.Arn' iam-role.json) --credential-duration-seconds 3600 > iot-role-alias.json
```

- 이제 역할 별칭을 사용하여 인증서가 연결되면 해당 역할을 AWS IoT 수입할 수 있는 정책을 생성할 수 있습니다.

다음 샘플 명령은 call에 대한 정책을 생성합니다 AWS IoT . KvsCameraIoTPolicy

```
aws --profile default iot create-policy --policy-name KvsCameraIoTPolicy --policy-document 'file://iot-policy-document.json'
```

다음 명령을 iot-policy-document 사용하여.json 문서 JSON을 만들 수 있습니다.

```
cat > iot-policy-document.json <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:AssumeRoleWithCertificate"
      ],
      "Resource": "$(jq --raw-output '.roleAliasArn' iot-role-alias.json)"
    }
  ]
}
EOF
```

3단계: X.509 인증서 생성 및 구성

장치 (비디오 스트림) 간의 AWS IoT 통신은 X.509 인증서를 사용하여 보호됩니다.

1. 이전에 만든 정책을 첨부해야 하는 인증서를 생성하십시오. AWS IoT

```
aws --profile default iot create-keys-and-certificate --set-as-active --
certificate-pem-outfile certificate.pem --public-key-outfile public.pem.key --
private-key-outfile private.pem.key > certificate
```

2. 이전에 KvsCameraIoTPolicy 만든 정책을 이 인증서에 연결합니다. AWS IoT

```
aws --profile default iot attach-policy --policy-name KvsCameraIoTPolicy --target
$(jq --raw-output '.certificateArn' certificate)
```

3. 방금 생성한 인증서에 AWS IoT thing (kvs_example_camera_stream) 을 첨부하세요.

```
aws --profile default iot attach-thing-principal --thing-name
kvs_example_camera_stream --principal $(jq --raw-output '.certificateArn'
certificate)
```

4. AWS IoT 자격 증명 공급자를 통해 요청을 승인하려면 AWS 계정 ID에 고유한 AWS IoT 자격 증명 엔드포인트가 필요합니다. 다음 명령을 사용하여 AWS IoT 자격 증명 엔드포인트를 가져올 수 있습니다.

```
aws --profile default iot describe-endpoint --endpoint-type iot:CredentialProvider
--output text > iot-credential-provider.txt
```

5. 이전에 만든 X.509 인증서 외에도 TLS를 통해 백엔드 서비스와 신뢰를 구축하려면 CA 인증서도 있어야 합니다. 다음 명령을 사용하여 CA 인증서를 가져올 수 있습니다.

```
curl --silent 'https://www.amazontrust.com/repository/SFSRootCAG2.pem' --output
cacert.pem
```

4단계: Kinesis 비디오 스트림으로 AWS IoT 자격 증명 테스트

이제 지금까지 설정한 AWS IoT 자격 증명을 테스트할 수 있습니다.

1. 먼저, 이 구성을 테스트하려는 Kinesis 비디오 스트림을 생성합니다.

⚠ Important

이전 단계 (kvs_example_camera_stream) 에서 만든 AWS IoT 사물 이름과 동일한 이름으로 비디오 스트림을 생성합니다.

```
aws kinesismedia create-stream --data-retention-in-hours 24 --stream-name
kvs_example_camera_stream
```

2. 그런 다음 AWS IoT 자격 증명 공급자에게 전화하여 임시 자격 증명을 받으십시오.

```
curl --silent -H "x-amzn-iot-thingname:kvs_example_camera_stream" --cert
certificate.pem --key private.pem.key https://IOT_GET_CREDENTIAL_ENDPOINT/role-
aliases/KvsCameraIoTRoleAlias/credentials --cacert ./cacert.pem > token.json
```

ℹ Note

다음 명령을 사용하여 가져올 수 있습니다 IOT_GET_CREDENTIAL_ENDPOINT.

```
IOT_GET_CREDENTIAL_ENDPOINT=`cat iot-credential-provider.txt`
```

출력 JSON에는 Kinesis Video Streams에 액세스하는 데 사용할 수 있는 액세스키, 시크릿키 및 세션토큰이 포함되어 있습니다.

3. 테스트를 위해 이러한 자격 증명을 사용하여 kvs_example_camera_stream 샘플 비디오 DescribeStream 스트림에 대해 Kinesis Video Streams API를 호출할 수 있습니다.

```
AWS_ACCESS_KEY_ID=$(jq --raw-output '.credentials.accessKeyId' token.json)
AWS_SECRET_ACCESS_KEY=$(jq --raw-output '.credentials.secretAccessKey' token.json)
AWS_SESSION_TOKEN=$(jq --raw-output '.credentials.sessionToken' token.json) aws
kinesisvideo describe-stream --stream-name kvs_example_camera_stream
```

5단계: 카메라의 파일 시스템에 AWS IoT 인증서 및 자격 증명을 배포하고 비디오 스트림으로 데이터를 스트리밍합니다.

Note

이 섹션의 단계에서는 를 사용하는 카메라에서 Kinesis 비디오 스트림으로 미디어를 전송하는 방법을 설명합니다. [the section called “C++ 생산자 라이브러리”](#)

1. 이전 단계에서 생성한 X.509 인증서, 개인 키 및 CA 인증서를 카메라의 파일 시스템에 복사합니다. 이러한 파일이 저장되는 경로, 역할 별칭 이름, `gst-launch-1.0` 명령 또는 샘플 애플리케이션을 실행하기 위한 AWS IoT 자격 증명 엔드포인트를 지정합니다.
2. 다음 샘플 명령은 AWS IoT 인증서 인증을 사용하여 Kinesis Video Streams로 비디오를 전송합니다.

```
gst-launch-1.0 rtspsrc location=rtsp://YourCameraRtspUrl short-header=TRUE !
rtph264depay ! video/x-h264,format=avc,alignment=au ! h264parse ! kvssink stream-
name="kvs_example_camera_stream" aws-region="YourAWSRegion" iot-certificate="iot-
certificate,endpoint=credential-account-specific-prefix.credentials.iot.aws-
region.amazonaws.com,cert-path=/path/to/certificate.pem,key-path=/path/to/
private.pem,key,ca-path=/path/to/cacert.pem,role-aliases=KvsCameraIoTRoleAlias"
```

AWS IoT CertificateId 스트림 이름으로

사물을 통해 장치 (예: 카메라) 를 나타내면서 다른 스트림 이름을 승인하려면 AWS IoT `certificateId` 속성을 스트림 이름으로 사용하고 를 사용하여 스트림에 Kinesis Video Streams 권한을 제공할 수 있습니다. AWS IoT AWS IoT이 작업을 수행하는 단계는 몇 가지 변경 사항을 제외하면 앞서 설명한 단계와 비슷합니다.

- 다음과 같이 권한 정책을 IAM 역할 (`iam-permission-document.json`) 에 맞게 수정하십시오.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kinesisvideo:DescribeStream",
```

```

        "kinesisvideo:PutMedia",
        "kinesisvideo:TagStream",
        "kinesisvideo:GetDataEndpoint"
    ],
    "Resource": "arn:aws:kinesisvideo:*:*:stream/${credentials-
iot:AwsCertificateId}/*"
    }
}
}

```

Note

리소스 ARN은 인증서 ID를 스트림 이름에 대한 자리 표시자로 사용합니다. 인증서 ID를 스트림 이름으로 사용하면 IAM 권한이 작동합니다. 다음 설명 스트림 API 호출에서 스트림 이름으로 사용할 수 있도록 인증서에서 인증서 ID를 가져오십시오.

```
export CERTIFICATE_ID=`cat certificate | jq --raw-output '.certificateId'`
```

- Kinesis Video Streams describe-stream CLI 명령을 사용하여 다음 변경 사항을 확인합니다.

```

AWS_ACCESS_KEY_ID=$(jq --raw-output '.credentials.accessKeyId' token.json)
AWS_SECRET_ACCESS_KEY=$(jq --raw-output '.credentials.secretAccessKey' token.json)
AWS_SESSION_TOKEN=$(jq --raw-output '.credentials.sessionToken' token.json) aws
kinesisvideo describe-stream --stream-name ${CERTIFICATE_ID}

```

- Kinesis Video Streams C++ AWS IoT SDK의 [샘플](#) 애플리케이션에 있는 자격 증명 공급자에게 인증서 ID를 전달합니다.

```

credential_provider =
make_unique<IotCertCredentialProvider>(iot_get_credential_endpoint,
    cert_path,
    private_key_path,
    role_alias,
    ca_cert_path,
    certificateId);

```

Note

자격 증명 공급자에게 thingname을 전달한다는 점에 유의하십시오. AWS IoT getenv를 사용하여 다른 속성을 전달하는 것과 마찬가지로 thingname을 데모 애플리케이션에 전달할 수

있습니다. AWS IoT 샘플 애플리케이션을 실행할 때 명령줄 파라미터의 스트림 이름으로 인증서 ID를 사용합니다.

AWS IoT 자격 증명을 사용하여 하드 코딩된 스트림 이름으로 스트리밍할 수 있습니다.

사물을 통해 디바이스 (예: 카메라) 를 나타내면서 특정 Amazon Kinesis 비디오 스트림으로의 스트리밍을 승인하려면 를 사용하여 스트림에 대한 Amazon Kinesis Video Streams 권한을 제공하십시오. AWS IoT 프로세스는 이전 섹션과 비슷하지만 몇 가지 변경 사항이 있습니다.

다음과 같이 권한 정책을 IAM 역할 (iam-permission-document.json) 에 맞게 수정하십시오.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kinesisvideo:DescribeStream",
        "kinesisvideo:PutMedia",
        "kinesisvideo:TagStream",
        "kinesisvideo:GetDataEndpoint"
      ],
      "Resource": "arn:aws:kinesisvideo:*:*:stream/YourStreamName/*"
    }
  ]
}
```

이전 단계에서 생성한 X.509 인증서, 프라이빗 키, CA 인증서를 카메라의 파일 시스템에 복사합니다.

이러한 파일이 저장되는 경로, 역할 별칭 이름, AWS IoT 사물 이름, gst-launch-1.0 명령 또는 샘플 애플리케이션을 실행하기 위한 AWS IoT 자격 증명 엔드포인트를 지정합니다.

다음 샘플 명령은 AWS IoT 인증서 인증을 사용하여 Amazon Kinesis Video Streams로 비디오를 전송합니다.

```
gst-launch-1.0 rtspsrc location=rtsp://YourCameraRtspUrl short-header=TRUE !
rtph264depay ! video/x-h264,format=avc,alignment=au ! h264parse ! kvssink
stream-name="YourStreamName" aws-region="YourAWSRegion" iot-certificate="iot-
```

```
certificate,endpoint=credential-account-specific-prefix.credentials.iot.aws-region.amazonaws.com,cert-path=/path/to/certificate.pem,key-path=/path/to/private.pem,key,ca-path=/path/to/cacert.pem,role-aliases=KvsCameraIoTRoleAlias,iot-thing-name=YourThingName"
```

아마존 키네시스 비디오 스트림 모니터링

Kinesis Video Streams는 전송 스트림에 대한 모니터링 기능을 제공합니다. 자세한 정보는 [모니터링](#)을 참조하세요.

Amazon Kinesis Video Streams에 대한 규정 준수 검증

특정 규정 준수 프로그램의 범위 내에 AWS 서비스 있는지 알아보려면 AWS 서비스 규정 준수 [프로그램의 AWS 서비스 범위별, 규정](#) 참조하여 관심 있는 규정 준수 프로그램을 선택하십시오. 일반 정보는 [AWS 규정 준수 프로그램 AWS 보증 프로그램 규정 AWS](#) 참조하십시오.

를 사용하여 AWS Artifact 타사 감사 보고서를 다운로드할 수 있습니다. 자세한 내용은 의 보고서 <https://docs.aws.amazon.com/artifact/latest/ug/downloading-documents.html> 참조하십시오 AWS Artifact.

사용 시 규정 준수 AWS 서비스 책임은 데이터의 민감도, 회사의 규정 준수 목표, 관련 법률 및 규정에 따라 결정됩니다. AWS 규정 준수에 도움이 되는 다음 리소스를 제공합니다.

- [보안 및 규정 준수 킷스타트 가이드](#) - 이 배포 가이드에서는 아키텍처 고려 사항을 설명하고 보안 및 규정 준수에 AWS 중점을 둔 기본 환경을 배포하기 위한 단계를 제공합니다.
- [Amazon Web Services의 HIPAA 보안 및 규정 준수를 위한 설계 — 이 백서에서는 기업이 HIPAA 적격 애플리케이션을 만드는 AWS 데 사용할 수 있는 방법을 설명합니다.](#)

Note

모든 AWS 서비스 사람이 HIPAA 자격을 갖춘 것은 아닙니다. 자세한 내용은 [HIPAA 적격 서비스 참조](#)를 참조하십시오.

- [AWS 규정 준수 리소스 AWS](#) — 이 워크북 및 가이드 모음은 해당 산업 및 지역에 적용될 수 있습니다.
- [AWS 고객 규정 준수 가이드](#) — 규정 준수의 관점에서 공동 책임 모델을 이해하십시오. 이 가이드에서는 보안을 유지하기 위한 모범 사례를 AWS 서비스 요약하고 여러 프레임워크 (미국 표준 기술 연

구소 (NIST), 결제 카드 산업 보안 표준 위원회 (PCI), 국제 표준화기구 (ISO) 등) 에서 보안 제어에 대한 지침을 매핑합니다.

- [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 내부 AWS 보안 상태를 포괄적으로 파악할 수 있습니다. 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 참조](#)를 참조하십시오.
- [Amazon GuardDuty](#) — 환경에 의심스럽고 악의적인 활동이 있는지 AWS 계정모니터링하여 워크로드, 컨테이너 및 데이터에 대한 잠재적 위협을 AWS 서비스 탐지합니다. GuardDuty 특정 규정 준수 프레임워크에서 요구하는 침입 탐지 요구 사항을 충족하여 PCI DSS와 같은 다양한 규정 준수 요구 사항을 해결하는 데 도움이 될 수 있습니다.
- [AWS Audit Manager](#) — 이를 AWS 서비스 통해 AWS 사용량을 지속적으로 감사하여 위협을 관리하고 규정 및 업계 표준을 준수하는 방법을 단순화할 수 있습니다.

Amazon Kinesis Video Streams의 레질리언스

AWS 글로벌 인프라는 지역 및 가용 AWS 영역을 중심으로 구축됩니다. AWS 지역은 물리적으로 분리되고 격리된 여러 가용 영역을 제공하며, 이러한 가용 영역은 지연 시간이 짧고 처리량이 높으며 중복성이 높은 네트워크로 연결됩니다. 가용 영역을 사용하면 중단 없이 가용 영역 간에 자동으로 장애 조치가 이루어지는 애플리케이션 및 데이터베이스를 설계하고 운영할 수 있습니다. 가용 영역은 기존의 단일 또는 복수 데이터 센터 인프라보다 가용성, 내결함성, 확장성이 뛰어납니다.

AWS [지역 및 가용 영역에 대한 자세한 내용은 글로벌 인프라를 참조하십시오](#).

Kinesis Video Streams의 인프라 보안

관리형 서비스인 Amazon Kinesis Video Streams는 [Amazon Web Services: 보안 프로세스 개요 백서에 설명된 글로벌 네트워크 보안 절차에 따라](#) 보호됩니다. AWS

AWS 게시된 API 호출을 사용하여 네트워크를 통해 Kinesis Video Streams에 액세스할 수 있습니다. 클라이언트가 전송 계층 보안(TLS) 1.2 이상을 지원해야 합니다. 클라이언트는 Ephemeral Diffie-Hellman(DHE) 또는 Elliptic Curve Ephemeral Diffie-Hellman(ECDHE)과 같은 완전 전송 보안(PFS)이 포함된 암호 제품군도 지원해야 합니다. Java 7 이상의 최신 시스템은 대부분 이러한 모드를 지원합니다.

또한 IAM 보안 주체와 연결된 액세스 키 ID 및 비밀 액세스 키를 사용하여 요청에 서명해야 합니다. 또는 [AWS Security Token Service](#)(AWS STS)를 사용하여 임시 보안 인증을 생성하여 요청에 서명할 수 있습니다.

Kinesis Video Streams의 보안 모범 사례

Amazon Kinesis Video Streams는 자체 보안 정책을 개발하고 구현할 때 고려할 수 있는 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.

원격 디바이스의 보안 모범 사례를 보려면 [디바이스 에이전트의 보안 모범 사례](#)를 참조하세요.

최소 권한 액세스 구현

권한을 부여할 때 누가 어떤 권한을 어떤 Kinesis Video Streams 리소스에 부여할지 결정합니다. 해당 리소스에서 허용할 작업을 사용 설정합니다. 따라서 작업을 수행하는 데 필요한 권한만 부여해야 합니다. 최소 권한 액세스를 구현하는 것이 오류 또는 악의적인 의도로 인해 발생할 수 있는 보안 위협과 영향을 최소화할 수 있는 근본적인 방법입니다.

예를 들어, Kinesis Video Streams에 데이터를 전송하는 생산자에게는 PutMedia, GetStreamingEndpoint 및 DescribeStream만 필요합니다. 생산자 애플리케이션에 모든 작업(*) 또는 GetMedia 등의 다른 작업에 대한 사용 권한을 부여하지 마세요.

자세한 내용은 [What Is Least Privilege & Why Do You Need It?](#)을 참조하세요.

IAM 역할 사용

생산자 및 클라이언트 애플리케이션이 Kinesis Video Streams에 액세스하려면 유효한 자격 증명이 있어야 합니다. AWS 보안 인증을 클라이언트 애플리케이션이나 Amazon S3 버킷에 직접 저장하면 안 됩니다. 이러한 자격 증명은 자동으로 교체되지 않는 장기 자격 증명으로, 침해될 경우 비즈니스에 심각한 영향을 미칠 수 있습니다.

대신 IAM 역할을 사용하여 생산자 및 클라이언트 애플리케이션이 Kinesis Video Streams에 액세스할 수 있도록 하기 위한 임시 자격 증명을 관리해야 합니다. 역할을 사용하면 다른 리소스에 액세스하기 위해 장기 자격 증명 (예: 사용자 이름, 암호 또는 액세스 키) 을 사용하지 않아도 됩니다.

자세한 설명은 IAM 사용자 가이드에서 다음 주제를 참조하십시오:

- [IAM 역할](#)

- [역할에 대한 일반적인 시나리오: 사용자, 애플리케이션 및 서비스](#)

API 호출을 모니터링하는 CloudTrail 데 사용합니다.

Kinesis Video Streams는 Kinesis Video Streams에서 사용자, 역할 또는 사용자가 수행한 작업의 기록을 제공하는 AWS 서비스 서비스인 AWS CloudTrail함께 작동합니다.

에서 수집한 CloudTrail 정보를 사용하여 Kinesis Video Streams에 대한 요청, 요청이 이루어진 IP 주소, 요청한 사람, 요청 시기 및 추가 세부 정보를 확인할 수 있습니다.

자세한 정보는 [the section called “을 사용하여 CloudTrail API 호출 로깅”](#)을 참조하세요.

Kinesis 비디오 스트림 프로듀서 라이브러리

Amazon Kinesis 비디오 스트림 프로듀서 라이브러리는 Kinesis 비디오 스트림 프로듀서 SDK의 라이브러리 세트입니다. 클라이언트는 라이브러리와 SDK를 사용하여 Kinesis Video Streams에 안전하게 연결하고 미디어 데이터를 스트리밍하여 콘솔 또는 클라이언트 애플리케이션에서 실시간으로 볼 수 있는 온디바이스 애플리케이션을 구축합니다.

다음과 같은 방식으로 미디어 데이터를 스트리밍할 수 있습니다.

- 실시간으로
- 몇 초 동안 버퍼링한 후
- 미디어 업로드 후

Kinesis Video Streams 스트림을 생성한 후 데이터 전송을 시작할 수 있습니다. SDK를 사용하여 미디어 소스에서 프레임이라는 비디오 데이터를 추출하여 Kinesis Video Streams에 업로드하는 애플리케이션 코드를 생성할 수 있습니다. 이러한 애플리케이션을 생산자 애플리케이션이라고도 합니다.

생산자 라이브러리에는 다음의 구성 요소가 포함됩니다.

- [Kinesis 비디오 스트림 프로듀서 클라이언트](#)
- [Kinesis 비디오 스트림 프로듀서 라이브러리](#)

Kinesis 비디오 스트림 프로듀서 클라이언트

Kinesis 비디오 스트림 프로듀서 클라이언트에는 다음이 포함됩니다. `KinesisVideoClient` 수업. 이 클래스는 미디어 소스를 관리하고, 소스에서 데이터를 수신하며, 미디어 소스에서 Kinesis Video Streams로 데이터가 흐를 때 스트림 라이프사이클을 관리합니다. 또한 다음을 제공합니다. `MediaSourceKinesis Video Streams`와 전용 하드웨어 및 소프트웨어 간의 상호 작용을 정의하기 위한 인터페이스입니다.

미디어 원본은 거의 모든 것이 될 수 있습니다. 예를 들어, 카메라 미디어 원본 또는 마이크 미디어 원본을 사용할 수 있습니다. 미디어 원본은 오디오 및 비디오 원본만으로 국한되지 않습니다. 예를 들어, 데이터 로그는 텍스트 파일이어야 하지만, 데이터 스트림으로도 전송할 수 있습니다. 데이터를 동시에 스트리밍하는 여러 대의 카메라를 스마트폰에 탑재할 수도 있습니다.

임의의 원본으로부터 데이터를 얻기 위해서는 MediaSource 인터페이스를 구현해야 합니다. 이 인터페이스는 기본적으로는 지원하지 않는 추가적인 시나리오도 구현할 수 있습니다. 예를 들어 다음을 Kinesis Video Streams로 전송하도록 선택할 수 있습니다.

- 진단 데이터 스트림(예: 애플리케이션 로그 및 이벤트)
- 적외선 카메라, RADAR 또는 깊이 카메라의 데이터

Kinesis Video Streams는 카메라와 같은 미디어 제작 장치를 위한 내장 구현을 제공하지 않습니다. 이러한 디바이스로부터 데이터를 추출하려면 코드를 구현해야 하기 때문에 자체적인 미디어 원본 구현을 생성해야 합니다. 그런 다음 사용자 지정 미디어 소스를 명시적으로 등록할 수 있습니다. KinesisVideoClient 데이터를 Kinesis 비디오 스트림에 업로드합니다.

Kinesis 비디오 스트림 프로듀서 클라이언트는 자바 및 Android 애플리케이션에서 사용할 수 있습니다. 자세한 내용은 [Java 생산자 라이브러리 사용하기](#) 및 [Android 생산자 라이브러리 사용](#) 단원을 참조하세요.

Kinesis 비디오 스트림 프로듀서 라이브러리

Kinesis 비디오 스트림 프로듀서 라이브러리는 Kinesis 비디오 스트림 프로듀서 클라이언트에 포함되어 있습니다. Kinesis Video Streams와의 긴밀한 통합을 원하는 사용자가 라이브러리를 직접 사용할 수도 있습니다. 전용 운영 체제, 네트워크 스택 또는 제한적 내장 리소스를 갖춘 디바이스와의 통합도 가능합니다.

Kinesis 비디오 스트림 프로듀서 라이브러리는 Kinesis Video Streams로 스트리밍하기 위한 상태를 구현합니다. 사용자 자체의 전송 구현을 요구하고 서비스를 왕래하는 각 메시지를 명시적으로 처리하는 콜백 후크를 제공합니다.

다음과 같은 이유로 Kinesis Video Streams 프로듀서 라이브러리를 직접 사용할 수 있습니다.

- 애플리케이션을 실행하고자 하는 디바이스에 Java 가상 머신이 없는 경우.
- Java 이외의 언어로 애플리케이션 코드를 작성하고자 하는 경우.
- 메모리 및 처리 능력과 같은 제한으로 인해 코드의 오버헤드 양을 줄이고 최소한의 추상화 수준으로 제한해야 합니다.

현재 Kinesis 비디오 스트림 프로듀서 라이브러리는 Android, C, C++ 및 Java 애플리케이션에서 사용할 수 있습니다. 자세한 내용은 다음에서 지원되는 언어를 참조하십시오. 관련 주제.

관련 주제

[Java 생산자 라이브러리 사용하기](#)

[Android 생산자 라이브러리 사용](#)

[C++ 생산자 라이브러리 사용하기](#)

[C 생산자 라이브러리 사용](#)

[Raspberry Pi에서 C++ 생산자 SDK 사용](#)

Java 생산자 라이브러리 사용하기

Amazon Kinesis Video Streams에서 제공하는 Java 프로듀서 라이브러리를 사용하여 최소한의 구성으로 애플리케이션 코드를 작성하고, 디바이스에서 Kinesis 비디오 스트림으로 미디어 데이터를 전송할 수 있습니다.

애플리케이션에서 Kinesis 비디오 스트림으로 데이터 스트리밍을 시작할 수 있도록 코드를 Kinesis Video Streams와 통합하려면 다음 단계를 수행하십시오.

1. `KinesisVideoClient` 객체의 인스턴스를 생성합니다.
2. 미디어 원본 정보를 제공하여 `MediaSource` 객체를 생성합니다. 예를 들어 카메라 미디어 원본을 생성할 때 카메라를 식별하고 카메라가 사용하는 인코딩을 지정하는 등 정보를 제공합니다.

스트리밍을 시작하고자 할 때 사용자 지정 미디어 원본을 생성해야 합니다.

3. 미디어 원본을 `KinesisVideoClient`에 등록합니다.

`KinesisVideoClient`에 미디어 원본을 등록한 후에 해당 미디어 원본으로 데이터를 사용할 수 있을 때 해당 데이터로 `KinesisVideoClient`를 호출합니다.

절차: Java 생산자 SDK 사용

이 절차는 Java 애플리케이션에서 Kinesis Video Streams Java 프로듀서 클라이언트를 사용하여 Kinesis 비디오 스트림으로 데이터를 보내는 방법을 보여줍니다.

이 단계에서는 카메라 또는 마이크와 같은 미디어 소스를 필요로 하지 않습니다. 대신 테스트 목적으로 코드가 일련의 바이트로 구성된 샘플 프레임을 생성합니다. 카메라 및 마이크와 같은 실제 소스로부터 미디어 데이터를 전송할 때 동일한 코딩 패턴을 사용할 수 있습니다.

이 절차에는 다음 단계가 포함됩니다.

- [코드 다운로드 및 구성](#)
- [코드 쓰기 및 검사](#)
- [코드 실행 및 확인](#)

사전 조건

- 샘플 코드에서는 자격 증명 프로필 파일에 설정한 프로필을 지정하여 AWS 자격 증명을 제공합니다. 아직 설정하지 않았다면 먼저 자격 증명 프로필을 설정합니다. 자세한 내용은 [의 개발용 AWS 자격 증명 및 지역 설정을](#) 참조하십시오 AWS SDK for Java.

Note

Java 예제에서는 SystemPropertiesCredentialsProvider 객체를 사용하여 자격 증명을 얻습니다. 제공자가 aws.accessKeyId 및 aws.secretKey Java 시스템 속성으로부터 이들 자격 증명을 검색합니다. Java 개발 환경에서 이 시스템 속성을 설정합니다. Java 시스템 속성 설정 방법에 관한 정보는 특정 통합 개발 환경(IDE)에 관한 문서를 참조하십시오.

- 파일을 NativeLibraryPath 포함해야 합니다. KinesisVideoProducerJNI 파일은 <https://github.com/aws-labs/amazon-kinesis-video-streams-producer-sdk-cpp>에서 확인할 수 producer-sdk-cpp 있습니다. 이 파일의 파일 이름 확장자는 운영 체제에 따라 다릅니다.
 - KinesisVideoProducer리눅스용 JNI.so
 - KinesisVideoProducermacOS용 jni.dylib
 - KinesisVideoProducer윈도우용 JNI.dll

Note

macOS, 우분투, 윈도우, 라즈비안용으로 사전 빌드된 라이브러리는 <https://github.com/aws-labs/amazon-kinesis-video-streams-producer-sdk-java>에서 이용할 수 있습니다src/main/resources/lib. 다른 환경에서는 [C++ 생산자 라이브러리](#)를 컴파일합니다.

1단계: Java 프로듀서 라이브러리 코드 다운로드 및 구성

Java 생산자 라이브러리 절차의 이 단원에서는 Java 예제 코드를 다운로드하고, 프로젝트를 Java IDE에 가져온 다음, 라이브러리 위치를 구성합니다.

사전 요건과 이 예제에 관한 기타 세부 정보는 [Java 생산자 라이브러리 사용](#)을 참조하십시오.

1. 디렉터리를 생성한 다음 GitHub 리포지토리에서 예제 소스 코드를 복제합니다.

```
git clone https://github.com/aws-labs/amazon-kinesis-video-streams-producer-sdk-java
```

2. 사용하는 자바 통합 개발 환경 (IDE) (예: [Eclipse](#) 또는 [JetBrains IntelliJ IDEA](#)) 을 열고 다운로드한 Apache Maven 프로젝트를 가져옵니다.

- IntelliJ IDEA에서 [Import]를 선택합니다. 다운로드한 패키지의 루트에 있는 pom.xml 파일을 찾습니다.
- Eclipse: [File], [Import], [Maven], [Existing Maven Projects]를 차례로 선택합니다. 그런 다음 kinesis-video-java-demo 디렉터리로 이동합니다.

자세한 내용은 IDE 관련 문서를 참조하십시오.

3. Java 예제 코드는 현재 자격 증명을 사용합니다. AWS 다른 자격 증명 프로필을 사용하려면 DemoAppMain.java에 있는 다음 코드를 찾습니다.

```
final KinesisVideoClient kinesisVideoClient = KinesisVideoJavaClientFactory
    .createKinesisVideoClient(
        Regions.US_WEST_2,
        AuthHelper.getSystemPropertiesCredentialsProvider());
```

코드를 다음으로 변경합니다.

```
final KinesisVideoClient kinesisVideoClient = KinesisVideoJavaClientFactory
    .createKinesisVideoClient(
        Regions.US_WEST_2,
        new ProfileCredentialsProvider("credentials-profile-name"));
```

자세한 내용은 [ProfileCredentialsProviderAWS SDK for Java](#)참조를 참조하십시오.

다음 단계

[the section called “2단계: 코드 작성 및 검토”](#)

2단계: 코드 작성 및 검토

[Java Producer Library 프로시저](#)의 이 섹션에서는 이전 섹션에서 다운로드한 Java 예제 코드를 작성하고 검토합니다.

Java 테스트 애플리케이션([DemoAppMain](#))은 다음과 같은 코딩 패턴을 보입니다.

- KinesisVideoClient의 인스턴스를 만듭니다.
- MediaSource의 인스턴스를 만듭니다.
- MediaSource를 클라이언트에 등록합니다.
- 스트리밍을 시작합니다. MediaSource를 시작하면 데이터를 클라이언트로 보내기 시작합니다.

다음 단원들에서 세부 정보가 제공됩니다.

인스턴스 만들기 KinesisVideoClient

createKinesisVideoClient 작업을 호출하여 KinesisVideoClient 객체를 생성합니다.

```
final KinesisVideoClient kinesisVideoClient = KinesisVideoJavaClientFactory
    .createKinesisVideoClient(
        Regions.US_WEST_2,
        AuthHelper.getSystemPropertiesCredentialsProvider());
```

KinesisVideoClient가 네트워크 호출을 하려면 인증을 위한 자격 증명이 필요합니다.

SystemPropertiesCredentialsProvider 인스턴스를 전달하면, 인스턴스가 자격 증명 파일에 있는 기본 프로필의 AWSCredentials을 읽습니다.

```
[default]
aws_access_key_id = ABCDEFGHIJKLMOPQRSTU
aws_secret_access_key = AbCd1234EfGh5678IjKl9012MnOp3456QrSt7890
```

인스턴스 만들기 MediaSource

Kinesis 비디오 스트림으로 바이트를 전송하려면 데이터를 생성해야 합니다. Amazon Kinesis Video Streams는 데이터 소스를 나타내는 인터페이스를 제공합니다MediaSource.

예를 들어 Kinesis Video Streams Java 라이브러리는 ImageFileMediaSource 인터페이스 구현을 제공합니다. 이 클래스는 Kinesis 비디오 스트림이 아닌 일련의 미디어 파일에서만 데이터를 읽지만 코드를 테스트하는 데 사용할 수 있습니다.

```
final MediaSource bytesMediaSource = createImageFileMediaSource();
```

MediaSource 클라이언트에 등록

클라이언트에 관해 알도록 생성한 미디어 원본을 KinesisVideoClient에 등록합니다(그런 다음 데이터를 클라이언트에 전송할 수 있습니다).

```
kinesisVideoClient.registerMediaSource(mediaSource);
```

미디어 소스 시작

데이터 생성을 시작하고 클라이언트에 전송할 수 있도록 미디어 소스를 시작합니다.

```
bytesMediaSource.start();
```

다음 단계

[the section called “3단계: 코드 실행 및 확인”](#)

3단계: 코드 실행 및 확인

Java [프로듀서 라이브러리용 Java](#) 테스트 하네스를 실행하려면 다음과 같이 하십시오.

1. 선택하세요 DemoAppMain.
2. 실행, DemoAppMain'실행을 선택합니다.
3. 애플리케이션용 JVM 인수에 자격 증명을 추가합니다.
 - 임시 AWS 자격 증명인 경우: "-Daws.accessKeyId={YourAwsAccessKey} -Daws.secretKey={YourAwsSecretKey} -Djava.library.path={NativeLibraryPath}"
 - 임시 AWS 자격 증명의 경우: "-Daws.accessKeyId={YourAwsAccessKey} -Daws.secretKey={YourAwsSecretKey} -Daws.sessionToken={YourAwsSessionToken} -Djava.library.path={NativeLibraryPath}"

4. [에 로그인하여 Kinesis Video Streams 콘솔을 엽니다.](#) AWS Management Console

[Manage streams] 페이지에서 스트림을 선택합니다.

5. 내장 플레이어에서 샘플 비디오가 재생됩니다. 프레임이 쌓여 비디오가 재생될 때까지 잠깐 기다려야 할 수 있습니다(일반적인 대역폭과 프로세서 상태에서 최대 10초).

코드 예제는 스트림을 생성합니다. 코드에 있는 MediaSource가 시작되면 샘플 프레임을 KinesisVideoClient로 전송하기 시작합니다. 그러면 클라이언트가 Kinesis 비디오 스트림으로 데이터를 전송합니다.

Android 생산자 라이브러리 사용

Amazon Kinesis Video Streams에서 제공하는 Android 프로듀서 라이브러리를 사용하여 Android 디바이스에서 Kinesis 비디오 스트림으로 미디어 데이터를 전송하는 애플리케이션 코드를 최소한의 구성으로 작성할 수 있습니다.

애플리케이션에서 Kinesis 비디오 스트림으로 데이터 스트리밍을 시작할 수 있도록 코드를 Kinesis Video Streams와 통합하려면 다음 단계를 수행하십시오.

1. KinesisVideoClient 객체의 인스턴스를 생성합니다.
2. 미디어 원본 정보를 제공하여 MediaSource 객체를 생성합니다. 예를 들어 카메라 미디어 원본을 생성할 때 카메라를 식별하고 카메라가 사용하는 인코딩을 지정하는 등 정보를 제공합니다.

스트리밍을 시작하고자 할 때 사용자 지정 미디어 원본을 생성해야 합니다.

절차: Android 생산자 SDK 사용

이 절차는 안드로이드 애플리케이션에서 Kinesis Video Streams Android 프로듀서 클라이언트를 사용하여 Kinesis 비디오 스트림으로 데이터를 보내는 방법을 보여줍니다.

이 절차에는 다음 단계가 포함됩니다.

- [the section called “사전 조건”](#)
- [the section called “1 단계: 코드 다운로드 및 구성”](#)
- [the section called “2단계: 코드 검사”](#)
- [the section called “3단계: 코드 실행 및 확인”](#)

사전 조건

- 애플리케이션 코드 검사, 편집, 실행에는 [Android Studio](#)를 권장합니다. 안정적인 최신 버전을 사용하는 것이 좋습니다.
- 샘플 코드에서는 Amazon Cognito 자격 증명을 제공합니다.

다음 절차에 따라 Amazon Cognito 사용자 풀과 자격 증명 풀을 설정하십시오.

- [사용자 풀 설정](#)
- [자격 증명 풀 설정](#)

사용자 풀 설정

사용자 풀을 설정하려면

1. [Amazon Cognito 콘솔에](#) 로그인하여 지역이 올바른지 확인합니다.
2. 왼쪽 탐색에서 사용자 풀을 선택합니다.
3. 사용자 풀 섹션에서 사용자 풀 생성을 선택합니다.
4. 다음 섹션을 완료하십시오.
 - a. 1단계: 로그인 환경 구성 - Cognito 사용자 풀 로그인 옵션 섹션에서 적절한 옵션을 선택합니다.

다음을 선택합니다.
 - b. 2단계: 보안 요구 사항 구성 - 적절한 옵션을 선택합니다.

다음을 선택합니다.
 - c. 3단계: 가입 환경 구성 - 적절한 옵션을 선택합니다.

다음을 선택합니다.
 - d. 4단계: 메시지 전송 구성 - 적절한 옵션을 선택합니다.

IAM 역할 선택 필드에서 기존 역할을 선택하거나 새 역할을 생성합니다.

다음을 선택합니다.
 - e. 5단계: 앱 통합 - 적절한 옵션을 선택합니다.

초기 앱 클라이언트 필드에서 기밀 클라이언트를 선택합니다.

다음을 선택합니다.

- f. 6단계: 검토 및 생성 - 이전 섹션에서 선택한 내용을 검토한 다음 사용자 풀 생성을 선택합니다.
5. 사용자 풀 페이지에서 방금 생성한 풀을 선택합니다.

사용자 풀 ID를 복사하고 나중에 사용할 수 있도록 기록해 두십시오.

awsconfiguration.json파일에는 다음과 같습니다CognitoUserPool.Default.PoolId.

6. 앱 통합 탭을 선택하고 페이지 하단으로 이동합니다.
7. 앱 클라이언트 목록 섹션에서 방금 생성한 앱 클라이언트 이름을 선택합니다.

클라이언트 ID를 복사하고 나중에 사용할 수 있도록 기록해 둡니다.

awsconfiguration.json파일에는 다음과 같습니다

다CognitoUserPool.Default.AppClientId.

8. 클라이언트 암호를 보여주고 나중에 사용할 수 있도록 기록해 두십시오. awsconfiguration.json파일에는 다음과 같습니다CognitoUserPool.Default.AppClientSecret.

자격 증명 풀 설정

자격 증명 풀을 설정하려면

1. [Amazon Cognito 콘솔에](#) 로그인하여 지역이 올바른지 확인합니다.
2. 왼쪽 탐색에서 자격 증명 풀을 선택합니다.
3. 자격 증명 풀 생성을 선택합니다.
4. 자격 증명 풀을 구성합니다.

a. 1단계: ID 풀 신뢰 구성 - 다음 섹션을 완료하십시오.

- 사용자 액세스 - 인증된 액세스를 선택합니다.
- 인증된 자격 증명 소스 — Amazon Cognito 사용자 풀 선택

다음을 선택합니다.

b. 2단계: 권한 구성 - 인증된 역할 섹션에서 다음 필드를 작성합니다.

- IAM 역할 - 새 IAM 역할 생성을 선택합니다.

- IAM 역할 이름 - 이름을 입력하고 이후 단계를 위해 기록해 둡니다.

다음을 선택합니다.

- c. 3단계: ID 공급자 연결 - 사용자 풀 세부 정보 섹션에서 다음 필드를 작성합니다.

- 사용자 풀 ID - 이전에 생성한 사용자 풀을 선택합니다.
- 앱 클라이언트 ID - 이전에 생성한 앱 클라이언트 ID를 선택합니다.

다음을 선택합니다.

- d. 4단계: 속성 구성 - ID 풀 이름 필드에 이름을 입력합니다.

다음을 선택합니다.


- e. 5단계: 검토 및 생성 - 각 섹션에서 선택 내용을 검토한 다음 ID 풀 생성을 선택합니다.

5. 자격 증명 풀 페이지에서 새 자격 증명 풀을 선택합니다.

자격 증명 풀 ID를 복사하고 나중에 사용할 수 있도록 기록해 두십시오. `awsconfiguration.json` 파일에는 다음과 같습니다 `CredentialsProvider.CognitoIdentity.Default.PoolId`.

6. IAM 역할에 대한 권한을 업데이트하십시오.

- a. <https://console.aws.amazon.com/iam/> 에서 AWS Management Console 로그인하고 IAM 콘솔을 엽니다.
- b. 왼쪽 탐색에서 [Roles] 를 선택합니다.
- c. 위에서 생성한 역할을 찾아 선택합니다.

 Note

필요한 경우 검색 창을 사용하세요.

- d. 첨부된 권한 정책을 선택합니다.

[편집(Edit)]을 선택합니다.

- e. JSON 탭을 선택하고 정책을 다음과 같이 대체합니다.

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

```

    "Statement": [
      {
        "Effect": "Allow",
        "Action": [
          "cognito-identity:*",
          "kinesisvideo:*"
        ],
        "Resource": [
          "*"
        ]
      }
    ]
  }
}

```

다음을 선택합니다.

- f. 새 버전을 기본값으로 설정 (아직 선택하지 않은 경우) 옆의 상자를 선택합니다.

변경 사항 저장(Save changes)을 선택합니다.

1단계: Android 프로듀서 라이브러리 코드 다운로드 및 구성

Android 생산자 라이브러리 절차의 이 단원에서는 Android 예제 코드를 다운로드하고 Android Studio 에서 프로젝트를 엽니다.

사전 요건과 이 예제에 관한 기타 세부 정보는 [Android 생산자 라이브러리 사용](#)을 참조하십시오.

1. 디렉터리를 생성한 다음 AWS Mobile SDK for Android GitHub 리포지토리에서 해당 디렉터리를 복제합니다.

```
git clone https://github.com/aws-labs/aws-sdk-android-samples
```

2. [Android Studio](#)를 엽니다.
3. 열리는 화면에서 [Open an existing Android Studio project]를 선택합니다.
4. aws-sdk-android-samples/AmazonKinesisVideoDemoApp 디렉터리로 이동하여 [OK]를 선택합니다.
5. AmazonKinesisVideoDemoApp/src/main/res/raw/awsconfiguration.json 파일을 엽니다.

CredentialsProvider노드에서 [사전 요구 사항](#) 섹션의 자격 증명 풀 설정 절차의 자격 증명 풀 ID를 제공하고 사용자 ID AWS 리전 (예:) 를 입력합니다. **us-west-2**

CognitoUserPool노드에서 [사전 요구 사항](#) 섹션의 사용자 풀 설정 절차에 나와 있는 앱 클라이언트 암호, 앱 클라이언트 ID 및 풀 ID를 제공하고 사용자 AWS 리전 (예:) 를 입력합니다. **us-west-2**

6. awsconfiguration.json 파일은 다음과 비슷합니다.

```
{
  "Version": "1.0",
  "CredentialsProvider": {
    "CognitoIdentity": {
      "Default": {
        "PoolId": "us-west-2:01234567-89ab-cdef-0123-456789abcdef",
        "Region": "us-west-2"
      }
    }
  },
  "IdentityManager": {
    "Default": {}
  },
  "CognitoUserPool": {
    "Default": {
      "AppClientSecret": "abcdefghijklmnopqrstuvwxy0123456789abcdefghijklmnop",
      "AppClientId": "0123456789abcdefghijklmnop",
      "PoolId": "us-west-2_qRsTuVwXy",
      "Region": "us-west-2"
    }
  }
}
```

7. 해당 지역에 맞게 AmazonKinesisVideoDemoApp/src/main/java/com/amazonaws/kinesisvideo/demoapp/KinesisVideoDemoApp.java 업데이트하십시오 (다음 샘플에서는 US_WEST_2로 설정됨).

```
public class KinesisVideoDemoApp extends Application {
    public static final String TAG = KinesisVideoDemoApp.class.getSimpleName();
    public static Regions KINESIS_VIDEO_REGION = Regions.US_WEST_2;
```


AWS 리전 [상수에 대한 자세한 내용은 지역을 참조하십시오.](#)

다음 단계

[the section called “2단계: 코드 검사”](#)

2단계: 코드 검사

[Android 생산자 라이브러리 절차](#)의 이 단원에서는 예제 코드를 검사합니다.

Android 테스트 애플리케이션(AmazonKinesisVideoDemoApp)은 다음과 같은 코딩 패턴을 보입니다.

- KinesisVideoClient의 인스턴스를 만듭니다.
- MediaSource의 인스턴스를 만듭니다.
- 스트리밍을 시작합니다. MediaSource를 시작하면 데이터가 클라이언트로 전송되기 시작합니다.

다음 단원들에서 세부 정보가 제공됩니다.

인스턴스 만들기 KinesisVideoClient

[createKinesisVideoClient](#) 작업을 호출하여 [KinesisVideoClient](#) 객체를 생성합니다.

```
mKinesisVideoClient = KinesisVideoAndroidClientFactory.createKinesisVideoClient(
    getActivity(),
    KinesisVideoDemoApp.KINESIS_VIDEO_REGION,
    KinesisVideoDemoApp.getCredentialsProvider());
```

KinesisVideoClient가 네트워크 호출을 하려면 인증을 위한 자격 증명が必要です. 이전 섹션에서 수정한 파일에서 Amazon Cognito 자격 증명을 읽는 인스턴스를 전달합니다.

AWSCredentialsProvider awsconfiguration.json

의 인스턴스 생성 MediaSource

Kinesis 비디오 스트림으로 바이트를 전송하려면 데이터를 생성해야 합니다. Amazon Kinesis Video Streams는 데이터 소스를 나타내는 인터페이스를 제공합니다 [MediaSource](#).

예를 들어 Kinesis Video Streams Android 라이브러리는 [AndroidCameraMediaSource](#) 인터페이스 구현을 제공합니다 MediaSource. 이 클래스는 디바이스의 카메라 중 하나에서 데이터를 읽습니다.

다음 코드 예제([fragment/StreamConfigurationFragment.java](#) 파일의 예제)에서 미디어 원본의 구성이 생성됩니다.

```
private AndroidCameraMediaSourceConfiguration getCurrentConfiguration() {
return new AndroidCameraMediaSourceConfiguration(
    AndroidCameraMediaSourceConfiguration.builder()
        .withCameraId(mCamerasDropdown.getSelectedItem().getCameraId())

        .withEncodingMimeType(mMimeTypeDropdown.getSelectedItem().getMimeType())

        .withHorizontalResolution(mResolutionDropdown.getSelectedItem().getWidth())

        .withVerticalResolution(mResolutionDropdown.getSelectedItem().getHeight())
            .withCameraFacing(mCamerasDropdown.getSelectedItem().getCameraFacing())
            .withIsEncoderHardwareAccelerated(

mCamerasDropdown.getSelectedItem().isEncoderHardwareAccelerated())
            .withFrameRate(FRAMERATE_20)
            .withRetentionPeriodInHours(RETENTION_PERIOD_48_HOURS)
            .withEncodingBitRate(BITRATE_384_KBPS)
            .withCameraOrientation(-
mCamerasDropdown.getSelectedItem().getCameraOrientation())

        .withNalAdaptationFlags(StreamInfo.NalAdaptationFlags.NAL_ADAPTATION_ANNEXB_CPD_AND_FRAME_NALS
            .withIsAbsoluteTimecode(false));
}
}
```

다음 코드 예제([fragment/StreamingFragment.java](#) 파일의 예제)에서 미디어 원본이 생성됩니다.

```
mCameraMediaSource = (AndroidCameraMediaSource) mKinesisVideoClient
    .createMediaSource(mStreamName, mConfiguration);
```

미디어 소스 시작

미디어 원본이 데이터 생성 및 클라이언트로의 전송을 시작할 수 있도록 미디어 원본을 시작합니다. 다음 코드 예제는 [fragment/StreamingFragment.java](#) 파일의 예제입니다.

```
mCameraMediaSource.start();
```

다음 단계

[the section called “3단계: 코드 실행 및 확인”](#)

3단계: 코드 실행 및 확인

[Android 생산자 라이브러리](#)용 Android 예제 애플리케이션을 실행하려면 다음을 수행합니다.

1. Android 디바이스에 연결합니다.
2. [Run], [Run...], [Edit configurations...]를 차례로 선택합니다.
3. 더하기 아이콘 (+), Android 앱을 선택합니다. 이름 필드에 **AmazonKinesisVideoDemoApp**을 입력합니다. 모듈 폴다운에서 선택합니다. AmazonKinesisVideoDemoApp 확인을 선택합니다.
4. [Run]과 [Run]을 차례로 선택합니다.
5. [Select a Deployment Target] 화면에서 연결된 디바이스를 선택하고 [OK]를 선택합니다.
6. 장치의 AWSKinesisVideoDemoApp애플리케이션에서 새 계정 만들기를 선택합니다.
7. [USERNAME], [Password], [Given name], [Email address] 및 [Phone number]의 값을 입력한 다음 [Sign up]을 선택합니다.

Note

이러한 값의 제약 조건은 다음과 같습니다.

- Password: 대문자, 소문자, 숫자, 특수 문자를 포함해야 합니다. [Amazon Cognito](#) 콘솔의 사용자 풀 페이지에서 이러한 제약 조건을 변경할 수 있습니다.
- Email address: 확인 코드를 수신할 수 있는 유효한 주소여야 합니다.
- Phone number: +<Country code><Number> 형식이어야 합니다(예: +12065551212).

8. 이메일로 받은 코드를 입력하고 [Confirm] 을 선택합니다. 확인을 선택합니다.
9. 다음 페이지에서 기본값을 유지하고 [Stream] 을 선택합니다.
10. 에 AWS Management Console 로그인하고 미국 서부 (오레곤) 지역에서 [Kinesis Video Streams](#) 콘솔을 엽니다.

[Manage Streams] 페이지에서 [demo-stream]을 선택합니다.

11. 내장 플레이어에서 스트리밍 비디오가 재생됩니다. 프레임이 쌓여 비디오가 재생될 때까지 잠깐 기다려야 할 수 있습니다(일반적인 대역폭과 프로세서 상태에서 최대 10초).

Note

디바이스의 화면이 회전하면 (예: 세로에서 가로로) 애플리케이션이 비디오 스트리밍을 중지합니다.

코드 예제는 스트림을 생성합니다. 코드에 있는 `MediaSource`가 시작되면 카메라에서 `KinesisVideoClient`로 프레임을 전송하기 시작합니다. 그런 다음 클라이언트는 데모-스트림이라는 Kinesis 비디오 스트림으로 데이터를 전송합니다.

C++ 생산자 라이브러리 사용하기

Amazon Kinesis Video Streams에서 제공하는 C++ 프로듀서 라이브러리를 사용하여 디바이스에서 Kinesis 비디오 스트림으로 미디어 데이터를 전송하는 애플리케이션 코드를 작성할 수 있습니다.

객체 모델

C++ 라이브러리는 Kinesis 비디오 스트림으로의 데이터 전송을 관리하기 위해 다음과 같은 객체를 제공합니다.

- `KinesisVideoProducer`: 미디어 소스 및 AWS 자격 증명에 대한 정보를 포함하고 Kinesis Video Streams 이벤트를 보고하기 위한 콜백을 유지 관리합니다.
- `KinesisVideoStream`: Kinesis 비디오 스트림을 나타냅니다. 이름, 데이터 보존 기간, 미디어 콘텐츠 유형과 같은 비디오 스트림의 매개 변수에 대한 정보를 포함합니다.

스트림에 미디어 넣기

C++ 라이브러리에서 제공하는 메서드 (예: `PutFrame`) 를 사용하여 `KinesisVideoStream` 객체에 데이터를 넣을 수 있습니다. 그런 다음 라이브러리는 내부 데이터 상태를 관리하는데, 다음과 같은 작업을 포함할 수 있습니다.

- 인증 수행.
- 네트워크 지연 시간 감시. 지연 시간이 너무 길면 라이브러리가 프레임 드롭을 선택할 수 있습니다.
- 진행 중인 스트리밍의 상태 추적.

콜백 인터페이스

이 계층은 일단의 콜백 인터페이스를 노출시키는데, 애플리케이션 계층에 알리는 데 사용됩니다. 이들 콜백 인터페이스에는 다음이 포함됩니다.

- 서비스 콜백 인터페이스 (CallbackProvider): 라이브러리는 스트림을 생성하고, 스트림 설명을 가져오고, 스트림을 삭제할 때 이 인터페이스를 통해 얻은 이벤트를 호출합니다.
- 클라이언트 준비 상태 또는 스토리지 부족 이벤트 인터페이스(ClientCallbackProvider): 라이브러리는 클라이언트가 준비 상태에 있거나 가용한 스토리지 혹은 메모리가 부족할 수 있음을 감지하면 이 인터페이스에서 이벤트를 호출합니다.
- 스트림 이벤트 콜백 인터페이스(StreamCallbackProvider): 라이브러리는 스트림이 준비 상태에 돌입하거나, 프레임이 끊기거나 스트림 오류가 발생하는 이벤트가 발생할 때 이 인터페이스에 이벤트를 호출합니다.

Kinesis Video Streams는 이러한 인터페이스의 기본 구현을 제공합니다. 사용자 지정 네트워킹 로직이 필요하거나 사용자 인터페이스에 스토리지 부족 조건을 노출하려는 경우 등과 같이 사용자 지정 구현을 제공할 수도 있습니다.

생산자 라이브러리의 콜백에 대한 자세한 내용은 [프로듀서 SDK 콜백](#) 단원을 참조하십시오.

절차: C++ 생산자 SDK 사용

이 절차는 C++ 애플리케이션의 Kinesis Video Streams 클라이언트와 미디어 소스를 사용하여 Kinesis 비디오 스트림으로 데이터를 보내는 방법을 보여줍니다.

이 절차에는 다음 단계가 포함됩니다.

- [1 단계: 코드 다운로드 및 구성](#)
- [2 단계: 코드 쓰기 및 검사](#)
- [3 단계: 코드 실행 및 확인](#)

필수 조건

- 자격 증명: 샘플 코드에서는 자격 증명 프로필 파일에 설정한 프로필을 지정하여 자격 증명을 제공합니다. AWS 아직 설정하지 않았다면 먼저 자격 증명 프로필을 설정합니다.

자세한 내용은 [개발용 AWS 자격 증명 및 지역 설정](#)을 참조하십시오.

- 인증서 저장소 통합: Kinesis Video Streams 프로듀서 라이브러리는 호출하는 서비스와 신뢰를 구축해야 합니다. 이는 공용 인증서 저장소의 인증 기관 (CA) 의 유효성을 검사하여 수행됩니다. Linux 기반 모델은 이 스토어가 /etc/ssl/ 디렉터리에 위치합니다.

다음 위치에서 인증 스토어로 인증서를 다운로드합니다.

<https://www.amazontrust.com/repository/SFSRootCAG2.pem>

- 다음의 macOS용 빌드 종속성을 설치합니다.
 - [Autoconf 2.69](#) (라이선스 GPLv3+/Autoconf: GNU GPL 버전 3 이상)
 - [CMake 3.7 또는 3.8](#)
 - [Pkg-Config](#)
 - xCode(macOS)/clang/gcc(xcode-select 버전 2347)
 - Java Development Kit (JDK) (Java JNI 컴파일용)
 - [Lib-Pkg](#)
- Ubuntu용 다음 빌드 종속 항목을 설치하세요.
 - Git: `sudo apt install git`
 - [CMake](#): `sudo apt install cmake`
 - G++: `sudo apt install g++`
 - 패키지 구성: `sudo apt install pkg-config`
 - OpenJDK: `sudo apt install openjdk-8-jdk`

Note

이는 자바 네이티브 인터페이스 (JNI) 를 빌드하는 경우에만 필요합니다.

- JAVA_HOME 환경 변수 설정: `export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/`

다음 단계

[1 단계: C++ 생산자 라이브러리 코드 다운로드 및 구성](#)

1단계: C++ 프로듀서 라이브러리 코드 다운로드 및 구성

C++ 프로듀서 라이브러리를 다운로드하고 구성하는 방법에 대한 자세한 내용은 [Amazon Kinesis Video Streams CPP 프로듀서, GStreamer](#) 플러그인 및 JNI를 참조하십시오.

[이 예제에 대한 사전 요구 사항 및 자세한 내용은 C++ 프로듀서 라이브러리 사용을 참조하십시오.](#)

다음 단계

[2단계: 코드 작성 및 검토](#)

2단계: 코드 작성 및 검토

[C++ 생산자 라이브러리 절차](#)의 이 단원에서는 C++ 테스트 도구(`tst/ProducerTestFixture.h` 및 기타 다른 파일)에서 코드를 검사합니다. 이전 단원에서 이 코드를 다운로드했습니다.

비종속 플랫폼 C++ 예제는 다음과 같은 코딩 패턴을 보여 줍니다.

- Kinesis Video `KinesisVideoProducer` Streams에 액세스할 인스턴스를 생성합니다.
- `KinesisVideoStream`의 인스턴스를 만듭니다. 이렇게 하면 같은 이름의 스트림이 아직 없는 AWS 계정 경우 Kinesis 비디오 스트림이 생성됩니다.
- 가용한 경우 모든 데이터 프레임에 대해 `putFrame`을 `KinesisVideoStream`에 호출하여 스트림에 전송합니다.

다음 섹션에서는 이 코딩 패턴에 대한 자세한 정보를 제공합니다.

인스턴스 만들기 `KinesisVideoProducer`

`KinesisVideoProducer::createSync` 메서드를 호출하여 `KinesisVideoProducer` 객체를 생성합니다. 다음 예제에서는 `ProducerTestFixture.h` 파일에서 `KinesisVideoProducer`를 생성합니다.

```
kinesis_video_producer_ = KinesisVideoProducer::createSync(move(device_provider_),
    move(client_callback_provider_),
    move(stream_callback_provider_),
    move(credential_provider_),
    defaultRegion_);
```

`createSync` 방법으로 다음 파라미터를 취합니다.

- 디바이스 또는 스토리지 구성에 관한 정보를 포함한 `DeviceInfoProvider` 객체를 반환하는 `DeviceInfo` 객체.

Note

`deviceInfo.storageInfo.storageSize` 파라미터를 사용하여 콘텐츠 스토어 크기를 구성합니다. 콘텐츠 스트림은 콘텐츠 스토어를 공유합니다. 스토리지 크기 요구 사항을 확인하려면 평균 프레임 크기에 모든 스트림의 최대 지속 시간 동안 저장된 프레임 수를 곱합니다. 그런 다음 조각 모음을 고려하여 1.2를 곱합니다. 예를 들어 애플리케이션에 다음 구성이 있는 것으로 가정하겠습니다.

- 스트림 세 개
- 최대 지속 시간 3분
- 각 스트림은 초당 30프레임(FPS)
- 프레임마다 10,000KB

이 애플리케이션의 콘텐츠 저장소 요구 사항은 $3 \text{ (스트림)} * 3 \text{ (분)} * 60 \text{ (분당 초)} * 10000 \text{ (kb)} * 1.2 \text{ (조각 모음 허용량)} = 194.4\text{Mb} \sim 200\text{Mb}$ 입니다.

- 특정 클라이언트의 이벤트를 보고하는 함수 별칭을 반환하는 `ClientCallbackProvider` 객체.
- 특정 스트림의 이벤트가 발생할 때 콜백되는 함수 별칭을 반환하는 `StreamCallbackProvider` 객체.
- 자격 증명 환경 변수에 대한 액세스를 제공하는 `CredentialProvider` 객체입니다. AWS
- AWS 리전 ("us-west-2"). 서비스 엔드포인트는 리전으로부터 결정됩니다.

의 인스턴스 만들기 `KinesisVideoStream`

`StreamDefinition` 파라미터로 `KinesisVideoProducer::CreateStream` 메서드를 호출하여 `KinesisVideoStream` 객체를 생성합니다. 예제에서는 `ProducerTestFixture.h` 파일에서 트랙 유형을 비디오로, 트랙 ID를 1로 지정하여 `KinesisVideoStream`을 생성합니다.

```
auto stream_definition = make_unique<StreamDefinition>(stream_name,
                                                    hours(2),
                                                    tags,
                                                    "",
                                                    STREAMING_TYPE_REALTIME,
                                                    "video/h264",
                                                    milliseconds::zero(),
                                                    seconds(2),
                                                    milliseconds(1),
                                                    true,
```



```

        true,
        true);
return kinesis_video_producer_->createStream(move(stream_definition));

```

StreamDefinition 객체에는 다음의 필드가 있습니다.

- 스트림 이름.
- 데이터 보존 기간.
- 스트림 태그. 이들 태그는 소비자 애플리케이션이 정확한 스트림을 찾거나 스트림에 관한 상세를 정보를 얻는데 사용할 수 있습니다. AWS Management Console에서도 태그를 볼 수 있습니다.
- AWS KMS 스트림의 암호화 키. 자세한 내용은 [Kinesis Video Streams에서의 서버 측 암호화 사용을](#) 참조하십시오.
- 스트리밍 유형. 현재 유일한 유효 값은 STREAMING_TYPE_REALTIME입니다.
- 미디어 콘텐츠 유형.
- 미디어 지연 시간. 이 값은 현재 사용되지 않으므로 0으로 설정해야 합니다.
- 각 조각의 재생 시간.
- 미디어 타임코드 척도.
- 미디어가 키 프레임 조각을 사용하는지 여부.
- 미디어가 타임코드를 사용하는지 여부.
- 미디어가 절대 조각 시간을 사용하는지 여부.

Kinesis 비디오 스트림에 오디오 트랙 추가

다음의 AddTrack 메서드를 사용하여 비디오 트랙 스트림 정의에 오디오 트랙 세부 정보를 추가할 수 있습니다. StreamDefinition

```

stream_definition->addTrack(DEFAULT_AUDIO_TRACKID, DEFAULT_AUDIO_TRACK_NAME,
    DEFAULT_AUDIO_CODEC_ID, MKV_TRACK_INFO_TYPE_AUDIO);

```

addTrack 방법에는 다음의 파라미터가 필요합니다.

- 트랙 ID (오디오의 경우와 동일). 이 값은 고유한 값으로서 0이 아니어야 합니다.
- 사용자 정의 트랙 이름 (예: 오디오 트랙의 경우 “오디오”)

- 이 트랙의 코덱 ID (예: 오디오 트랙 “A_AAC”의 경우)
- 트랙 유형 (예: 오디오의 경우 MKV_TRACK_INFO_TYPE_AUDIO의 열거형 값 사용)

오디오 트랙용 코덱 프라이빗 데이터가 있는 경우, addTrack 함수를 호출할 때 해당 데이터를 전달할 수 있습니다. 객체를 만든 후 에서 시작 메서드를 호출하면서 코덱 개인 데이터를 전송할 수도 있습니다. KinesisVideoStream KinesisVideoStream

Kinesis 비디오 스트림에 프레임 넣기

헤더와 미디어 데이터가 포함된 Frame 객체를 전달하여 Kinesis 비디오 스트림에 미디어를 넣습니다. KinesisVideoStream::putFrame 예제에서는 ProducerApiTest.cpp 파일에서 putFrame를 호출합니다.

```

frame.duration = FRAME_DURATION_IN_MICROS * HUNDREDS_OF_NANOS_IN_A_MICROSECOND;
frame.size = sizeof(frameBuffer_);
frame.frameData = frameBuffer_;
memset(frame.frameData, 0x55, frame.size);

while (!stop_producer_) {
    // Produce frames
    timestamp = std::chrono::duration_cast<std::chrono::nanoseconds>(
        std::chrono::system_clock::now().time_since_epoch()).count() /
    DEFAULT_TIME_UNIT_IN_NANOS;
    frame.index = index++;
    frame.decodingTs = timestamp;
    frame.presentationTs = timestamp;

    // Key frame every 50th
    frame.flags = (frame.index % 50 == 0) ? FRAME_FLAG_KEY_FRAME : FRAME_FLAG_NONE;
    ...

    EXPECT_TRUE(kinesis_video_stream->putFrame(frame));

```

Note

앞선 C++ 생산자 예제에서는 테스트 데이터의 버퍼를 전송합니다. 실제 애플리케이션에서는 미디어 소스(예: 카메라)의 프레임 데이터로부터 프레임 버퍼 및 크기를 알아야 합니다.

Frame 객체에는 다음의 필드가 있습니다.

- 프레임 인덱스. 이는 단순 점증 값이어야 합니다.
- 프레임과 관련된 플래그. 예를 들어, 인코더가 키 프레임을 생산하도록 구성되어 있다면 이 프레임에는 `FRAME_FLAG_KEY_FRAME` 플래그가 할당될 것입니다.
- 디코딩 타임스탬프.
- 프레젠테이션 타임스탬프.
- 프레임 지속 시간(최대 100 ns 단위).
- 프레임 크기(바이트).
- 프레임 데이터.

프레임 형식에 대한 자세한 내용은 [Kinesis Video Streams 데이터 모델](#)을 참조하십시오.

특정 KinesisVideoFrame 트랙에 넣기 KinesisVideoStream

PutFrameHelper 클래스를 사용하여 프레임 데이터를 특정 트랙에 넣을 수 있습니다. 먼저 Buffer를 호출하여 데이터를 채울 사전 할당된 getFrameData 버퍼 중 하나에 대한 포인터를 가져옵니다. KinesisVideoFrame 그런 다음 putFrameMulti Track을 호출하여 프레임 데이터의 유형을 나타내는 Boolean 값과 KinesisVideoFrame 함께 전송할 수 있습니다. 비디오 데이터이면 true를 사용하고, 프레임에 오디오 데이터가 포함되어 있으면 false를 사용합니다. putFrameMultiTrack 메서드는 큐 메커니즘을 사용하여 MKV 프래그먼트가 단조롭게 증가하는 프레임 타임스탬프를 유지하고 두 프래그먼트가 겹치지 않도록 합니다. 예를 들어 프래그먼트의 첫 번째 프레임의 MKV 타임스탬프는 항상 이전 프래그먼트의 마지막 프레임의 MKV 타임스탬프보다 커야 합니다.

에는 다음과 같은 필드가 있습니다. PutFrameHelper

- 대기열에 있는 최대 오디오 프레임 수
- 대기열에 있는 최대 비디오 프레임 수
- 단일 오디오 프레임에 할당할 크기입니다.
- 단일 비디오 프레임에 할당할 크기.

지표 및 지표 로깅

C++ 생산자 SDK에는 메트릭 및 메트릭 로깅 기능이 포함되어 있습니다.

getKinesisVideoMetrics 및 getKinesisVideoStreamMetrics API 작업을 사용하여 Kinesis Video Streams 및 활성 스트림에 대한 정보를 검색할 수 있습니다.

다음은 `kinesis-video-pic/src/client/include/com/amazonaws/kinesis/video/client/Include.h` 파일의 코드입니다.

```
/**
 * Gets information about the storage availability.
 *
 * @param 1 CLIENT_HANDLE - the client object handle.
 * @param 2 PKinesisVideoMetrics - OUT - Kinesis Video metrics to be filled.
 *
 * @return Status of the function call.
 */
PUBLIC_API STATUS getKinesisVideoMetrics(CLIENT_HANDLE, PKinesisVideoMetrics);

/**
 * Gets information about the stream content view.
 *
 * @param 1 STREAM_HANDLE - the stream object handle.
 * @param 2 PStreamMetrics - Stream metrics to fill.
 *
 * @return Status of the function call.
 */
PUBLIC_API STATUS getKinesisVideoStreamMetrics(STREAM_HANDLE, PStreamMetrics);
```

`getKinesisVideoMetrics`로 채워지는 `PClientMetrics` 객체에는 다음 정보가 포함되어 있습니다.

- `contentStoreSize`: 콘텐츠 스토어의 전체 크기 (스트리밍 데이터를 저장하는 데 사용되는 메모리) (바이트).
- `contentStoreAvailable` 크기: 콘텐츠 저장소에서 사용 가능한 메모리 (바이트).
- `contentStoreAllocated` 크기: 콘텐츠 저장소에 할당된 메모리입니다.
- `totalContentViews` 크기: 콘텐츠 뷰에 사용된 총 메모리입니다. 콘텐츠 보기는 콘텐츠 저장소에 있는 일련의 정보 인덱스입니다.
- `totalFrameRate`: 모든 활성 스트림의 초당 총 프레임 수입니다.
- `totalTransferRate`: 모든 스트림에서 전송되는 총 초당 비트 수 (bps) 입니다.

`getKinesisVideoStreamMetrics`로 채워지는 `PStreamMetrics` 객체에는 다음 정보가 포함되어 있습니다.

- `currentViewDuration`: 콘텐츠 뷰의 헤드 (프레임이 인코딩된 경우) 와 현재 위치 (프레임 데이터가 Kinesis Video Streams로 전송되는 경우) 간의 100ns 단위 차이.
- `overallViewDuration`: 콘텐츠 뷰의 헤드 (프레임이 인코딩될 때) 부터 끝까지 (콘텐츠 뷰에 할당된 총 공간을 초과했거나 Kinesis Video Streams에서 `PersistedAck` 메시지를 수신하고 지속되는 것으로 알려진 프레임이 플러시되어 프레임이 메모리에서 플러시되는 경우) 사이의 100ns 단위 차이.
- `currentViewSize`: 헤드 (프레임이 인코딩된 경우) 에서 현재 위치 (프레임이 Kinesis Video Streams로 전송될 때) 까지의 콘텐츠 뷰 크기 (바이트) 입니다.
- `overallViewSize`: 콘텐츠 보기의 총 크기 (바이트).
- `currentFrameRate`: 마지막으로 측정한 스트림 속도 (초당 프레임 수).
- `currentTransferRate`: 마지막으로 측정한 스트림 속도 (초당 바이트 수).

해제

나머지 바이트를 버퍼에 넣고 ACK를 대기하려는 경우 `stopSync`를 사용할 수 있습니다.

```
kinesis_video_stream->stopSync();
```

또는 `stop`를 호출하여 스트리밍을 종료할 수 있습니다.

```
kinesis_video_stream->stop();
```

스트리밍이 중지된 후 다음 API를 호출하여 스트림을 해제할 수 있습니다.

```
kinesis_video_producer_->freeStream(kinesis_video_stream);
```

다음 단계

[the section called “3단계: 코드 실행 및 확인”](#)

3단계: 코드 실행 및 확인

[C++ 제작자 라이브러리 프로시저의](#) 코드를 실행하고 확인하려면 다음 OS별 지침을 참조하십시오.

- [Linux](#)
- [macOS](#)
- [Windows](#)
- [라즈베리 파이 OS](#)

Amazon CloudWatch 콘솔에서 스트림과 관련된 지표 (예:) 를 보면서 스트림의 트래픽을 모니터링할 수 `PutMedia.IncomingBytes` 있습니다.

C++ 프로듀서 SDK를 GStreamer 플러그인으로 사용

[GStreamer](#)는 여러 카메라 및 비디오 소스에서 모듈식 플러그인을 결합하여 사용자 지정 미디어 파이프라인을 만들 때 사용하는 인기 있는 미디어 프레임워크입니다. Kinesis Video Streams GStreamer 플러그인은 기존 GStreamer 미디어 파이프라인과 Kinesis Video Streams의 통합을 간소화합니다.

C++ 생산자 SDK를 GStreamer 플러그인으로 사용하는 방법에 대한 자세한 내용은 [예: Kinesis Video Streams 프로듀서 SDK GStreamer 플러그인 - kvssink](#) 단원을 참조하십시오.

C++ 프로듀서 SDK를 도커 컨테이너에서 GStreamer 플러그인으로 사용

[GStreamer](#)는 여러 카메라 및 비디오 소스에서 모듈식 플러그인을 결합하여 사용자 지정 미디어 파이프라인을 만드는 데 사용하는 인기 있는 미디어 프레임워크입니다. Kinesis Video Streams GStreamer 플러그인은 기존 GStreamer 미디어 파이프라인과 Kinesis Video Streams의 통합을 간소화합니다.

또한 [Docker](#)를 사용하여 GStreamer 파이프라인을 생성하면 Kinesis Video Streams의 운영 환경이 표준화되어 애플리케이션 구축 및 실행이 간소화됩니다.

Docker 컨테이너에서 C++ 생산자 SDK를 GStreamer 플러그인으로 사용하는 방법에 대한 자세한 내용은 [도커 컨테이너에서 GStreamer 요소를 실행합니다](#) 단원을 참조하십시오.

C++ 프로듀서 SDK를 통한 로깅 사용

`kvs_log_configuration` 폴더의 `kinesis-video-native-build` 파일에서 C++ 생산자 SDK 애플리케이션에 대한 로깅을 구성합니다.

다음 예제는 DEBUG 수준 로그 항목을 AWS Management Console에 기록하는 애플리케이션을 구성하는 기본 구성 파일의 첫 번째 줄을 보여줍니다.

```
log4cplus.rootLogger=DEBUG, KvsConsoleAppender
```

로깅 수준을 INFO로 설정하여 로깅 상세도를 낮출 수 있습니다.

로그 파일에 로그 항목을 기록하도록 애플리케이션을 구성하려면 파일의 첫 번째 줄을 다음과 같이 업데이트하십시오.

```
log4cplus.rootLogger=DEBUG, KvsConsoleAppender, KvsFileAppender
```

그러면 애플리케이션이 `kvs.log` 폴더의 `kinesis-video-native-build/log`에 로그 항목을 기록하도록 구성됩니다.

로그 파일 위치를 변경하려면 다음 줄을 새 경로로 수정합니다.

```
log4cplus.appender.KvsFileAppender.File=../Log/kvs.log
```

Note

DEBUG 수준 로깅이 파일에 기록될 경우 로그 파일이 디바이스의 가용 저장 공간을 빠르게 소모할 수 있습니다.

C 생산자 라이브러리 사용

Amazon Kinesis Video Streams에서 제공하는 C 프로듀서 라이브러리를 사용하여 디바이스에서 Kinesis 비디오 스트림으로 미디어 데이터를 전송하는 애플리케이션 코드를 작성할 수 있습니다.

객체 모델

Kinesis Video Streams C 프로듀서 라이브러리는 플랫폼 독립 코드베이스 (PIC) 라는 공통 구성 요소를 기반으로 하며, 이 구성 요소는 <https://github.com/aws-labs/-pic>에서 [amazon-kinesis-video-streams](#) 사용할 수 있습니다 GitHub . PIC에는 기본 구성 요소에 대한 플랫폼 독립적인 비즈니스 로직이 포함되어 있습니다. Kinesis Video Streams C 프로듀서 라이브러리는 시나리오 및 플랫폼별 콜백 및 이벤트를 허용하는 추가 API 계층으로 PIC를 래핑합니다. Kinesis Video Streams C 프로듀서 라이브러리에는 PIC를 기반으로 구축된 다음과 같은 구성 요소가 있습니다.

- 디바이스 정보 제공자 — PIC API에 직접 제공할 수 있는 DeviceInfo 구조를 공개합니다. 애플리케이션이 처리하는 스트림의 수와 유형, 사용 가능한 RAM의 양에 따라 구성된 필수 버퍼링의 양을 기반으로 콘텐츠 저장소를 최적화할 수 있는 애플리케이션 시나리오에 최적화된 제공자를 비롯한 제공자 세트를 구성할 수 있습니다.

- 스트림 정보 제공자 — PIC API에 직접 제공할 수 있는 `StreamInfo` 구조를 공개합니다. 애플리케이션 유형과 일반적인 유형의 스트리밍 시나리오에 맞는 제공자 집합이 있습니다. 여기에는 비디오, 오디오, 오디오 및 비디오 멀티트랙과 같은 공급자가 포함됩니다. 각 시나리오에는 응용 프로그램 요구 사항에 따라 사용자 지정할 수 있는 기본값이 있습니다.
- 콜백 제공자 - PIC API에 직접 제공할 수 있는 `ClientCallbacks` 구조를 공개합니다. 여기에는 네트워킹 (CURL 기반 API 콜백), 권한 부여 (AWS 자격 증명 API), 오류 콜백에 대한 재시도 스트리밍을 위한 콜백 제공자 집합이 포함됩니다. 콜백 공급자 API는 여러 인수를 사용하여 구성합니다 (예: 및 권한 부여 정보). AWS 리전 또는 IoT 인증서를 사용하거나 `AWS AccessKeyId SecretKey`, 또는 `SessionToken`를 사용하여 수행됩니다. 애플리케이션에서 일부 애플리케이션별 로직을 구현하기 위해 특정 콜백을 추가로 처리해야 하는 경우 사용자 지정 콜백으로 콜백 공급자를 향상할 수 있습니다.
- `FrameOrderCoordinator`— 멀티트랙 시나리오의 오디오 및 비디오 동기화를 처리하는 데 도움이 됩니다. 기본 동작이 있어 애플리케이션의 특정 로직을 처리하도록 사용자 지정할 수 있습니다. 또한 하위 계층 PIC API로 제출하기 전에 PIC 프레임 구조의 프레임 메타데이터 패키징을 간소화합니다. 멀티트랙이 아닌 시나리오의 경우 이 구성 요소는 PIC `putFrame` API에 대한 패스스루입니다.

C 라이브러리는 Kinesis 비디오 스트림에 대한 데이터 전송을 관리하기 위해 다음과 같은 객체를 제공합니다.

- `KinesisVideoClient`— 디바이스에 대한 정보를 포함하고 Kinesis Video Streams 이벤트를 보고하기 위한 콜백을 유지 관리합니다.
- `KinesisVideoStream`— 이름, 데이터 보존 기간, 미디어 콘텐츠 유형과 같은 비디오 스트림의 파라미터에 대한 정보를 나타냅니다.

스트림에 미디어 넣기

C 라이브러리에서 제공하는 메서드 (예: `PutKinesisVideoFrame`) 를 사용하여 `KinesisVideoStream` 객체에 데이터를 넣을 수 있습니다. 그런 다음 라이브러리는 내부 데이터 상태를 관리하는데, 다음과 같은 작업을 포함할 수 있습니다.

- 인증 수행.
- 네트워크 지연 시간 감시. 지연 시간이 너무 길면 라이브러리가 프레임 드롭을 선택할 수 있습니다.
- 진행 중인 스트리밍의 상태 추적.

절차: C 생산자 SDK 사용

이 절차는 C 애플리케이션의 Kinesis Video Streams 클라이언트와 미디어 소스를 사용하여 H.264로 인코딩된 비디오 프레임을 Kinesis 비디오 스트림으로 보내는 방법을 보여줍니다.

이 절차에는 다음 단계가 포함됩니다.

- [1단계: C 프로듀서 라이브러리 코드 다운로드](#)
- [2단계: 코드 작성 및 검토](#)
- [3단계: 코드 실행 및 확인](#)

필수 조건

- 자격 증명 - 샘플 코드에서는 자격 증명 프로필 파일에 설정한 프로필을 지정하여 자격 증명을 제공합니다. AWS 아직 설정하지 않았다면 먼저 자격 증명 프로필을 설정합니다.

자세한 내용은 [개발용 AWS 자격 증명 및 지역 설정](#)을 참조하십시오.

- 인증서 저장소 통합 - Kinesis Video Streams 프로듀서 라이브러리는 호출하는 서비스와 신뢰를 구축해야 합니다. 이는 공용 인증서 저장소의 인증 기관 (CA) 의 유효성을 검사하여 수행됩니다. Linux 기반 모델은 이 스토어가 /etc/ssl/ 디렉터리에 위치합니다.

다음 위치에서 인증 스토어로 인증서를 다운로드합니다.

<https://www.amazontrust.com/repository/SFSRootCAG2.pem>

- 다음의 macOS용 빌드 종속성을 설치합니다.
 - [Autoconf 2.69](#) (라이선스 GPLv3+/Autoconf: GNU GPL 버전 3 이상)
 - [CMake 3.7 또는 3.8](#)
 - [Pkg-Config](#)
 - xCode(macOS)/clang/gcc(xcode-select 버전 2347)
 - Java Development Kit (JDK) (Java JNI 컴파일용)
 - [Lib-Pkg](#)
- Ubuntu용 다음 빌드 종속 항목을 설치하세요.
 - Git: `sudo apt install git`
 - [CMake](#): `sudo apt install cmake`
 - G++: `sudo apt install g++`

- 패키지 구성: `sudo apt install pkg-config`
- OpenJDK: `sudo apt install openjdk-8-jdk`
- JAVA_HOME 환경 변수 설정: `export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/`

다음 단계

[1단계: C 프로듀서 라이브러리 코드 다운로드](#)

1단계: C 프로듀서 라이브러리 코드 다운로드

이 단원에서는 하위 수준 라이브러리를 다운로드합니다. 사전 조건과 이 예제에 대한 기타 세부 정보는 [C 생산자 라이브러리 사용](#)을 참조하십시오.

1. 디렉토리를 만든 다음 GitHub 리포지토리에서 예제 소스 코드를 복제합니다.

```
git clone --recursive https://github.com/aws-labs/amazon-kinesis-video-streams-producer-c.git
```

Note

--recursive와 함께 git clone을 실행하지 않는 경우 amazon-kinesis-video-streams-producer-c/open-source 디렉토리에서 `git submodule update --init`을 실행합니다. 또한 pkg-config, CMake 및 빌드 환경을 설치해야 합니다. [자세한 내용은 https://github.com/aws-labs/ -producer-c.git의 내용을 참조하십시오. README.md amazon-kinesis-video-streams](https://github.com/aws-labs/amazon-kinesis-video-streams-producer-c/blob/master/README.md)

2. 여러분이 선택한 통합 개발 환경(IDE)(예: [Eclipse](#))에서 코드를 엽니다.

다음 단계

[2단계: 코드 작성 및 검토](#)

2단계: 코드 작성 및 검토

이 섹션에서는 [https://github.com/aws-labs/ amazon-kinesis-video-streams -producer-c](https://github.com/aws-labs/amazon-kinesis-video-streams-producer-c) 리포지토리의 `KvsVideoOnlyStreamingSample.c samples` 폴더에 있는 샘플 애플리케이션의 코드를 살

퍼봅니다. GitHub 이전 단계에서 이 코드를 다운로드했습니다. 이 샘플은 C 프로듀서 라이브러리를 사용하여 폴더 내의 H.264로 인코딩된 비디오 프레임을 Kinesis 비디오 스트림으로 samples/h264SampleFrames 보내는 방법을 보여줍니다.

이 샘플 애플리케이션에는 세 개의 섹션이 있습니다.

- 초기화 및 구성:
 - 플랫폼별 미디어 파이프라인 초기화 및 구성.
 - 파이프라인 초기화 KinesisVideoClient 및 KinesisVideoStream 구성, 콜백 설정, 시나리오별 인증 통합, 코덱 개인 데이터 추출 및 제출, 스트림을 READY 상태로 만들기
- 기본 루프:
 - 타임스탬프 및 플래그와 함께 미디어 파이프라인에서 프레임 가져오기.
 - KinesisVideoStream에 프레임 제출.
- 해제:
 - 중지 (동기화) KinesisVideoStream, 해제 KinesisVideoStream, 해제 KinesisVideoClient

이 샘플 애플리케이션은 다음 작업을 완료합니다.

- createDefaultDeviceInfo API를 호출하여 디바이스 또는 스토리지 구성에 대한 정보가 포함된 deviceInfo 객체를 생성합니다.

```
// default storage size is 128MB. Use setDeviceInfoStorageSize after create to change
// storage size.
CHK_STATUS(createDefaultDeviceInfo(&pDeviceInfo));
// adjust members of pDeviceInfo here if needed
pDeviceInfo->clientInfo.loggerLogLevel = LOG_LEVEL_DEBUG;
```

- createRealtimeVideoStreamInfoProvider API를 호출하여 StreamInfo 객체를 생성합니다.

```
CHK_STATUS(createRealtimeVideoStreamInfoProvider(streamName,
// adjust members of pStreamInfo here if needed
DEFAULT_RETENTION_PERIOD, DEFAULT_BUFFER_DURATION, &pStreamInfo));
```

- `createDefaultCallbacksProviderWithAwsCredentialsAPI`를 호출하여 정적 자격 증명을 기반으로 기본 콜백 공급자를 생성합니다. AWS

```
CHK_STATUS(createDefaultCallbacksProviderWithAwsCredentials(accessKey,
                                                           secretKey,
                                                           sessionToken,
                                                           MAX_UINT64,
                                                           region,
                                                           cacertPath,
                                                           NULL,
                                                           NULL,
                                                           FALSE,
                                                           &pClientCallbacks));
```

- `createKinesisVideoClientAPI`를 호출하여 디바이스 스토리지에 대한 정보를 포함하고 Kinesis Video Streams 이벤트를 보고하기 위한 콜백을 유지 관리하는 `KinesisVideoClient` 객체를 생성합니다.

```
CHK_STATUS(createKinesisVideoClient(pDeviceInfo, pClientCallbacks, &clientHandle));
```

- `createKinesisVideoStreamSync` API를 호출하여 `KinesisVideoStream` 객체를 생성합니다.

```
CHK_STATUS(createKinesisVideoStreamSync(clientHandle, pStreamInfo, &streamHandle));
```

- 샘플 프레임을 설정하고 `PutKinesisVideoFrame` API를 호출하여 해당 프레임을 `KinesisVideoStream` 객체로 전송합니다.

```
// setup sample frame
MEMSET(frameBuffer, 0x00, frameSize);
frame.frameData = frameBuffer;
frame.version = FRAME_CURRENT_VERSION;
frame.trackId = DEFAULT_VIDEO_TRACK_ID;
frame.duration = HUNDREDS_OF_NANOS_IN_A_SECOND / DEFAULT_FPS_VALUE;
frame.decodingTs = defaultGetTime(); // current time
frame.presentationTs = frame.decodingTs;
```

```

while(defaultGetTime() > streamStopTime) {
    frame.index = frameIndex;
    frame.flags = fileIndex % DEFAULT_KEY_FRAME_INTERVAL == 0 ?
FRAME_FLAG_KEY_FRAME : FRAME_FLAG_NONE;
    frame.size = sizeof(frameBuffer);

    CHK_STATUS(readFrameData(&frame, frameFilePath));

    CHK_STATUS(putKinesisVideoFrame(streamHandle, &frame));
    defaultThreadSleep(frame.duration);

    frame.decodingTs += frame.duration;
    frame.presentationTs = frame.decodingTs;
    frameIndex++;
    fileIndex++;
    fileIndex = fileIndex % NUMBER_OF_FRAME_FILES;
}

```

- 해제:

```

CHK_STATUS(stopKinesisVideoStreamSync(streamHandle));
CHK_STATUS(freeKinesisVideoStream(&streamHandle));
CHK_STATUS(freeKinesisVideoClient(&clientHandle));

```

다음 단계

[3단계: 코드 실행 및 확인](#)

3단계: 코드 실행 및 확인

[생산자 라이브러리 절차](#)의 코드를 실행하고 확인하려면 다음을 수행합니다.

1. 다음 명령어를 실행하여 [다운로드한 C SDK에 build](#) 디렉터리를 만들고 해당 cmake 디렉터리에서 실행합니다.

```
mkdir -p amazon-kinesis-video-streams-producer-c/build;
```

```
cd amazon-kinesis-video-streams-producer-c/build;
cmake ..
```

cmake ..에 다음 옵션을 사용할 수 있습니다.

- -DBUILD_DEPENDENCIES- 소스에서 종속 라이브러리를 빌드할지 여부.
- -DBUILD_TEST=TRUE- 빌드 유닛 및 통합 테스트 기기 지원을 확인하는 데 유용할 수 있습니다.

```
./tst/webrtc_client_test
```

- -DCODE_COVERAGE- 커버리지 리포트를 활성화합니다.
- -DCOMPILER_WARNINGS- 모든 컴파일러 경고를 활성화합니다.
- -DADDRESS_SANITIZER- 로 빌드하세요. AddressSanitizer
- -DMEMORY_SANITIZER- 로 빌드하세요 MemorySanitizer.
- -DTHREAD_SANITIZER- 로 빌드하세요 ThreadSanitizer.
- -DUNDEFINED_BEHAVIOR_SANITIZER- 로 빌드하세요 UndefinedBehaviorSanitizer.
- -DALIGNED_MEMORY_MODEL - 정렬된 메모리 모델 전용 디바이스용으로 빌드 기본값은 OFF입니다.

2. 이전 단계에서 방금 생성한 build 디렉토리로 이동하고 를 make 실행하여 WebRTC C SDK와 제공된 샘플을 빌드합니다.

```
make
```

3. 샘플 애플리케이션은 폴더 내의 h.264로 인코딩된 비디오 프레임을 Kinesis samples/h264SampleFrames Video Streams로 kinesis_video_cproducer_video_only_sample 보냅니다. 다음 명령은 비디오 프레임을 10초 동안 루프로 Kinesis Video Streams로 보냅니다.

```
./kinesis_video_cproducer_video_only_sample YourStreamName 10
```

다른 폴더 (예:MyH264FramesFolder) 에서 H.264로 인코딩된 프레임을 보내려면 다음 인수를 사용하여 샘플을 실행하십시오.

```
./kinesis_video_cproducer_video_only_sample YourStreamName 10 MyH264FramesFolder
```

4. 세부 정보 표시 로그를 활성화하려면 CMakeList.txt에서 해당 라인의 주석을 제거하여 HEAP_DEBUG 및 LOG_STREAMING C-정의를 정의합니다.

IDE의 디버그 출력에서 테스트 제품군의 진행 상황을 모니터링할 수 있습니다. Amazon CloudWatch 콘솔에서 스트림과 관련된 지표 (예:) 를 보면서 스트림의 트래픽을 모니터링할 수도 PutMedia.IncomingBytes 있습니다.

Note

테스트 도구가 빈 바이트의 프레임만 전송하기 때문에 콘솔은 데이터를 비디오 스트림으로 표시하지 않습니다.

Raspberry Pi에서 C++ 생산자 SDK 사용

Raspberry Pi는 기본 컴퓨터 프로그래밍 기술을 가르치고 학습하는 데 사용할 수 있는 저렴한 소형 컴퓨터입니다. 이 자습서에서는 라즈베리 파이 디바이스에서 Amazon Kinesis Video Streams C++ 프로듀서 SDK를 설정하고 사용하는 방법을 설명합니다. 또한 이러한 단계에는 GStreamer 데모 애플리케이션을 사용하여 설치를 확인하는 방법도 포함되어 있습니다.

주제

- [사전 조건](#)
- [Kinesis Video Streams에 쓰기 권한을 가진 IAM 사용자 생성](#)
- [라즈베리파이를 Wi-Fi 네트워크에 연결하세요.](#)
- [라즈베리파이에 원격으로 연결](#)
- [라즈베리파이 카메라 설정하기](#)
- [소프트웨어 사전 요구 사항 설치](#)
- [Kinesis Video Streams C++ 프로듀서 SDK를 다운로드하고 빌드하십시오.](#)
- [Kinesis 비디오 스트림으로 비디오를 스트리밍하고 라이브 스트림을 보십시오.](#)

사전 조건

Raspberry Pi에서 C++ 생산자 SDK를 설치하기 전에 다음 사전 조건을 갖춰야 합니다.

- 다음과 같이 구성된 Raspberry Pi 디바이스:
 - 보드 버전: 3 모델 B 이상
 - 연결된 카메라 모듈
 - 8GB 이상의 용량이 남아 있는 SD 카드
 - 설치된 Raspbian 운영 체제(커널 버전 4.9 이상) [라즈베리 파이 웹 사이트에서 최신 라즈베리 파이 OS \(이전에는 Raspbian이라고 함\) 이미지를 다운로드할 수 있습니다.](#) Raspberry Pi 지침에 따라 [다운로드한 이미지를 SD 카드에 설치](#)합니다.
- 그리고 AWS 계정 Kinesis 비디오 스트림도 사용할 수 있습니다. 자세한 내용은 [Kinesis 비디오 스트림 시작하기](#)를 참조하십시오.

Note

C++ Producer SDK는 기본적으로 미국 서부 (오레곤) (us-west-2) 지역을 사용합니다. 기본값을 사용하려면 미국 서부 (오레곤) 지역에서 Kinesis 비디오 스트림을 AWS 리전 생성하십시오.

Kinesis 비디오 스트림에 다른 지역을 사용하려면 다음 중 하나를 수행하십시오.

- 다음 환경 변수를 리전에 설정합니다(예: *us-east-1*).

```
export AWS_DEFAULT_REGION=us-east-1
```

Kinesis Video Streams에 쓰기 권한을 가진 IAM 사용자 생성

아직 설정하지 않았다면 Kinesis 비디오 스트림에 쓸 수 있는 권한을 가진 AWS Identity and Access Management (IAM) 사용자를 설정하십시오.

이 절차는 AWS 액세스 키 쌍을 사용하여 빠르게 시작할 수 있도록 돕기 위한 것입니다. 장치는 X.509 인증서를 사용하여 연결할 수 있습니다. AWS IoT인증서 기반 인증을 사용하도록 디바이스를 구성하는 방법에 [the section called “를 사용하여 Kinesis Video Streams 리소스에 대한 액세스 제어 AWS IoT”](#) 대한 자세한 내용은 을 참조하십시오.

1. <https://console.aws.amazon.com/iam/> 에서 AWS Management Console 로그인하고 IAM 콘솔을 엽니다.

2. 왼쪽 탐색 메뉴에서 사용자를 클릭합니다.
3. 새 사용자를 생성하려면 사용자 추가를 선택합니다.
4. 사용자에게 설명적 사용자 이름을 지정합니다(예: **kinesis-video-raspberry-pi-producer**).
5. 액세스 유형에서 프로그래밍 방식 액세스를 선택합니다.
6. 다음: 권한을 선택합니다.
7. kinesis-video-raspberry-pi-producer에 대한 권한 설정에서 기존 정책 직접 연결을 선택합니다.
8. 정책 생성을 선택합니다. 새 웹 브라우저 탭에서 정책 생성 페이지가 열립니다.
9. JSON 탭을 선택합니다.
10. 다음 JSON 정책을 복사해 텍스트 영역에 붙여 넣습니다. 이 정책은 사용자에게 Kinesis 비디오 스트림에 데이터를 생성하고 쓸 수 있는 권한을 부여합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "kinesisvideo:DescribeStream",
      "kinesisvideo:CreateStream",
      "kinesisvideo:GetDataEndpoint",
      "kinesisvideo:PutMedia"
    ],
    "Resource": [
      "*"
    ]
  }]
}
```

11. 정책 검토를 선택합니다.
12. 정책에 이름을 지정합니다(예: **kinesis-video-stream-write-policy**).
13. 정책 생성을 선택합니다.
14. 브라우저에서 사용자 추가 탭으로 돌아가서 새로 고침을 선택합니다.
15. 검색 상자에 생성한 정책 이름을 입력합니다.
16. 목록에서 새 정책 옆의 확인란을 선택합니다.
17. 다음: 검토를 선택합니다.
18. 사용자 생성을 선택합니다.

19. 콘솔에 새 사용자의 액세스 키 ID가 표시됩니다. 표시를 선택해 보안 액세스 키를 표시합니다. 애플리케이션을 구성할 때 필요하므로 이러한 값을 기록해 둡니다.

라즈베리파이를 Wi-Fi 네트워크에 연결하세요.

헤드리스 모드에서 즉, 연결된 키보드, 모니터 또는 네트워크 케이블 없이 Raspberry Pi를 사용할 수 있습니다. 연결된 모니터와 키보드를 사용하는 경우에는 [라즈베리파이 카메라 설정하기](#).

1. 컴퓨터에서 `wpa_supplicant.conf`라는 이름의 파일을 하나 만듭니다.
2. 다음 텍스트를 복사하여 `wpa_supplicant.conf` 파일에 붙여넣습니다.

```
country=US
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

network={
  ssid="Your Wi-Fi SSID"
  scan_ssid=1
  key_mgmt=WPA-PSK
  psk="Your Wi-Fi Password"
}
```

Wi-Fi 네트워크에 대한 정보로 `ssid` 및 `psk` 값을 바꿉니다.

3. `wpa_supplicant.conf` 파일을 SD 카드로 복사합니다. boot 볼륨의 루트에 복사해야 합니다.
4. SD 카드를 Raspberry Pi에 삽입하고 디바이스의 전원을 켭니다. 그러면 디바이스가 Wi-Fi 네트워크에 연결되고 SSH가 활성화됩니다.

라즈베리파이에 원격으로 연결

헤드리스 모드에서 Raspberry Pi에 원격으로 연결할 수 있습니다. 모니터와 키보드를 연결한 상태에서 Raspberry Pi를 사용하는 경우 [라즈베리파이 카메라 설정하기](#).

1. Raspberry Pi 디바이스에 원격으로 연결하기 전에 다음 중 하나를 수행하여 이 디바이스의 IP 주소를 확인합니다.
 - 네트워크의 Wi-Fi 라우터에 액세스할 수 있는 경우 연결된 Wi-Fi 디바이스를 찾습니다. Raspberry Pi라는 디바이스를 찾아 디바이스의 IP 주소를 확인합니다.

- 네트워크의 Wi-Fi 라우터에 액세스할 수 없는 경우에는 다른 소프트웨어를 사용하여 네트워크의 디바이스를 찾을 수 있습니다. [Fing](#)은 Android 및 iOS 디바이스에서 사용할 수 있는 널리 사용되는 애플리케이션입니다. 이 애플리케이션의 무료 버전을 사용하여 네트워크에 있는 디바이스의 IP 주소를 찾을 수 있습니다.
2. Raspberry Pi 디바이스의 IP 주소를 알고 있는 경우에는 임의의 터미널 애플리케이션을 사용해 연결할 수 있습니다.
 - macOS 또는 Linux에서는 ssh를 사용합니다.

```
ssh pi@<IP address>
```

- Windows에서는 Windows용 무료 SSH 클라이언트인 [PuTTY](#)를 사용합니다.

Raspbian을 새로 설치할 경우 사용자 이름은 **pi**이고 암호는 **raspberry**입니다. [기본 암호를 변경하는 것이 좋습니다.](#)

라즈베리파이 카메라 설정하기

다음 단계에 따라 디바이스에서 Kinesis 비디오 스트림으로 비디오를 전송하도록 Raspberry Pi 카메라를 구성합니다.

1. 편집기를 열고 다음 명령을 사용해 modules 파일을 업데이트합니다.

```
sudo nano /etc/modules
```

2. 없는 경우 이 파일의 끝에 다음 행을 추가합니다.

```
bcm2835-v4l2
```

3. 파일을 저장하고 편집기를 종료합니다(Ctrl-X).
4. Raspberry Pi를 재부팅합니다.

```
sudo reboot
```

5. 원격으로 연결되어 있는 경우 디바이스가 재부팅되면 터미널 애플리케이션을 통해 다시 연결합니다.
6. Open raspi-config:

```
sudo raspi-config
```

7. 인터페이스 옵션, 레거시 카메라를 선택합니다. 이전 Raspbian 운영 체제 빌드에서는 이 메뉴 옵션이 인터페이스 옵션, 카메라 아래에 있을 수 있습니다.

아직 활성화되지 않은 경우 카메라를 활성화하고 메시지가 표시되면 재부팅합니다.

8. 다음 명령을 입력하여 카메라가 작동하는지 확인합니다.

```
raspistill -v -o test.jpg
```

카메라가 올바르게 구성된 경우 이 명령은 카메라에서 이미지를 캡처하여 이름이 지정된 파일에 저장하고 정보 test.jpg 메시지를 표시합니다.

소프트웨어 사전 요구 사항 설치

C++ 생산자 SDK를 사용하려면 Raspberry Pi에 다음 소프트웨어 사전 조건을 설치해야 합니다.

1. 패키지 목록을 업데이트하고 SDK를 빌드하는 데 필요한 라이브러리를 설치합니다. 다음 명령을 입력합니다.

```
sudo apt update
sudo apt install -y \
  automake \
  build-essential \
  cmake \
  git \
  gstreamer1.0-plugins-base-apps \
  gstreamer1.0-plugins-bad \
  gstreamer1.0-plugins-good \
  gstreamer1.0-plugins-ugly \
  gstreamer1.0-tools \
  gstreamer1.0-omx-generic \
  libcurl4-openssl-dev \
  libgstreamer1.0-dev \
  libgstreamer-plugins-base1.0-dev \
  liblog4cplus-dev \
  libssl-dev \
  pkg-config
```

2. 다음 PEM 파일을 /etc/ssl/cert.pem에 복사합니다.

```
sudo curl https://www.amazontrust.com/repository/AmazonRootCA1.pem -o /etc/ssl/
AmazonRootCA1.pem
sudo chmod 644 /etc/ssl/AmazonRootCA1.pem
```

Kinesis Video Streams C++ 프로듀서 SDK를 다운로드하고 빌드하십시오.

다음 절차를 사용하여 Kinesis Video Streams C++ 프로듀서 SDK를 다운로드하고 빌드할 수 있습니다. 이 방법을 사용할 경우 네트워크 연결 및 프로세서 속도에 따라 빌드하는 데 오랜 시간이 걸릴 수 있습니다.

1. SDK를 다운로드합니다. 유형:

```
git clone https://github.com/aws-labs/amazon-kinesis-video-streams-producer-sdk-
cpp.git
```

2. 빌드 디렉터리를 준비하십시오. 유형:

```
mkdir -p amazon-kinesis-video-streams-producer-sdk-cpp/build
cd amazon-kinesis-video-streams-producer-sdk-cpp/build
```

3. SDK 및 샘플 애플리케이션을 빌드하세요. 빌드하는 Raspberry Pi 모델에 따라 처음 실행하는 데 몇 시간이 걸릴 수 있습니다.

```
cmake .. -DBUILD_GSTREAMER_PLUGIN=ON -DBUILD_DEPENDENCIES=FALSE
make
```

Kinesis 비디오 스트림으로 비디오를 스트리밍하고 라이브 스트림을 보십시오.

1. 샘플 애플리케이션을 실행하려면 다음 정보가 필요합니다.
 - [사전 조건](#) 단원에서 생성한 스트림 이름입니다.
 - [Kinesis Video Streams에 쓰기 권한을 가진 IAM 사용자 생성](#)에서 생성한 계정 자격 증명(액세스 키 ID 및 보안 액세스 키)입니다.
2. 다음 명령을 사용하여 샘플 애플리케이션을 실행합니다. 플레이스홀더를 사용자 환경에 맞는 값으로 바꾸십시오.

```
export GST_PLUGIN_PATH=Directory Where You Cloned the SDK/amazon-kinesis-video-streams-producer-sdk-cpp/build
export AWS_DEFAULT_REGION=AWS Region i.e. us-east-1
export AWS_ACCESS_KEY_ID=Access Key ID
export AWS_SECRET_ACCESS_KEY=Secret Access Key
./kvs_gstreamer_sample Your Stream Name
```

3. 샘플 애플리케이션이 library not found 오류와 함께 종료되는 경우 다음 명령을 입력하여 프로젝트가 오픈 소스 종속 항목에 올바르게 연결되었는지 확인하세요.

```
gst-inspect-1.0 kvssink
```

4. [Kinesis Video Streams](#) 콘솔을 엽니다.
5. 생성한 스트림의 스트림 이름을 선택합니다.

Raspberry Pi에서 전송된 비디오 스트림이 콘솔에 나타납니다.

스트림이 재생 중일 때 Kinesis Video Streams 콘솔의 다음 기능을 시험해 볼 수 있습니다.

- Video preview(비디오 미리 보기) 섹션에서 탐색 컨트롤을 사용하여 스트림을 되감거나 앞으로 빨리 재생합니다.
- Stream info(스트림 정보) 섹션에서 스트림의 코덱, 해상도, 비트레이트를 확인합니다. Raspberry Pi에서는 이 자습서를 진행하기 위한 대역폭 사용량을 최소화하기 위해 해상도와 비트 전송률 값을 의도적으로 낮게 설정합니다. 스트림에 대해 생성 중인 Amazon CloudWatch 지표를 보려면 에서 스트림 지표 보기를 선택합니다 CloudWatch.
- 데이터 보존 기간에서는 비디오 스트림이 1일 동안 보존된다고 표시되어 있습니다. 이 값을 편집해 No data retention(데이터 보존 안 함)으로 설정하거나 이 값을 1일~며칠로 설정할 수 있습니다.

서버 측 암호화에서는 AWS Key Management Service ()AWS KMS에서 유지 관리하는 키를 사용하여 데이터가 저장 중에 암호화되고 있음을 알 수 있습니다.

생산자 SDK 참조

이 단원에는 [Kinesis 비디오 스트림 프로듀서 라이브러리](#)에 대한 제한 사항, 오류 코드 및 기타 참조 정보가 들어 있습니다.

주제

- [프로듀서 SDK 한도](#)
- [오류 코드 참조](#)
- [NAL\(Network Abstraction Layer\) 적응 플래그 참조](#)
- [프로듀서 SDK 구조](#)
- [Kinesis 비디오 스트림 구조](#)
- [프로듀서 SDK 콜백](#)

프로듀서 SDK 한도

다음 표에는 [생산자 라이브러리](#)의 값에 대한 최신 제한 사항이 나와 있습니다.

Note

이러한 값을 설정하기 전에 입력을 확인해야 합니다. SDK는 이러한 제한을 확인하지 않으며 제한이 초과되면 실행 시간 오류가 발생합니다.

값	Limit	참고
최대 스트림 개수	128	생산자 객체에서 생성할 수 있는 최대 스트림 개수입니다. 이 제한은 소프트 제한으로, 제한을 늘리도록 요청할 수 있습니다. 이를 통해 제작자가 실수로 스트림을 재귀적으로 생성하지 않도록 할 수 있습니다.
디바이스의 최대 이름 길이	128자	
최대 태그 개수	스트림당 50개	
최대 스트림 이름 길이	256자	
최소 스토리지 크기	10MiB = 10 * 1024 * 1024바이트	

값	Limit	참고
최대 스토리지 크기	10 GiB = 10 * 1024 * 1024 * 1024바이트	
최대 루트 디렉터리 경로 길이	4,096자	
최대 인증 정보 길이	10,000바이트	
최대 URI 문자열 길이	10,000자	
최대 태그 이름 길이	128자	
최대 태그 값 길이	1,024자	
최소 보안 토큰 기간	30초	
보안 토큰 유예 기간	40분	지정된 지속 시간이 더 길면 이 값으로 제한됩니다.
보존 기간	0 또는 한 시간 이상	0은 보관하지 않음을 나타냅니다.
최소 클러스터 기간	1초	이 값은 SDK 표준인 100ns 단위로 지정됩니다.
최대 클러스터 기간	30초	이 값은 SDK 표준인 100ns 단위로 지정됩니다. 백엔드 API는 클러스터 기간을 더 짧게 적용할 수 있습니다.
최대 조각 크기	50MB	자세한 정보는 Kinesis Video Streams 서비스 할당량 을 참조하세요.
최대 조각 기간	20초	자세한 정보는 Kinesis Video Streams 서비스 할당량 을 참조하세요.

값	Limit	참고
최대 연결 기간	45분	이 시간이 경과하면 백엔드가 연결을 닫습니다. SDK는 이 시간 이내에 토큰을 교체하고 새 연결을 설정합니다.
최대 ACK 세그먼트 길이	1,024자	ACK 구문 분석기 함수로 전송된 승인의 최대 세그먼트 길이입니다.
최대 콘텐츠 유형 문자열 길이	128자	
최대 코덱 ID 문자열 길이	32자	
최대 추적 이름 문자열 길이	32자	
최대 코덱 프라이빗 데이터 길이	1MiB = 1 * 1024 * 1024바이트	
최소 타임코드 스케일 값 길이	100ns	결과 MKV 클러스터에서 프레임 타임스탬프를 나타내는 최소 타임코드 스케일 값입니다. 이 값은 SDK 표준인 100ns 단위씩 증분되어 지정됩니다.
최대 타임코드 스케일 값 길이	1초	결과 MKV 클러스터에서 프레임 타임스탬프를 나타내는 최대 타임코드 스케일 값입니다. 이 값은 SDK 표준인 100ns 단위씩 증분되어 지정됩니다.
최소 콘텐츠 보기 항목 수	10	
최소 버퍼 기간	20초	이 값은 SDK 표준인 100ns 단위씩 증분되어 지정됩니다.
최대 업데이트 버전 길이	128자	

값	Limit	참고
최대 ARN 길이	1024자	
최대 조각 시퀀스 길이	128자	
최대 보존 기간	10년	

오류 코드 참조

이 섹션에는 [생산자 라이브러리](#)에 대한 오류 및 상태 코드 정보가 포함되어 있습니다.

공통 문제 해결 방법에 대한 자세한 내용은 [Kinesis Video Streams 문제 해결](#) 단원을 참조하십시오.

주제

- [PutFrame 콜백에서 반환된 오류 및 상태 코드 - 플랫폼 독립 코드 \(PIC\)](#)
- [PutFrame 콜백에서 반환된 오류 및 상태 코드 - C 프로듀서 라이브러리](#)

PutFrame 콜백에서 반환된 오류 및 상태 코드 - 플랫폼 독립 코드 (PIC)

다음 섹션에는 플랫폼 독립 코드 (PIC) 내 PutFrame 작업에 대한 콜백에서 반환되는 오류 및 상태 정보가 포함되어 있습니다.

주제

- [클라이언트 라이브러리에서 반환된 오류 및 상태 코드](#)
- [기간 라이브러리에서 반환된 오류 및 상태 코드](#)
- [공용 라이브러리에서 반환된 오류 및 상태 코드](#)
- [힙 라이브러리에서 반환된 오류 및 상태 코드](#)
- [MKVGen 라이브러리에서 반환된 오류 및 상태 코드](#)
- [Trace 라이브러리에서 반환된 오류 및 상태 코드](#)
- [Utils 라이브러리에서 반환한 오류 및 상태 코드](#)
- [View 라이브러리에서 반환된 오류 및 상태 코드](#)

클라이언트 라이브러리에서 반환된 오류 및 상태 코드

다음 표에는 Kinesis Video Client Streams 라이브러리의 메서드에서 반환되는 오류 및 상태 정보가 나와 있습니다.

코드	메시지	설명	권장 조치
0x52000001	STATUS_MAXIMUM_STREAM_COUNT	최대 스트림 수에 도달함.	프로듀서 SDK 한도 에 지정된 대로 DeviceInfo에서 최대 스트림 개수를 늘려 지정합니다..
0x52000002	STATUS_MIN_STREAM_COUNT	최소 스트림 수 오류.	0인치보다 큰 스트림의 최대 수를 지정하십시오. DeviceInfo
0x52000003	STATUS_INVALID_DEVICE_NAME_LENGTH	잘못된 디바이스 이름 길이.	에 지정된 최대 장치 이름 길이 (문자 수) 를 참조하십시오 프로듀서 SDK 한도 .
0x52000004	STATUS_INVALID_DEVICE_INFO_VERSION	잘못된 DeviceInfo 구조 버전.	구조의 올바른 현재 버전을 지정합니다.
0x52000005	STATUS_MAXIMUM_TAG_COUNT	최대 태그 개수에 도달함.	에 지정된 현재 최대 태그 수를 참조하십시오 프로듀서 SDK 한도 .
0x52000006	STATUS_DEVICE_FINGERPRINT_LENGTH		
0x52000007	STATUS_INVALID_CALLBACKS_VERSION	잘못된 Callbacks 구조 버전.	구조의 올바른 현재 버전을 지정합니다.
0x52000008	STATUS_INVALID_STREAM_INFO_VERSION	잘못된 StreamInfo 구조 버전.	구조의 올바른 현재 버전을 지정합니다.

코드	메시지	설명	권장 조치
0x52000009	STATUS_INVALID_STREAM_NAME_LENGTH	잘못된 스트림 이름 길이.	에 지정된 최대 스트림 이름 길이 (문자 수) 를 참조하십시오 프로듀서 SDK 한도 .
0x5200000a	STATUS_INVALID_STORAGE_SIZE	잘못된 스토리지 크기가 지정됨.	바이트 단위의 스토리지 크기는 프로듀서 SDK 한도 에 지정된 제한을 벗어나면 안 됩니다.
0x5200000b	STATUS_INVALID_ROOT_DIRECTORY_LENGTH	잘못된 루트 디렉터리 문자열 길이.	에 지정된 최대 루트 디렉터리 경로 길이를 참조하십시오 프로듀서 SDK 한도 .
0x5200000c	STATUS_INVALID_SPILL_RATIO	잘못된 유출 비율.	유출 비율을 0—100 사이의 백분율로 표현하십시오.
0x5200000d	STATUS_INVALID_STORAGE_INFO_VERSION	잘못된 StorageInfo 구조 버전.	구조의 올바른 현재 버전을 지정합니다.
0x5200000e	STATUS_INVALID_STREAM_STATE	스트림이 현재 작업을 허용하지 않는 상태입니다.	가장 일반적으로 이 오류는 SDK가 요청된 작업을 수행하는 데 필요한 상태에 도달하지 못할 때 발생합니다. 예를 들어 GetStreamingEndpoint API 호출이 실패했는데 클라이언트 애플리케이션이 이를 무시하고 스트림에 프레임을 계속 넣으려고 할 경우 이 오류가 발생합니다.

코드	메시지	설명	권장 조치
0x5200000f	STATUS_SERVICE_CALLBACKS_MISSING	Callbacks 구조에 일부 필수 함수에 대한 함수 진입점이 누락됨.	필수 콜백이 클라이언트 애플리케이션에 구현되어 있는지 확인하십시오. 이 오류는 플랫폼 독립 코드 (PIC) 클라이언트에만 노출됩니다. C++ 및 기타 더 높은 수준의 래퍼는 이러한 호출을 충족합니다.
0x52000010	STATUS_SERVICE_NOT_AUTHORIZED_ERROR	인증되지 않음.	보안 토큰, 인증서, 보안 토큰 통합 및 만료를 확인하십시오. 토큰에 올바른 관련 권한이 있는지 확인하십시오. Kinesis Video Streams 샘플 애플리케이션의 경우 환경 변수가 올바르게 설정되었는지 확인하십시오.
0x52000011	STATUS_DESCRIBE_STREAM_CALL_FAILED	DescribeStream API 실패.	이 오류는 DescribeStream API 재시도 실패 후 반환됩니다. PIC 클라이언트는 재시도를 중지한 후 이 오류를 반환합니다.
0x52000012	STATUS_INVALID_DESCRIPTOR_RESPONSE	잘못된 DescribeStreamResponse 구조.	DescribeStreamResultEvent에 전달된 구조가 null이거나 null인 Amazon 리소스 이름 (ARN)과 같이 잘못된 항목을 포함하고 있습니다.

코드	메시지	설명	권장 조치
0x52000013	STATUS_STREAM_IS_BEING_DELETED_ERROR	스트림 삭제 중	삭제 중인 스트림으로 인해 API 실패가 발생했습니다. 스트림이 사용 중인 동안 다른 프로세스가 스트림을 삭제하려고 하지 않는지 확인하십시오.
0x52000014	STATUS_SERVICE_CALL_INVALID_ARG_ERROR	서비스 호출에 대해 잘못된 인수가 지정됨.	서비스 호출 인수가 유효하지 않거나 SDK에서 해석할 수 없는 오류가 발생하면 백엔드에서 이 오류를 반환합니다.
0x52000015	STATUS_SERVICE_CALL_DEVICE_NOT_FOUND_ERROR	디바이스를 찾을 수 없음.	사용 중에 기기가 삭제되지 않았는지 확인하세요.
0x52000016	STATUS_SERVICE_CALL_DEVICE_NOT_PROVISIONED_ERROR	디바이스가 프로비저닝되지 않음.	기기가 프로비저닝되었는지 확인하세요.
0x52000017	STATUS_SERVICE_CALL_RESOURCE_NOT_FOUND_ERROR	서비스에서 반환된 일반 리소스가 없음.	이 오류는 서비스에서 리소스(예: 스트림)를 찾을 수 없는 경우 발생합니다. 이 오류는 상황마다 의미하는 바가 다를 수 있지만 스트림을 생성하기 전에 API를 사용한 것이 원인일 수 있습니다. SDK를 사용하면 스트림이 먼저 생성되었는지 확인할 수 있습니다.

코드	메시지	설명	권장 조치
0x52000018	STATUS_INVALID_AUTH_LEN	잘못된 인증 정보 길이.	프로듀서 SDK 한도 에 지정된 현재 값을 참조하십시오.
0x52000019	STATUS_CREATE_STREAM_CALL_FAILED	CreateStream API 호출에 실패했습니다.	작업 실패 이유에 대한 자세한 정보는 오류 문자열을 참조하십시오.
0x5200002a	STATUS_GET_STREAMING_TOKEN_CALL_FAILED	GetStreamingToken 호출 실패.	작업 실패 이유에 대한 자세한 정보는 오류 문자열을 참조하십시오.
0x5200002b	STATUS_GET_STREAMING_ENDPOINT_CALL_FAILED	GetStreamingEndpoint API 호출에 실패했습니다.	작업 실패 이유에 대한 자세한 정보는 오류 문자열을 참조하십시오.
0x5200002c	STATUS_INVALID_URI_LEN	GetStreamingEndpoint API에서 잘못된 URI 문자열 길이가 반환됨.	프로듀서 SDK 한도 에 지정된 현재 최대값을 참조하십시오.
0x5200002d	STATUS_PUT_STREAM_CALL_FAILED	PutMedia API 호출에 실패했습니다.	작업 실패 이유에 대한 자세한 정보는 오류 문자열을 참조하십시오.

코드	메시지	설명	권장 조치
0x5200002e	STATUS_STORE_OUT_OF_MEMORY	콘텐츠 저장소에 메모리가 부족함.	콘텐츠 스토어는 스트림 간에 공유되기 때문에 모든 스트림에 대한 최대 기간을 저장할 수 있는 충분한 공간에 (조각 모음을 처리하기 위한) ~20%의 충분한 용량이 필요합니다. 스토리지를 오버플로우를 방지해야 합니다. 누적된 스토리지 크기 및 지연 시간 허용 오차에 해당하는 스트림당 최대 기간 값을 선택합니다. 프레임이 그냥 놓는 것보다는 콘텐츠 보기 창 밖으로 떨어질 때 삭제하는 것이 좋습니다 (콘텐츠 저장소 메모리 부족). 프레임을 삭제하면 스트림 압력 알림 콜백이 시작되기 때문입니다. 그러면 애플리케이션이 업스트림 미디어 구성 요소(예: 인코더)를 조정하여 비트 전송률 또는 프레임 수를 줄이거나 적절히 작동합니다.

코드	메시지	설명	권장 조치
0x5200002f	STATUS_NO_MORE_DATA_AVAILABLE	현재 스트림에 사용 가능한 데이터가 없음.	미디어 파이프라인이 서비스로 전송하기 위해 네트워크 스레드에서 프레임을 소비하는 것보다 훨씬 더 느리게 생성하는 경우 이는 잠재적으로 유효한 결과입니다. 상위 수준 클라이언트 (예: C++, Java 또는 Android)에서는 내부적으로 처리되므로 이 경고가 표시되지 않습니다.
0x52000030	STATUS_INVALID_TAG_VERSION	잘못된 Tag 구조 버전.	구조의 올바른 현재 버전을 지정합니다.
0x52000031	STATUS_SERVICE_UNKNOWN_ERROR	네트워크 스택에서 알 수 없는 오류 또는 일반 오류가 반환됨.	자세한 내용은 로그를 참조하십시오.
0x52000032	STATUS_SERVICE_RESOURCE_IN_USE_ERROR	리소스 사용 중.	서비스에서 반환됩니다. 자세한 내용은 Kinesis Video Streams API 레퍼런스를 참조하십시오.
0x52000033	STATUS_SERVICE_CLIENT_LIMIT_ERROR	클라이언트 제한.	서비스에서 반환됩니다. 자세한 내용은 Kinesis Video Streams API 레퍼런스를 참조하십시오.
0x52000034	STATUS_SERVICE_DEVICE_LIMIT_ERROR	디바이스 제한.	서비스에서 반환됩니다. 자세한 내용은 Kinesis Video Streams API 레퍼런스를 참조하십시오.

코드	메시지	설명	권장 조치
0x52000035	STATUS_SERVICE_CALL_STREAM_LIMIT_ERROR	스트림 제한.	서비스에서 반환됩니다. 자세한 내용은 Kinesis Video Streams API 레퍼런스를 참조하십시오.
0x52000036	STATUS_SERVICE_CALL_RESOURCE_DELETED_ERROR	리소스가 삭제되었거나 삭제 중임.	서비스에서 반환됩니다. 자세한 내용은 Kinesis Video Streams API 레퍼런스를 참조하십시오.
0x52000037	STATUS_SERVICE_CALL_TIMEOUT_ERROR	서비스 호출 시간 초과.	특정 서비스 API 호출로 인해 시간 초과가 발생했습니다. 네트워크 연결이 올바른지 확인하십시오. PIC에서는 이 작업을 자동으로 다시 시도합니다.
0x52000038	STATUS_STREAM_READ_CALLBACK_FAILED	스트림 준비 상태 알림.	이 알림은 PIC에서 클라이언트로 전송되어 비동기 스트림이 생성되었는지를 나타냅니다.
0x52000039	STATUS_DEVICE_TAGS_COUNT_NON_ZERO_TAGS_NULL	잘못된 태그가 지정됨.	태그 수는 0이 아니지만 태그가 비어 있습니다. 태그가 지정되었는지 또는 개수가 0인지 확인하십시오.
0x5200003a	STATUS_INVALID_STREAM_DESCRIPTION_VERSION	잘못된 StreamDescription 구조 버전.	구조의 올바른 현재 버전을 지정합니다.
0x5200003b	STATUS_INVALID_TAG_NAME_LEN	잘못된 태그 이름 길이.	프로듀서 SDK 한도 에 지정된 태그 이름에 대한 제한 사항을 참조하십시오.

코드	메시지	설명	권장 조치
0x5200003c	STATUS_INVALID_TAG_VALUE_LEN	잘못된 태그 값 길이.	프로듀서 SDK 한도 에 지정된 태그 값에 대한 제한 사항을 참조하십시오.
0x5200003d	STATUS_TAG_STREAM_CALL_FAILED	TagResource API 실패.	TagResource API 호출에 실패했습니다. 유효한 네트워크 연결이 있는지 확인하십시오. 실패에 대한 자세한 내용은 로그를 참조하십시오.
0x5200003e	STATUS_INVALID_CUSTOM_DATA	PIC API를 호출하는 잘못된 사용자 지정 데이터.	PIC API에 대한 호출에 잘못된 사용자 지정 데이터가 지정되었습니다. 이 오류는 PIC를 직접 사용하는 클라이언트에서만 발생할 수 있습니다.
0x5200003f	STATUS_INVALID_CREATE_STREAM_RESPONSE	잘못된 CreateStreamResponse 구조.	구조 또는 멤버 필드가 잘못되었습니다(즉, ARN이 null이거나 프로듀서 SDK 한도 에 지정된 값보다 큼).
0x52000040	STATUS_CLIENT_AUTH_CALL_FAILED	클라이언트 인증 실패.	여러 번 재시도해도 PIC가 적절한 인증 정보 (AccessKeyId 또는 SecretAccessKey)를 가져오지 못했습니다. 인증 통합을 확인하십시오. 샘플 애플리케이션에서는 환경 변수를 사용하여 C++ 생산자 라이브러리로 자격 증명 정보를 전달합니다.

코드	메시지	설명	권장 조치
0x52000041	STATUS_GET_CLIENT_TOKEN_CALL_FAILED	보안 토큰 호출 실패.	이러한 상황은 PIC를 직접 사용하는 클라이언트에서 발생할 수 있습니다. 여러 번의 재시도 후 호출에 실패하고 이 오류가 발생합니다.
0x52000042	STATUS_CLIENT_PROVISION_CALL_FAILED	프로비저닝 오류.	프로비저닝이 구현되지 않았습니다.
0x52000043	STATUS_CREATE_CLIENT_CALL_FAILED	생산자 클라이언트 생성 실패.	클라이언트 생성에 실패한 경우 여러 번의 재시도 후 PIC에서 일반 오류를 반환합니다.
0x52000044	STATUS_CLIENT_READ_CALLBACK_FAILED	생산자 클라이언트 준비 상태 실패.	PIC가 준비 상태로 되지 못한 경우 PIC 상태 시스템에서 반환합니다. 근본 원인에 대한 자세한 내용은 로그를 참조하십시오.
0x52000045	STATUS_TAG_CLIENT_CALL_FAILED	생산자 클라이언트에 대한 TagResource에 실패.	생산자 클라이언트에 대한 TagResource API 호출에 실패했습니다. 근본 원인에 대한 자세한 내용은 로그를 참조하십시오.
0x52000046	STATUS_INVALID_CREATE_DEVICE_RESPONSE	디바이스/생산자 생성 실패.	상위 수준 SDK (예: C++ 또는 Java)는 아직 기기 또는 제작자 생성 API를 구현하지 않습니다. PIC를 사용하는 클라이언트는 결과 알림을 사용해 직접 실패를 나타낼 수 있습니다.

코드	메시지	설명	권장 조치
0x52000047	STATUS_ACK_TIMESTAMP_NOT_IN_VIEW_WINDOW	수신된 ACK의 타임스탬프가 보기에 없음.	수신된 ACK에 해당하는 프레임이 콘텐츠 보기 창을 벗어나는 경우 이 오류가 발생합니다. 일반적으로 이 오류는 ACK 전송이 느린 경우 발생합니다. 이 오류는 다운링크가 느리다는 경고 및 표시로 해석할 수 있습니다.
0x52000048	STATUS_INVALID_FRAGMENT_ACK_VERSION	잘못된 FragmentAck 구조 버전.	FragmentAck 구조의 올바른 현재 버전을 지정합니다.
0x52000049	STATUS_INVALID_TOKEN_EXPIRATION	잘못된 보안 토큰 만료.	보안 토큰 만료는 미래에는 현재 타임스탬프보다 큰 절대 타임스탬프를 가져야 하며, 유예 기간이 적용됩니다. 유예 기간에 대한 제한 사항은 프로듀서 SDK 한도 단원을 참조하십시오.
0x5200004a	STATUS_END_OF_STREAM	EOS(스트림 끝) 표시기.	GetStreamData API 호출에서 현재 업로드 처리 세션이 종료되었음을 나타냅니다. 세션이 종료되었으나 오류가 발생한 경우 또는 세션 토큰이 만료되어 세션이 교체 중인 경우 이 오류가 발생합니다.

코드	메시지	설명	권장 조치
0x5200004b	STATUS_DUPLICATE_STREAM_NAME	중복 스트림 이름.	여러 스트림이 동일한 스트림 이름을 가질 수 없습니다. 스트림에 대해 고유한 이름을 선택하십시오.
0x5200004c	STATUS_INVALID_RETENTION_PERIOD	잘못된 보존 기간.	StreamInfo 구조에 잘못된 보존 기간이 지정되어 있습니다. 보존 기간에 유효한 값의 범위에 대한 자세한 내용은 프로듀서 SDK 한도 단원을 참조하십시오.
0x5200004d	STATUS_INVALID_ACK_KEY_START	잘못된 FragmentAck .	조각 ACK 문자열을 구문 분석하지 못했습니다. 잘못된 키 시작 표시기입니다. 조각 ACK 문자열이 손상되었을 수 있습니다. 자동 수정이 가능하므로 이 오류는 경고로 처리될 수 있습니다.
0x5200004e	STATUS_INVALID_ACK_DUPLICATE_KEY_NAME	잘못된 FragmentAck .	조각 ACK 문자열을 구문 분석하지 못했습니다. 여러 키의 이름이 동일합니다. 조각 ACK 문자열이 손상되었을 수 있습니다. 자동 수정이 가능하므로 이 오류는 경고로 처리될 수 있습니다.

코드	메시지	설명	권장 조치
0x5200004f	STATUS_INVALID_ACK_INVALID_VALUE_START	잘못된 FragmentAck .	키 값 시작 표시기가 잘못되어 조각 ACK 문자열을 구문 분석하지 못했습니다. 조각 ACK 문자열이 손상되었을 수 있습니다. 자동 수정이 가능하므로 이 오류는 경고로 처리될 수 있습니다.
0x52000050	STATUS_INVALID_ACK_INVALID_VALUE_END	잘못된 FragmentAck .	키 값 종료 표시기가 잘못되어 조각 ACK 문자열을 구문 분석하지 못했습니다. 조각 ACK 문자열이 손상되었을 수 있습니다. 자동 수정이 가능하므로 이 오류는 경고로 처리될 수 있습니다.
0x52000051	STATUS_INVALID_PARSED_ACK_TYPE	잘못된 FragmentAck .	ACK 유형이 잘못 지정되어 조각 ACK 문자열을 구문 분석하지 못했습니다.
0x52000052	STATUS_STREAM_HAS_BEEN_STOPPED	스트림이 중지됨.	스트림이 중지되었으나 프레임이 스트림에 계속 들어가고 있습니다.
0x52000053	STATUS_INVALID_STREAM_METRICS_VERSION	잘못된 StreamMetrics 구조 버전.	StreamMetrics 구조의 올바른 현재 버전을 지정합니다.
0x52000054	STATUS_INVALID_CLIENT_METRICS_VERSION	잘못된 ClientMetrics 구조 버전.	ClientMetrics 구조의 올바른 현재 버전을 지정합니다.

코드	메시지	설명	권장 조치
0x52000055	STATUS_INVALID_CLIENT_READY_STATE	생산자 초기화가 준비 상태에 도달하지 못함.	생산자 클라이언트 초기화 중 준비 상태에 도달하지 못했습니다. 자세한 내용은 로그를 참조하십시오.
0x52000056	STATUS_STATE_MACHINE_STATE_NOT_FOUND	내부 상태 시스템 오류.	공개적으로 표시되는 오류가 아닙니다.
0x52000057	STATUS_INVALID_FRAGMENT_ACK_TYPE	잘못된 ACK 유형이 FragmentAck 구조에 지정됨.	FragmentAck 구조에는 퍼블릭 헤더에 정의된 ACK 유형이 포함되어 있어야 합니다.
0x52000058	STATUS_INVALID_STREAM_READY_STATE	내부 상태 시스템 전환 오류.	공개적으로 표시되는 오류가 아닙니다.
0x52000059	STATUS_CLIENT_FREE_BEFORE_STREAM	생산자를 비운 후 스트림 객체를 비움.	생산자 객체를 비운 후 스트림 객체를 비우려는 시도가 있었습니다. 이 오류는 PIC를 직접 사용하는 클라이언트에서만 발생할 수 있습니다.
0x5200005a	STATUS_ALLOCATION_SIZE_SMALLER_THAN_REQUESTED	내부 스토리지 오류.	콘텐츠 저장소의 실제 할당 크기가 패키징된 프레임 및 프래그먼트의 크기보다 작다는 내부 오류가 발생했습니다.

코드	메시지	설명	권장 조치
0x5200005b	STATUS_VI EW_ITEM_S IZE_GREAT ER_THAN_A LLOCATION	내부 스토리지 오류.	콘텐츠 보기에서 할당에 저장된 크기가 콘텐츠 스토어의 할당 크기보다 큼니다.
0x5200005c	STATUS_AC K_ERR_STR EAM_READ_ERROR	스트림 읽기 오류 ACK.	ACK가 백엔드에서 반환한 오류로, 이는 스트림 읽기 또는 구문 분석 오류를 나타냅니다. 이 오류는 일반적으로 백엔드에서 스트림을 검색하지 못한 경우 발생합니다. 일반적으로 자동 다시 스트리밍이 수행되면 이 오류가 해결될 수 있습니다.
0x5200005d	STATUS_AC K_ERR_FRA GMENT_SIZ E_REACHED	최대 조각 크기에 도달함.	바이트 단위의 최대 조각 크기는 프로듀서 SDK 한도 에 정의되어 있습니다. 이 오류는 매우 큰 프레임이 있거나 관리 가능한 크기 조각을 생성하는 키 프레임이 없음을 나타냅니다. 인코더 설정을 확인하고 키 프레임이 제대로 생성되고 있는지 확인합니다. 밀도가 매우 높은 스트림의 경우 최대 크기를 관리하려면 보다 짧은 기간에 조각을 생성하도록 인코더를 구성합니다.

코드	메시지	설명	권장 조치
0x5200005e	STATUS_ACK_ERR_FRAGMENT_DURATION_REACHED	최대 조각 기간에 도달함.	최대 조각 기간은 프로듀서 SDK 한도 에 정의되어 있습니다. 이 오류는 초당 프레임 속도가 매우 낮거나 관리 가능한 기간 조각을 생성하는 키 프레임이 없음을 나타냅니다. 인코더 설정을 확인하고 키프레임이 일정한 간격으로 제대로 생성되고 있는지 확인하십시오.
0x5200005f	STATUS_ACK_ERR_CONNECTION_DURATION_REACHED	최대 연결 기간에 도달함.	Kinesis Video Streams는 에 지정된 최대 연결 기간을 적용합니다. 프로듀서 SDK 한도 Producer SDK는 최대 값에 도달하기 전에 스트림 또는 토큰을 자동으로 회전시킵니다. SDK를 사용하는 클라이언트는 이 오류를 수신하지 않아야 합니다.
0x52000060	STATUS_ACK_ERR_FRAGMENT_TIMESTAMP_NOT_MONOTONIC	타임코드가 단조 증가하지 않음.	Producer SDK는 타임스탬프를 적용하므로 SDK를 사용하는 클라이언트는 이 오류를 수신하지 않아야 합니다.
0x52000061	STATUS_ACK_ERR_MULTITRACK_MKV	MKV에 추적이 여러 개 있음.	Producer SDK는 단일 트랙 스트림을 적용하므로 SDK를 사용하는 클라이언트는 이 오류를 수신하지 않아야 합니다.

코드	메시지	설명	권장 조치
0x52000062	STATUS_ACK_ERR_INV_ALID_MKV_DATA	MKV 데이터가 잘못 됨.	백엔드 MKV 구문 분석기에서 스트림을 구문 분석하는 중 오류가 발생했습니다. 전환 과정에서 스트림이 손상되면 SDK를 사용하는 클라이언트에서 이 오류가 발생할 수 있습니다. 이는 버퍼 압력으로 인해 SDK가 부분적으로 전송된 테일 프레임을 강제로 삭제해야 하는 경우에도 발생할 수 있습니다. 후자의 경우 FPS와 해상도를 낮추거나 압축률을 높이거나, 네트워크가 “버스트”인 경우 일시적인 압력을 감당할 수 있도록 더 큰 콘텐츠 저장소와 버퍼 지속 시간을 허용하는 것이 좋습니다.

코드	메시지	설명	권장 조치
0x52000063	STATUS_ACK_ERR_INVALID_PRODUCER_TIMESTAMP	생산자 타임스탬프가 잘못됨.	생산자 클록에 미래에 대해 큰 드리프트가 발생하면 서비스에서 이러한 오류 ACK를 반환합니다. 더 높은 수준의 SDK(예: Java 또는 C++)에서는 몇 가지 시스템 클록 버전을 사용하여 PIC의 현재 시간 콜백을 충족합니다. 시스템 클럭이 제대로 설정되었는지 확인하십시오. PIC를 직접 사용하는 클라이언트는 콜백 함수가 올바른 타임스탬프를 반환하는지 확인해야 합니다.
0x52000064	STATUS_ACK_ERR_STREAM_NOT_ACTIVE	비활성 스트림.	스트림이 "활성" 상태가 아닌데 백엔드 API가 호출되었습니다. 클라이언트가 스트림을 생성한 후 즉시 해당 스트림에 프레임을 집어 넣으려고 하면 이 오류가 발생합니다. SDK에서는 상태 시스템 및 복구 시스템을 통해 이러한 시나리오를 처리합니다.
0x52000065	STATUS_ACK_ERR_KMS_KEY_ACCESS_DENIED	AWS KMS 액세스 거부 오류입니다.	계정에 지정된 키에 대한 액세스 권한이 없는 경우 반환됩니다.
0x52000066	STATUS_ACK_ERR_KMS_KEY_DISABLED	AWS KMS 키가 비활성화되었습니다.	지정된 키가 비활성화되었습니다.

코드	메시지	설명	권장 조치
0x52000067	STATUS_ACK_ERR_KMS_KEY_VALIDATION_ERROR	AWS KMS 키 검증 오류가 발생했습니다.	일반적인 확인 오류입니다. 자세한 내용은 AWS Key Management Service API 참조 를 참조하세요.
0x52000068	STATUS_ACK_ERR_KMS_KEY_UNAVAILABLE	AWS KMS key 사용할 수 없습니다.	이 키를 사용할 수 없습니다. 자세한 내용은 AWS Key Management Service API 참조 를 참조하세요.
0x52000069	STATUS_ACK_ERR_KMS_KEY_INVALID_USAGE	KMS 키 사용이 잘못되었습니다.	AWS KMS key 는 이 컨텍스트에서 사용하도록 구성되지 않았습니다. 자세한 내용은 AWS Key Management Service API 참조 를 참조하세요.
0x5200006a	STATUS_ACK_ERR_KMS_KEY_INVALID_STATE	AWS KMS 잘못된 상태입니다.	자세한 내용은 AWS Key Management Service API 참조 를 참조하세요.
0x5200006b	STATUS_ACK_ERR_KMS_KEY_NOT_FOUND	KMS 키를 찾을 수 없습니다.	이 키를 찾을 수 없습니다. 자세한 내용은 AWS Key Management Service API 참조 를 참조하세요.
0x5200006c	STATUS_ACK_ERR_STREAM_DELETED	스트림이 삭제되었거나 삭제 중임.	다른 애플리케이션 또는 AWS Management Console을 통해 스트림이 삭제 중입니다.
0x5200006d	STATUS_ACK_ERR_INTERNAL_ERROR	내부 오류.	일반적인 서비스 내부 오류

코드	메시지	설명	권장 조치
0x5200006e	STATUS_ACK_ERR_FRAGMENT_ARC_HIVAL_ERROR	조각 보관 오류.	서비스에서 조각을 지속적으로 유지하고 인덱싱하지 못한 경우 반환됩니다. 드문 경우이긴 하지만 이 오류는 다양한 이유로 발생할 수 있습니다. 기본적으로 SDK에서는 조각 전송을 다시 시도합니다.
0x5200006f	STATUS_ACK_ERR_UNKNOWN_ACK_ERROR	알 수 없는 오류.	서비스에서 알 수 없는 오류를 반환했습니다.
0x52000070	STATUS_MISSED_ERR_ACK_ID	ACK 정보 누락.	ACK 구문 분석기가 구문 분석을 완료했지만 FragmentAck 정보가 누락되었습니다.
0x52000071	STATUS_INVALID_ACK_SEGMENT_LEN	잘못된 ACK 세그먼트 길이.	ACK 구문 분석기에 대해 길이가 잘못된 ACK 세그먼트 문자열이 지정되었습니다. 자세한 정보는 프로듀서 SDK 한도 를 참조하세요.

코드	메시지	설명	권장 조치
0x52000074	STATUS_MAX_FRAGMENTS_METADATA_COUNT	메타데이터 항목의 최대 수가 조각에 추가되었습니다.	Kinesis 비디오 스트림은 비지속적 항목을 프래그먼트에 추가하거나 지속적 항목을 메타데이터 대기열에 추가하여 프래그먼트에 최대 10개의 메타데이터 항목을 추가할 수 있습니다. 자세한 정보는 Kinesis Video Streams에서 스트리밍 메타데이터 사용 을 참조하세요.
0x52000075	STATUS_ACK_ERR_FRAGMENT_METADATA_LIMIT_REACHED	한도(최대 메타데이터 개수, 메타데이터 이름 길이 또는 메타데이터 값 길이)에 도달했습니다.	생산자 SDK는 메타데이터 항목의 개수와 크기를 제한합니다. Producer SDK 코드의 한도를 변경하지 않는 한 이 오류는 발생하지 않습니다. 자세한 정보는 Kinesis Video Streams에서 스트리밍 메타데이터 사용 을 참조하세요.
0x52000076	STATUS_BLOCKING_PUT_INTERRUPTED_STREAM_TERMINATED	구현되지 않음.	

코드	메시지	설명	권장 조치
0x52000077	STATUS_INVALID_METADATA_NAME	메타데이터 이름이 잘못되었습니다.	메타데이터 이름은 문자열 "AWS"로 시작할 수 없습니다. 이 오류가 발생하면 메타데이터 항목이 프래그먼트 또는 메타데이터 대기열에 추가되지 않습니다. 자세한 정보는 Kinesis Video Streams에서 스트리밍 메타데이터 사용 을 참조하세요.
0x52000078	STATUS_END_OF_FRAGMENT_FRAME_INVALID_STATE	한 조각 프레임의 끝 부분이 잘못된 상태에 있습니다.	프래그먼트의 끝을 프래그먼트된 스트림으로 non-key-frame 전송해서는 안 됩니다.
0x52000079	STATUS_TRACK_INFO_MISSING	트랙 정보가 누락되었습니다.	트랙 번호는 0보다 커야 하며 트랙 ID와 일치해야 합니다.
0x5200007a	STATUS_MAXIMUM_TRACK_COUNT_EXCEEDED	최대 트랙 수를 초과했습니다.	스트림당 최대 3개의 트랙을 포함할 수 있습니다.
0x5200007b	STATUS_OFFLINE_MODE_WITH_ZERO_RETENTION	오프라인 스트리밍 모드 보관 기간이 0으로 설정되어 있습니다.	오프라인 스트리밍 모드 유지 시간을 0으로 설정해서는 안 됩니다.
0x5200007c	STATUS_ACK_TRACK_NUMBER_MISMATCH	오류 ACK의 트랙 번호가 일치하지 않습니다.	

코드	메시지	설명	권장 조치
0x5200007d	STATUS_ACK_ERR_FRAMES_MISSING_FOR_TRACK	트랙에 대한 프레임이 누락되었습니다.	
0x5200007e	STATUS_ACK_ERR_MORE_THAN_ALLOWED_TRACKS_FOUND	허용되는 최대 트랙 수를 초과했습니다.	
0x5200007f	STATUS_UPLOAD_HANDLE_ABORTED	업로드 처리가 중단됩니다.	
0x52000080	STATUS_INVALID_CERT_PATH_LENGTH	잘못된 인증서 경로 길이입니다.	
0x52000081	STATUS_DUPLICATE_TRACK_ID_FOUND	중복된 트랙 ID가 확인되었습니다.	
0x52000082	STATUS_INVALID_CLIENT_INFO_VERSION		
0x52000083	STATUS_INVALID_CLIENT_ID_STRING_LENGTH		

코드	메시지	설명	권장 조치
0x52000084	STATUS_SETTING_KEY_FRAME_FLAG_WHILE_USING_EOFR		
0x52000085	STATUS_MAXIMUM_FRAME_TIMESTAMP_DELTA_BETWEEN_TRACKS_EXCEEDED		
0x52000086	STATUS_STREAM_SHUTTING_DOWN		
0x52000087	STATUS_CLIENT_SHUTTING_DOWN		
0x52000088	STATUS_PUTMEDIA_LAST_PERSISTENT_ACK_NOT_RECEIVED		
0x52000089	STATUS_NON_ALIGNED_HEAP_WITHIN_CONTENT_STORE_ALLOCATORS		

코드	메시지	설명	권장 조치
0x5200008a	STATUS_MULTIPLE_CONSECUTIVE_EOFR		
0x5200008b	STATUS_DUPLICATE_STREAM_EVENT_TYPE		
0x5200008c	STATUS_STREAM_NOT_STARTED		
0x5200008d	STATUS_INVALID_IMAGE_PREFIX_LENGTH		
0x5200008e	STATUS_INVALID_METADATA_KEY_LENGTH		
0x5200008f	STATUS_INVALID_METADATA_VALUE_LENGTH		

기간 라이브러리에서 반환된 오류 및 상태 코드

다음 표에는 Duration 라이브러리의 메서드에서 반환되는 오류 및 상태 정보가 나와 있습니다.

코드	메시지
0xFFFFFFFFFFFFFFFF	INVALID_DURATION_VALUE

공용 라이브러리에서 반환된 오류 및 상태 코드

다음 표에는 Common 라이브러리의 메서드에서 반환되는 오류 및 상태 정보가 나와 있습니다.

Note

이러한 오류 및 상태 정보 코드는 다수의 API에 대해 공통적으로 적용됩니다.

코드	앞에 0이 없는 코드	메시지	설명
0x00000001	0x1	STATUS_NULL_ARG	필수 인수에 NULL이 전달되었습니다.
0x00000002	0x2	STATUS_INVALID_ARG	인수에 대해 잘못된 값이 지정되었습니다.
0x00000003	0x3	STATUS_INVALID_ARG_LEN	잘못된 인수 길이가 지정되었습니다.
0x00000004	0x4	STATUS_NOT_ENOUGH_MEMORY	메모리를 충분히 할당할 수 없습니다.
0x00000005	0x5	STATUS_BUFFER_TOO_SMALL	지정된 버퍼 크기가 너무 작습니다.
0x00000006	0x6	STATUS_UNEXPECTED_EOF	예기치 않게 파일 끝에 도달했습니다.
0x00000007	0x7	STATUS_FORMAT_ERROR	잘못된 형식이 발생했습니다.
0x00000008	0x8	STATUS_INVALID_HANDLE_ERROR	잘못된 핸들 값입니다.
0x00000009	0x9	STATUS_OPEN_FILE_FAILED	파일을 열지 못했습니다.
0x0000000a	0xa	STATUS_READ_FILE_FAILED	파일에서 읽지 못했습니다.

코드	앞에 0이 없는 코드	메시지	설명
0x0000000b	0xb	STATUS_WRITE_TO_FILE_FAILED	파일에 쓰지 못했습니다.
0x0000000c	0xc	STATUS_INTERNAL_ERROR	일반적으로 발생하지 않지만 SDK 또는 서비스 API 버그를 나타낼 수 있는 내부 오류입니다.
0x0000000d	0xd	STATUS_INVALID_OPERATION	잘못된 작업이 있거나 작업이 허용되지 않습니다.
0x0000000e	0xe	STATUS_NOT_IMPLEMENTED	기능이 구현되지 않았습니다.
0x0000000f	0xf	STATUS_OPERATION_TIMED_OUT	작업 시간이 초과되었습니다.
0x00000010	0x10	STATUS_NOT_FOUND	필요한 리소스가 없습니다.
0x00000011	0x11	STATUS_CREATE_THREAD_FAILED	스레드를 만들지 못했습니다.
0x00000012	0x12	STATUS_THREAD_NOT_ENOUGH_RESOURCES	다른 스레드를 생성하는데 필요한 리소스가 부족하거나 시스템에서 부과한 스레드 수 제한이 발생했습니다.

코드	앞에 0이 없는 코드	메시지	설명
0x00000013	0x13	STATUS_TH READ_INVA LID_ARG	잘못된 스레드 속성이 지정되었거나 다른 스레드가 이미 이 스레드에 참여하기 위해 대기 중입니다.
0x00000014	0x14	STATUS_TH READ_PERM ISSIONS	스레드 속성에 지정된 스케줄링 정책 및 매개변수를 설정할 권한이 없습니다.
0x00000015	0x15	STATUS_TH READ_DEAD LOCKED	교착 상태가 감지되었거나 조인 스레드가 호출 스레드를 지정합니다.
0x00000016	0x16	STATUS_TH READ_DOES _NOT_EXIST	지정된 스레드 ID를 가진 스레드를 찾을 수 없습니다.
0x00000017	0x17	STATUS_JO IN_THREAD _FAILED	스레드 조인 작업에서 알 수 없거나 일반적인 오류가 반환되었습니다.
0x00000018	0x18	STATUS_WA IT_FAILED	조건 변수를 기다릴 수 있는 최대 시간을 초과했습니다.
0x00000019	0x19	STATUS_CA NCEL_THRE AD_FAILED	스레드 취소 작업에서 알 수 없거나 일반적인 오류가 반환되었습니다.

코드	앞에 0이 없는 코드	메시지	설명
0x0000001a	0x1a	STATUS_THREAD_JOINABLE	스레드 조인 작업은 조인할 수 없는 스레드에서 요청됩니다.
0x0000001b	0x1b	STATUS_DETACH_THREAD_FAILED	스레드 분리 작업에서 알 수 없거나 일반적인 오류가 반환되었습니다.
0x0000001c	0x1c	STATUS_THREAD_ATTR_INIT_FAILED	스레드 속성 개체를 초기화하지 못했습니다.
0x0000001d	0x1d	STATUS_THREAD_ATTR_SET_STACK_SIZE_FAILED	스레드 속성 개체의 스택 크기를 설정하지 못했습니다.
0x0000001e	0x1e	STATUS_MEMORY_NOT_FREED	테스트에만 사용됩니다. 요청된 모든 메모리가 비워지지 않았음을 나타냅니다.

힙 라이브러리에서 반환된 오류 및 상태 코드

다음 표에는 Heap 라이브러리의 메서드에서 반환되는 오류 및 상태 정보가 나와 있습니다.

코드	메시지	설명
0x10000001	STATUS_HEAP_FLAGS_ERROR	플래그의 잘못된 조합이 지정되었습니다.
0x10000002	STATUS_HEAP_NOT_INITIALIZED	힙이 초기화되기 전에 작업이 시도되었습니다.

코드	메시지	설명
0x10000003	STATUS_HEAP_CORRUPTED	힙이 손상되었거나 (디버그 모드의) 보호 대역을 덮어 썼습니다. 클라이언트 코드의 버퍼 오버플로우가 힙 손상으로 이어질 수 있습니다.
0x10000004	STATUS_HEAP_VRAM_LIB_MISSING	VRAM (비디오 RAM) 사용자 또는 커널 모드 라이브러리를 로드할 수 없거나 누락되었습니다. 기본 플랫폼에서 VRAM 할당을 지원하는지 확인하십시오.
0x10000005	STATUS_HEAP_VRAM_LIB_REOPEN	VRAM 라이브러리를 열지 못했습니다.
0x10000006	STATUS_HEAP_VRAM_INIT_FUNC_SYMBOL	INIT 함수 내보내기를 로드하지 못했습니다.
0x10000007	STATUS_HEAP_VRAM_ALLOC_FUNC_SYMBOL	ALLOC 함수 내보내기를 로드하지 못했습니다.
0x10000008	STATUS_HEAP_VRAM_FREE_FUNC_SYMBOL	FREE 함수 내보내기를 로드하지 못했습니다.
0x10000009	STATUS_HEAP_VRAM_LOCK_FUNC_SYMBOL	LOCK 함수 내보내기를 로드하지 못했습니다.
0x1000000a	STATUS_HEAP_VRAM_UNLOCK_FUNC_SYMBOL	UNLOCK 함수 내보내기를 로드하지 못했습니다.
0x1000000b	STATUS_HEAP_VRAM_UNINIT_FUNC_SYMBOL	UNINIT 함수 내보내기를 로드하지 못했습니다.
0x1000000c	STATUS_HEAP_VRAM_GETMAX_FUNC_SYMBOL	GETMAX 함수 내보내기를 로드하지 못했습니다.

코드	메시지	설명
0x1000000d	STATUS_HEAP_DIRECT_MEM_INIT	하이브리드 힙에서 기본 힙 풀을 초기화하지 못했습니다.
0x1000000e	STATUS_HEAP_VRAM_INIT_FAILED	VRAM 동적 초기화에 실패했습니다.
0x1000000f	STATUS_HEAP_LIBRARY_FREE_FAILED	VRAM 라이브러리의 할당을 취소해 해당 라이브러리를 비우지 못했습니다.
0x10000010	STATUS_HEAP_VRAM_ALLOC_FAILED	VRAM 할당에 실패했습니다.
0x10000011	STATUS_HEAP_VRAM_FREE_FAILED	VRAM 비우기에 실패했습니다.
0x10000012	STATUS_HEAP_VRAM_MAP_FAILED	VRAM 매핑에 실패했습니다.
0x10000013	STATUS_HEAP_VRAM_UNMAP_FAILED	VRAM 매핑 해제에 실패했습니다.
0x10000014	STATUS_HEAP_VRAM_UNINIT_FAILED	VRAM 초기화 취소에 실패했습니다.
0x10000015	STATUS_INVALID_ALLOCATION_SIZE	
0x10000016	STATUS_HEAP_REALLOC_ERROR	
0x10000017	STATUS_HEAP_FILE_HEAP_FILE_CORRUPT	

MKVGen 라이브러리에서 반환된 오류 및 상태 코드

다음 표에는 라이브러리의 메서드에서 반환되는 오류 및 상태 정보가 포함되어 있습니다. MKVGen

코드	메시지	설명/ 권장 조치
0x32000001	STATUS_MKV_INVALID_FRAME_DATA	Frame 데이터 구조의 멤버가 잘못되었습니다. 재생 시간, 크기 및 프레임 데이터가 유효하고 에 지정된 한도 내에 있는지 확인하십시오 프로듀서 SDK 한도 .
0x32000002	STATUS_MKV_INVALID_FRAME_TIMESTAMP	프레임 타임스탬프가 잘못되었습니다. 계산된 PTS(프레젠테이션 타임스탬프) 및 DTS(디코딩 타임스탬프)가 조각의 시작 프레임에 대한 타임스탬프보다 크거나 같습니다. 이는 미디어 파이프라인 다시 시작 가능성 또는 인코더 안정성 문제를 나타냅니다. 문제 해결 정보는 오류: "Kinesis 비디오 클라이언트에 프레임을 제출하지 못 함" 를 참조하십시오.
0x32000003	STATUS_MKV_INVALID_CLUSTER_DURATION	잘못된 조각 기간이 지정되었습니다. 자세한 정보는 프로듀서 SDK 한도 을 참조하세요.
0x32000004	STATUS_MKV_INVALID_CONTENT_TYPE_LENGTH	잘못된 콘텐츠 유형 문자열 길이. 자세한 정보는 프로듀서 SDK 한도 을 참조하세요.
0x32000005	STATUS_MKV_NUMBER_TOO_BIG	EBML(Extensible Binary Meta Language) 형식으로 표시하기에는 너무 큰 숫자를 인코딩하려고 했습니다. 이 숫자가 SDK 클라이언트에 노출되면 안 됩니다.

코드	메시지	설명/ 권장 조치
0x32000006	STATUS_MKV_INVALID_CODEC_ID_LENGTH	코덱 ID 문자열 길이가 잘못되었습니다. 자세한 정보는 프로듀서 SDK 한도 를 참조하세요.
0x32000007	STATUS_MKV_INVALID_TRACK_NAME_LENGTH	추적 이름 문자열 길이가 잘못되었습니다. 자세한 정보는 프로듀서 SDK 한도 를 참조하세요.
0x32000008	STATUS_MKV_INVALID_CODEC_PRIVATE_LENGTH	코덱 프라이빗 데이터 길이가 잘못되었습니다. 자세한 정보는 프로듀서 SDK 한도 를 참조하세요.
0x32000009	STATUS_MKV_CODEC_PRIVATE_NULL	코덱 개인 데이터 (CPD)는 NULL이지만 CPD 크기는 0보다 큼니다.
0x3200000a	STATUS_MKV_INVALID_TIMECODE_SCALE	타임코드 스케일 값이 잘못되었습니다. 자세한 정보는 프로듀서 SDK 한도 를 참조하세요.
0x3200000b	STATUS_MKV_MAX_FRAME_TIMECODE	프레임 타임코드가 최댓값보다 큼니다. 자세한 정보는 프로듀서 SDK 한도 를 참조하세요.

코드	메시지	설명/ 권장 조치
0x3200000c	STATUS_MKV_LARGE_FRAME_TIMECODE	최대 프레임 타임코드에 도달했습니다. MKV 형식에서는 서명된 16비트를 사용하여 클러스터 시작에 대한 프레임의 상대 타임코드를 나타냅니다. 프레임 타임코드를 표시할 수 없는 경우 오류가 발생합니다. 이 오류는 타임코드 스케일 선택이 잘못되었거나 클러스터 기간이 너무 길어 프레임 타임코드가 서명된 16비트 공간에서 넘침을 나타냅니다.
0x3200000d	STATUS_MKV_INVALID_ANNEXB_NALU_IN_FRAME_DATA	잘못된 Annex-B 시작 코드가 발생했습니다. 예를 들어 Annex-B 적응 플래그가 지정되었으나 코드에서 0이 3개보다 많은 잘못된 시작 시퀀스가 발생했습니다. 잘못된 Annex-B 형식에는 바이트 스트림에서 0이 3개 이상인 시퀀스를 이스케이프하기 위한 "에물레이션 방지" 시퀀스가 있어야 합니다. 자세한 내용은 MPEG 사양을 참조하십시오. Android에서의 이 오류에 대한 자세한 내용은 Android에서 STATUS_MKV_INVALID_ANNEXB_NALU_IN_FRAME_DATA(0x3200000d) 오류 발생 단원을 참조하십시오.

코드	메시지	설명/ 권장 조치
0x3200000e	STATUS_MKV_INVALID_AVCC_NALU_IN_FRAME_DATA	적응형 AVCC 플래그가 지정된 경우 AVCC NALU 패키징이 유효하지 않습니다. 바이트스트림이 유효한 AVCC 형식인지 확인하십시오. 자세한 내용은 MPEG 사양을 참조하십시오.
0x3200000f	STATUS_MKV_BOTH_ANNEXB_AND_AVCC_SPECIFIED	적응형 AVCC와 부록-B NALU를 모두 지정했습니다. 둘 중 하나만 지정하거나 아무 것도 지정하지 마십시오.
0x32000010	STATUS_MKV_INVALID_ANNEXB_NALU_IN_CPD	Annex-B 적응 플래그 지정 시 CPD의 Annex-B 형식이 잘못되었습니다. CPD가 유효한 부록-B 형식인지 확인하십시오. 그렇지 않은 경우 CPD 부록-B 적응 플래그를 제거하십시오.
0x32000011	STATUS_MKV_PTS_DTS_ARE_NOT_SAME	Kinesis Video Streams는 프레임 시작 프레임에 대해 PTS (프레젠테이션 타임스탬프)와 DTS (디코딩 타임스탬프)를 동일하게 적용합니다. 둘 다 조각을 시작하는 키 프레임입니다.
0x32000012	STATUS_MKV_INVALID_H264_H265_CPD	H264/H265 코덱 프라이빗 데이터를 구문 분석하지 못했습니다.
0x32000013	STATUS_MKV_INVALID_H264_H265_SPS_WIDTH	코덱 프라이빗 데이터에서 너비를 추출하지 못했습니다.

코드	메시지	설명/ 권장 조치
0x32000014	STATUS_MKV_INVALID_H264_H265_SPS_HEIGHT	코덱 프라이빗 데이터에서 높이를 추출하지 못했습니다.
0x32000015	STATUS_MKV_INVALID_H264_H265_SPS_NALU	H264/H265 SPS NALU가 잘못되었습니다.
0x32000016	STATUS_MKV_INVALID_BIH_CPD	코덱 프라이빗 데이터의 비트맵 정도 헤더 형식이 잘못되었습니다.
0x32000017	STATUS_MKV_INVALID_HEVC_NALU_COUNT	잘못된 HEVC(High Efficiency Video Coding) NALU(Network Abstraction Layer Units) 수입니다.
0x32000018	STATUS_MKV_INVALID_HEVC_FORMAT	잘못된 HEVC 형식입니다.
0x32000019	STATUS_MKV_HEVC_SPS_NALU_MISSING	SPS(Sequence Parameter Set)에 HEVC NALU가 누락되었습니다.
0x3200001a	STATUS_MKV_INVALID_HEVC_SPS_NALU_SIZE	잘못된 HEVC SPS NALU 크기입니다.
0x3200001b	STATUS_MKV_INVALID_HEVC_SPS_CHROMA_FORMAT_IDC	잘못된 채도 형식 IDC입니다.
0x3200001c	STATUS_MKV_INVALID_HEVC_SPS_RESERVED	잘못된 HEVC 예약 SPS입니다.

코드	메시지	설명/ 권장 조치
0x3200001d	STATUS_MKV_MIN_ANNEX_B_CPD_SIZE	AnnexBb 최소 코덱 프라이빗 베타 값 크기 H264의 경우, 이 값이 11보다 크거나 같아야 합니다. H265의 경우, 이 값이 15보다 크거나 같아야 합니다.
0x3200001e	STATUS_MKV_ANNEXB_CPD_MISSING_NALUS	Annex-B NALU의 코덱 프라이빗 데이터가 누락되었습니다.
0x3200001f	STATUS_MKV_INVALID_ANNEXB_CPD_NALUS	Annex-B NALU의 코덱 프라이빗 베타가 잘못되었습니다.
0x32000020	STATUS_MKV_INVALID_TAG_NAME_LENGTH	잘못된 태그 이름 길이. 유효한 값의 범위는 0보다 크고 128보다 작아야 합니다.
0x32000021	STATUS_MKV_INVALID_TAG_VALUE_LENGTH	잘못된 태그 값 길이. 유효한 값은 0보다 크고 256보다 작습니다.
0x32000022	STATUS_MKV_INVALID_GENERATOR_STATE_TAGS	잘못된 생성기 상태 태그입니다.
0x32000023	STATUS_MKV_INVALID_AAC_CPD_SAMPLING_FREQUENCY_INDEX	빈도 인덱스를 샘플링하는 AAC 코덱 프라이빗 데이터가 잘못되었습니다.
0x32000024	STATUS_MKV_INVALID_AAC_CPD_CHANNEL_CONFIG	AAC 코덱 프라이빗 데이터 채널 구성이 잘못되었습니다.
0x32000025	STATUS_MKV_INVALID_AAC_CPD	잘못된 AAC 코덱 프라이빗 데이터입니다.
0x32000026	STATUS_MKV_TRACK_INFO_NOT_FOUND	트랙 정보를 찾을 수 없습니다.

코드	메시지	설명/ 권장 조치
0x32000027	STATUS_MKV_INVALID_SEGMENT_UUID	잘못된 세그먼트 UUID입니다.
0x32000028	STATUS_MKV_INVALID_TRACK_UUID	잘못된 트랙 UID입니다.
0x32000029	STATUS_MKV_INVALID_CLIENT_ID_LENGTH	
0x3200002a	STATUS_MKV_INVALID_AMS_ACM_CPD	
0x3200002b	STATUS_MKV_MISSING_SPS_FROM_H264_CPD	
0x3200002c	STATUS_MKV_MISSING_PPS_FROM_H264_CPD	
0x3200002d	STATUS_MKV_INVALID_PARENT_TYPE	

Trace 라이브러리에서 반환된 오류 및 상태 코드

다음 표에는 Trace 라이브러리의 메서드에서 반환되는 오류 및 상태 정보가 포함되어 있습니다.

코드	메시지
0x10100001	STATUS_MIN_PROFILER_BUFFER

Utils 라이브러리에서 반환한 오류 및 상태 코드

다음 표에는 라이브러리의 메서드에서 반환되는 오류 및 상태 정보가 나와 있습니다. Utils

코드	메시지
0x40000001	STATUS_INVALID_BASE64_ENCODE

코드	메시지
0x40000002	STATUS_INVALID_BASE
0x40000003	STATUS_INVALID_DIGIT
0x40000004	STATUS_INT_OVERFLOW
0x40000005	STATUS_EMPTY_STRING
0x40000006	STATUS_DIRECTORY_OPEN_FAILED
0x40000007	STATUS_PATH_TOO_LONG
0x40000008	STATUS_UNKNOWN_DIR_ENTRY_TYPE
0x40000009	STATUS_REMOVE_DIRECTORY_FAILED
0x4000000a	STATUS_REMOVE_FILE_FAILED
0x4000000b	STATUS_REMOVE_LINK_FAILED
0x4000000c	STATUS_DIRECTORY_ACCESS_DENIED
0x4000000d	STATUS_DIRECTORY_MISSING_PATH
0x4000000e	STATUS_DIRECTORY_ENTRY_STAT_ERROR
0x4000000f	STATUS_STRFTIME_FAILED
0x40000010	STATUS_MAX_TIMESTAMP_FORMAT_STR_LEN_EXCEEDED
0x40000011	STATUS_UTIL_MAX_TAG_COUNT
0x40000012	STATUS_UTIL_INVALID_TAG_VERSION
0x40000013	STATUS_UTIL_TAGS_COUNT_NON_ZERO_TAGS_NULL

코드	메시지
0x40000014	STATUS_UTIL_INVALID_TAG_NAME_LEN
0x40000015	STATUS_UTIL_INVALID_TAG_VALUE_LEN
0x4000002a	STATUS_EXPONENTIAL_BACKOFF_INVALID_STATE
0x4000002b	STATUS_EXPONENTIAL_BACKOFF_RETRIES_EXHAUSTED
0x4000002c	STATUS_THREADPOOL_MAX_COUNT
0x4000002d	STATUS_THREADPOOL_INTERNAL_ERROR
0x40100001	STATUS_HASH_KEY_NOT_PRESENT
0x40100002	STATUS_HASH_KEY_ALREADY_PRESENT
0x40100003	STATUS_HASH_ENTRY_ITERATION_ABORT
0x41000001	STATUS_BIT_READER_OUT_OF_RANGE
0x41000002	STATUS_BIT_READER_INVALID_SIZE
0x41100001	STATUS_TIMER_QUEUE_STOP_SCHEDULING
0x41100002	STATUS_INVALID_TIMER_COUNT_VALUE
0x41100003	STATUS_INVALID_TIMER_PERIOD_VALUE
0x41100004	STATUS_MAX_TIMER_COUNT_REACHED
0x41100005	STATUS_TIMER_QUEUE_SHUTDOWN

코드	메시지
0x41200001	STATUS_SEMAPHORE_OPERATION_AFTER_SHUTDOWN
0x41200002	STATUS_SEMAPHORE_ACQUIRE_WHEN_LOCKED
0x41300001	STATUS_FILE_LOGGER_INDEX_FILE_INVALID_SIZE

View 라이브러리에서 반환된 오류 및 상태 코드

다음 표에는 View 라이브러리의 메서드에서 반환되는 오류 및 상태 정보가 나와 있습니다.

코드	메시지	설명
0x30000001	STATUS_MIN_CONTENT_VIEW_ITEMS	잘못된 콘텐츠 보기 항목 수가 지정되었습니다. 자세한 정보는 프로듀서 SDK 한도 를 참조하세요.
0x30000002	STATUS_INVALID_CONTENT_VIEW_DURATION	잘못된 콘텐츠 보기 지속 시간이 지정되었습니다. 자세한 정보는 프로듀서 SDK 한도 를 참조하세요.
0x30000003	STATUS_CONTENT_VIEW_NO_MORE_ITEMS	헤더 위치를 지나치려는 시도가 있었습니다.
0x30000004	STATUS_CONTENT_VIEW_INVALID_INDEX	잘못된 인덱스가 지정되어 있습니다.
0x30000005	STATUS_CONTENT_VIEW_INVALID_TIMESTAMP	타임스탬프 또는 타임스탬프 중첩이 잘못되었습니다. 프레임 디코딩 타임스탬프는 이전 프레임 타임스탬프에 이전 프레임 지속 시간을 더한 값

코드	메시지	설명
		<p>보다 크거나 같아야 합니다.</p> <p><code>`DTS(n) >= DTS(n-1) + Duration(n-1)`</code> 일반적으로 이 오류는 "불안정한" 인코더를 나타냅니다. 인코더는 인코딩된 프레임의 버스트를 생성하는데 이러한 프레임의 타임스탬프는 인트라 프레임 지속 시간보다 작아야 합니다. 또는 스트림이 SDK 타임스탬프를 사용하도록 구성되어 있고 프레임이 프레임 기간보다 빠르게 전송됩니다. <code>StreamInfo.StreamCaps</code> 구조에서 프레임 기간을 더 짧게 지정하면 인코더의 일부 "지터"와 관련해 도움이 됩니다. 예를 들어 스트림이 25FPS인 경우 각 프레임의 지속 시간은 40ms입니다. 하지만 인코더 "지터"를 처리하려면 해당 프레임 지속 시간의 절반 (20ms) 을 사용하는 것이 좋습니다. 일부 스트림의 경우 오류를 감지하기 위해 보다 정확하게 타이밍을 제어해야 합니다.</p>
0x30000006	STATUS_INVALID_CONTENT_VIEW_LENGTH	잘못된 콘텐츠 보기 항목 데이터 길이가 지정되었습니다.

PutFrame 콜백에서 반환된 오류 및 상태 코드 - C 프로듀서 라이브러리

다음 섹션에는 C Producer 라이브러리 내 PutFrame 작업에 대해 콜백에서 반환되는 오류 및 상태 정보가 포함되어 있습니다.

코드	메시지	설명	권장 조치
0x15000001	STATUS_STOP_CALLBACK_CHAIN	콜백 체인이 중지되었습니다.	
0x15000002	STATUS_MAX_CALLBACK_CHAIN	최대 콜백 체인에도 달했습니다.	
0x15000003	STATUS_INVALID_PLATFORM_CALLBACKS_VERSION	잘못된 PlatformCallbacks 구조 버전.	구조의 올바른 현재 버전을 지정합니다.
0x15000004	STATUS_INVALID_PRODUCER_CALLBACKS_VERSION	잘못된 ProducerCallbacks 구조 버전.	구조의 올바른 현재 버전을 지정합니다.
0x15000005	STATUS_INVALID_STREAM_CALLBACKS_VERSION	잘못된 StreamCallbacks 구조 버전.	구조의 올바른 현재 버전을 지정합니다.
0x15000006	STATUS_INVALID_AUTH_CALLBACKS_VERSION	잘못된 AuthCallbacks 구조 버전.	구조의 올바른 현재 버전을 지정합니다.
0x15000007	STATUS_INVALID_API_CALLBACKS_VERSION	잘못된 ApiCallbacks 구조 버전.	구조의 올바른 현재 버전을 지정합니다.
0x15000008	STATUS_INVALID_AWS_CREDENTIALS_VERSION	잘못된 AwsCredentials 구조 버전.	구조의 올바른 현재 버전을 지정합니다.

코드	메시지	설명	권장 조치
0x15000009	STATUS_MAXIMUM_REQUEST_HEADER_COUNT	최대 요청 헤더 수에 도달했습니다.	
0x1500000a	STATUS_MAXIMUM_REQUEST_HEADER_NAME_LEN	최대 요청 헤더 이름 길이에 도달했습니다.	
0x1500000b	STATUS_MAXIMUM_REQUEST_HEADER_VALUE_LEN	최대 요청 헤더 값 길이에 도달했습니다.	
0x1500000c	STATUS_INVALID_API_CALL_RETURN_JSON	API 호출에 대한 잘못된 반환 JSON입니다.	
0x1500000d	STATUS_CURL_INIT_FAILED	Curl 초기화에 실패했습니다.	
0x1500000e	STATUS_CURL_LIBRARY_INIT_FAILED	Curl lib 초기화에 실패했습니다.	
0x1500000f	STATUS_INVALID_DESCRIPTOR_RETURN_JSON	에 대한 반환 JSON이 잘못되었습니다. DescribeStream	
0x15000010	STATUS_HMAC_GENERATION_ERROR	HMAC 생성 오류입니다.	
0x15000011	STATUS_IOT_FAILED	IoT 인증에 실패했습니다.	

코드	메시지	설명	권장 조치
0x15000012	STATUS_MAXIMUM_ROLE_ALIASES_LEN_EXCEEDED	최대 역할 별칭 길이에 도달했습니다.	더 짧은 별칭 길이를 지정하십시오.
0x15000013	STATUS_MAXIMUM_USER_AGENT_NAME_POSTFIX_LEN_EXCEEDED	최대 에이전트 이름 접미사 길이에 도달했습니다.	
0x15000014	STATUS_MAXIMUM_CUSTOM_USER_AGENT_LEN_EXCEEDED	최대 고객 사용자 에이전트 길이에 도달했습니다.	
0x15000015	STATUS_INVALID_USER_AGENT_LENGTH	잘못된 사용자 에이전트 길이입니다.	
0x15000016	STATUS_INVALID_ENDPOINT_CACHING_PERIOD	잘못된 엔드포인트 캐싱 기간입니다.	24시간 미만의 캐싱 기간을 지정하십시오.
0x15000017	STATUS_IOT_TIMESTAMP_EXPIRATION_OCCURS_IN_PAST	IoT 만료 타임스탬프는 과거에 발생했습니다.	
0x15000018	STATUS_IOT_TIMESTAMP_PARSING_FAILED	IoT 만료 파싱이 실패했습니다.	

코드	메시지	설명	권장 조치
0x15000019	STATUS_DUPLICATE_PRODUCER_CALLBACK_FUNC		
0x1500001a	STATUS_DUPLICATE_STREAM_CALLBACK_FREE_FUNC		
0x1500001b	STATUS_DUPLICATE_AUTH_CALLBACK_FREE_FUNC		
0x1500001c	STATUS_DUPLICATE_API_CALLBACK_FREE_FUNC		
0x1500001d	STATUS_FILE_LOGGER_INDEX_FILE_TOO_LARGE		
0x1500001e	STATUS_MAXIMUM_IOT_THING_NAME_LENGTH		
0x1500001f	STATUS_IOT_CREATE_LWS_CONTEXT_FAILED		

코드	메시지	설명	권장 조치
0x15000020	STATUS_INVALID_CERT_PATH		
0x15000022	STATUS_FILE_CREDENTIAL_PROVIDER_OPEN_FILE_FAILED		
0x15000023	STATUS_FILE_CREDENTIAL_PROVIDER_INVALID_FILE_LENGTH		
0x15000024	STATUS_FILE_CREDENTIAL_PROVIDER_INVALID_FILE_FORMAT		
0x15000026	STATUS_STREAM_BEING_SHUTDOWN		
0x15000027	STATUS_CLIENT_BEING_SHUTDOWN		
0x15000028	STATUS_CONTINUOUS_RETRY_RETRY_FAILED		

코드	메시지	설명	권장 조치
0x16000001	STATUS_CURL_PERFORM_FAILED		
0x16000002	STATUS_IOT_INVALID_RESPONSE_LENGTH		
0x16000003	STATUS_IOT_NULL_AWS_CREDS		
0x16000004	STATUS_IOT_INVALID_URI_LEN		
0x16000005	STATUS_TIMESTAMP_STRING_UNRECOGNIZED_FORMAT		

NAL(Network Abstraction Layer) 적응 플래그 참조

이 단원에는 `StreamInfo.NalAdaptationFlags` 열거에 사용할 수 있는 플래그에 대한 정보가 포함되어 있습니다.

애플리케이션의 [기본 스트림](#)은 Annex-B 또는 AVCC 형식일 수 있습니다.

- Annex-B 형식은 [NALU \(네트워크 추상화 계층 단위\)](#) 를 2바이트의 0, 1~3바이트, 숫자 1 (예: 00000001) 로 구분합니다.
- AVCC 형식도 NALU를 줄 바꿈하지만 각각의 NALU 앞에 NALU의 크기를 나타내는 값이 옵니다(일반적으로 4바이트).

많은 인코더가 Annex-B 비트스트림 형식을 생성합니다. 일부 상위 수준 비트스트림 프로세서 (예: 재생 엔진 또는 [미디어](#) 소스 확장 (MSE) 플레이어) 는 프레임에 AVCC 형식을 사용합니다. AWS Management Console

H.264용 SPS/PPS(Sequence Parameter Set/Picture Parameter Set)인 CPD(Codec Private Data)도 Annex-B 또는 AVCC 형식일 수 있습니다. 하지만 CPD의 경우, 앞서 설명한 것과 형식이 다릅니다.

플래그는 다음과 같이 NALU를 프레임 데이터와 CPD를 위한 AVCC 또는 Annex-B에 적응시키라고 SDK에게 지시합니다.

플래그	적응
NAL_ADAPTATION_FLAG_NONE	적응이 안 됐어요.
NAL_ADAPTATION_ANNEXB_NALS	부록-B NALU를 AVC NALU에 맞게 조정하십시오.
NAL_ADAPTATION_AVC_NALS	AVC ANLU를 부록-B NALU에 맞게 조정하십시오.
NAL_ADAPTATION_ANNEXB_CPD_NALS	코덱 프라이빗 데이터용 부록 B NALU를 AVC 형식 NALU로 조정하십시오.
NAL_ADAPTATION_ANNEXB_CPD_AND_FRAME_NALS	코덱용 부록 B NALU를 조정하고 프라이빗 데이터를 AVCC 형식 NALU로 설정하십시오.

[NALU 유형에 대한 자세한 내용은 섹션 1.3: RFC 3984의 네트워크 추상화 계층 단위 유형을 참조하십시오.](#)

프로듀서 SDK 구조

이 섹션에는 Kinesis Video Streams Producer 객체에 데이터를 제공하는 데 사용할 수 있는 구조에 대한 정보가 포함되어 있습니다.

주제

- [DeviceInfo/DefaultDeviceInfoProvider](#)
- [StorageInfo](#)

DeviceInfo/DefaultDeviceInfoProvider

DeviceInfo 및 DefaultDeviceInfoProvider 객체는 Kinesis Video Streams 프로듀서 객체의 동작을 제어합니다.

멤버 필드

- 버전 — 현재 버전의 코드베이스에서 올바른 버전의 구조가 사용되는지 확인하는 데 사용되는 정수 값입니다. 현재 버전은 DEVICE_INFO_CURRENT_VERSION 매크로를 사용하여 지정됩니다.
- 이름 - 사람이 읽을 수 있는 장치 이름입니다.
- 태그 수/태그 — 현재 사용되지 않습니다.
- StreamCount — 디바이스에서 처리할 수 있는 최대 스트림 수입니다. 이 값은 초기에 포인터를 위한 스토리지를 스트림 객체에 미리 할당하지만 실제 스트림 객체는 나중에 생성됩니다. 기본값은 16 스트림이지만 DefaultDeviceInfoProvider.cpp 파일에서 이 수를 변경할 수 있습니다.
- storageInfo: 기본 스토리지 구성을 설명하는 객체입니다. 자세한 정보는 [StorageInfo](#)을 참조하세요.

StorageInfo

Kinesis Video Streams의 기본 스토리지 구성을 지정합니다.

기본 구현은 스트리밍에 최적화된 빠른 힙의 낮은 조각화를 기반으로 합니다. 주어진 플랫폼에 중복 재가 가능한 MEMALLOC 할당자를 사용합니다. 어떤 플랫폼은 피지컬 페이지로 할당을 지원하지 않아도 가상 메모리가 할당됩니다. 메모리가 사용될 때 가상 페이지는 피지컬 페이지로 지원됩니다. 결과적으로 스토리지가 충분히 사용되지 않을 때에도 전체 시스템에 메모리 부족 압력이 발생합니다.

다음 수식을 기반으로 기본 스토리지 크기를 계산합니다. DefragmentationFactor를 1.2(20퍼센트)로 설정해야 합니다.

$$\text{Size} = \text{NumberOfStreams} * \text{AverageFrameSize} * \text{FramesPerSecond} * \text{BufferDurationInSeconds} * \text{DefragmentationFactor}$$

다음 예제에서는 디바이스에 오디오 및 비디오 스트림이 있습니다. 오디오 스트림의 초당 샘플 수는 512이며 샘플은 평균 100바이트입니다. 비디오 스트림의 초당 프레임 수는 25이며 평균 10,000바이트입니다. 각 스트림의 버퍼 지속 시간은 3분입니다.

$$\text{Size} = (512 * 100 * (3 * 60) + 25 * 10000 * (3 * 60)) * 1.2 = (9216000 + 45000000) * 1.2 = 65059200 = \sim 66\text{MB}.$$

디바이스에 사용 가능한 메모리가 더 많은 경우 심각한 조각화를 방지하기 위해 스토리지에 메모리를 추가하는 것이 좋습니다.

인코딩 복잡성이 높거나 (움직임이 많아 프레임 크기가 더 커지는 경우) 대역폭이 낮은 경우 모든 스트림의 전체 버퍼를 수용할 수 있을 만큼 스토리지 크기가 적절한지 확인하십시오. 생산자가 메모리 부족에 도달하면 스토리지 오버플로 압력 콜백 () 을 내보냅니다. StorageOverflowPressureFunc 하지만 콘텐츠 스토어에 사용 가능한 메모리가 없는 경우 Kinesis Video Streams로 푸시되는 프레임이 삭제되고 오류 STATUS_STORE_OUT_OF_MEMORY = 0x5200002e () 가 발생합니다. 자세한 정보는 [클라이언트 라이브러리에서 반환된 오류 및 상태 코드](#)을 참조하세요. 애플리케이션 승인(ACK)을 사용할 수 없거나 지속적인 ACK가 표시되는 경우에도 이 결과가 발생할 수 있습니다. 이 경우 이전 프레임이 손실되기 전에 버퍼가 “버퍼 지속 시간” 용량까지 채워집니다.

멤버 필드

- 버전 — 현재 버전의 코드베이스에서 올바른 버전의 구조가 사용되는지 확인하는 데 사용되는 정수 값입니다.
- StorageType — DEVICE_STORAGE_TYPE 스토리지의 기본 백업 및 구현을 지정하는 열거입니다. 현재 지원되는 값은 DEVICE_STORAGE_TYPE_IN_MEM입니다. 향후 구현에서는 DEVICE_STORAGE_TYPE_HYBRID_FILE이 지원되어, 스토리지가 파일 지원 콘텐츠 저장소로 다시 돌아갈 예정입니다.
- StorageSize - 사전 할당할 스토리지 크기 (바이트) 입니다. 최소 할당은 10MB이고, 최대 할당은 10GB입니다. (이 값은 파일 지원 콘텐츠 스토어의 향후 구현에서 변경됩니다.)
- SpillRatio — 보조 오버플로 스토리지 (파일 스토리지) 와 달리 직접 메모리 스토리지 유형 (RAM) 에서 할당할 스토리지의 비율을 나타내는 정수 값입니다. 현재 사용 중이지 않습니다.
- rootDirectory: 파일 지원 콘텐츠 저장소가 있는 디렉터리의 경로입니다. 현재 사용 중이지 않습니다.

Kinesis 비디오 스트림 구조

다음 구조를 사용하여 Kinesis 비디오 스트림의 인스턴스에 데이터를 제공할 수 있습니다.

주제

- [StreamDefinition/StreamInfo](#)
- [ClientMetrics](#)
- [StreamMetrics](#)

StreamDefinition/StreamInfo

C++ 계층의 StreamDefinition 객체는 PIC(Platform Independent Code)의 StreamInfo 객체를 래핑하고 생성자의 몇 가지 기본값을 제공합니다.

멤버 필드

필드	데이터 유형	설명	기본값
stream_name	string	선택적 스트림 이름. 스트림 이름 길이에 대한 자세한 내용은 프로듀서 SDK 한도 단원을 참조하십시오. 각 스트림은 고유한 이름을 지녀야 합니다.	이름을 지정하지 않으면 임의로 이름이 생성됩니다.
retention_period	duration<uint64_t, ratio<3600>>	스트림 보존 기간(초)입니다. 0을 지정하면 보관하지 않음을 나타냅니다.	3600(1시간)
tags	const map<string, string>*	사용자 정보가 포함된 키/값 페어의 맵입니다. 스트림에 태그 세트가 이미 있는 경우 새 태그는 기존 태그 세트에 추가됩니다.	태그 없음
kms_key_id	string	스트림을 암호화하는데 사용할 AWS KMS 키 ID입니다. 자세한 정보는 Kinesis Video Streams의 데이터 보호 를 참조하세요.	기본 KMS 키(aws/kinesis-video)입니다.
streaming_type	STREAMING_TYPE 열거	지원되는 유일한 값은 STREAMING	

필드	데이터 유형	설명	기본값
		<code>_TYPE_REALTIME</code> 입니다.	
<code>content_type</code>	<code>string</code>	스트림의 콘텐츠 형식. Kinesis Video Streams 콘솔은 <code>video/h264</code> 다음 형식의 콘텐츠를 재생할 수 있습니다.	<code>video/h264</code>
<code>max_latency</code>	<code>duration<uint64_t, milli></code>	천분의 1초 단위의 스트림 최대 지연 시간. 버퍼 지속 시간이 이 시간을 초과하면 스트림 지연 시간 압력 콜백(지정된 경우)이 호출됩니다. <code>0</code> 을 지정하면 스트림 지연 시간 압력 콜백이 회신되지 않습니다.	<code>milliseconds::zero()</code>

필드	데이터 유형	설명	기본값
fragment_duration	duration< uint64_t>	원하는 조각 지속 시간(초)입니다. 이 값은 key_frame_fragmentation 값과 함께 사용됩니다. 이 값이 이면 false Kinesis Video Streams는 이 기간이 경과한 후 키프레임에 프래그먼트를 생성합니다. 예를 들어, AAC(Advanced Audio Coding) 오디오 스트림의 각 프레임은 키 프레임입니다. key_frame_fragmentation = false를 지정하면 이 시간이 만료된 후 키 프레임에서 조각화가 수행되어 2초 조각이 생성됩니다.	2

필드	데이터 유형	설명	기본값
timecode_scale	duration< uint64_t, milli>	MKV 클러스터 내 프레임에 대한 타임코드 세부 수준을 지정하는 MKV 타임코드 척도 (밀리초)입니다. MKV 프레임 타임코드는 언제나 클러스터의 시작과 연관됩니다. MKV는 서명된 16비트 값 (0-32767)을 사용하여 클러스터(조각) 내 타임코드를 표시합니다. 프레임 타임코드가 지정된 타임코드 스케일로 표시될 수 있는지 확인하십시오. 기본 타임코드 척도 값인 1ms를 지정할 경우 표시할 수 있는 최대 프레임은 32767ms ≈ 32초입니다. 이 값은 Kinesis Video Streams 서비스 할당량 에 지정된 최대 조각 지속 시간인 10초를 초과합니다.	1

필드	데이터 유형	설명	기본값
key_frame_fragmentation	bool	키 프레임에서 조각을 생성할지 여부입니다. true이면 SDK는 키 프레임이 있을 때마다 조각의 시작을 생성합니다. 만약 false Kinesis Video Streams가 fragment_duration 적어도 기다린 후 그 뒤에 있는 키 프레임에 새 프래그먼트를 생성합니다.	true
frame_timecodes	bool	현재 시간 콜백을 사용하여 프레임 타임코드를 사용할지 타임스탬프를 생성할지 여부. 대부분의 인코더는 프레임과 함께 타임스탬프를 생성하지 않습니다. 따라서 이 파라미터를 false 지정하면 프레임이 Kinesis Video Streams에 추가될 때 타임스탬프가 지정됩니다.	true

필드	데이터 유형	설명	기본값
absolute_fragment_times	bool	Kinesis Video Streams는 MKV를 기본 패키징 메커니즘으로 사용합니다. MKV 사양에서는 엄격히 프레임 타임코드가 클러스터(조각)의 시작에 상대적입니다. 하지만 클러스터 타임코드는 스트림 시작 시간에 절대적이거나 상대적일 수 있습니다. 타임스탬프가 상대적이면 PutMedia 서비스 API 호출은 선택적 스트림 시작 타임스탬프를 사용하고 클러스터 타임스탬프를 조정합니다. 서비스는 항상 조각을 조각의 절대적 타임스탬프와 함께 저장합니다.	true
fragment_acks	bool	애플리케이션 레벨 프래그먼트 ACK (승인) 수신 여부	true는 SDK가 ACK를 수신하고 이에 따라 실행된다는 의미입니다.
restart_on_error	bool	특정 오류에 따라 재시작할지 여부.	true를 지정하면 SDK는 오류가 발생할 경우 스트리밍을 다시 시작하려고 합니다.

필드	데이터 유형	설명	기본값
recalculate_metrics	bool	지표를 다시 계산할지 여부입니다. 각 호출에 따라 지표를 검색하고 최신 "실행"값을 다시 계산하여 CPU에 영향을 적게 줍니다. 매우 낮은 전력/풋프린트 디바이스에서는 CPU 주기를 절약하기 위해 이 값을 false로 설정해야 할 수 있습니다. 그렇지 않으면 이 값에는 false를 사용하지 않는 것이 좋습니다.	true

필드	데이터 유형	설명	기본값
nal_adaptation_flags	uint32_t	<p>NALU(Network Abstraction Layer Unit) 적응 플래그를 지정합니다. 비트스트림이 H.264로 암호화되면 NALU에서 원시 또는 패키징된 상태로 처리될 수 있습니다. 이러한 비트스트림은 Annex-B 또는 AVCC 형식일 수 있습니다. 대부분의 기본 스트림 생산자와 소비자 (읽기 인코더 및 디코더)는 Annex-B 형식을 사용합니다. Annex-B 형식은 오류 복구와 같은 이점이 있기 때문입니다. 더 높은 수준의 시스템은 MPEG, HLS, DASH 등의 기본 형식인 AVCC 형식을 사용합니다. 콘솔 재생은 브라우저의 MSE(미디어 소스 확장)를 사용하여 AVCC 형식을 사용하는 스트림을 디코딩하고 재생합니다. H.264의 경우(그리고 M-JPEG 및 H.265의 경우) SDK는 적응 기능을 제공합니다.</p> <p>대부분의 초기 스트림은 다음 형식입니</p>	기본값은 프레임 데이터와 CPD(코덱 프라이빗 데이터) 모두에 대해 Annex-B 형식을 AVCC 형식으로 조정하는 것입니다.

필드	데이터 유형	설명	기본값
		<p>다. 이 예제에서 Ab는 Annex-B 시작 코드 (001 또는 0001)입니다.</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin: 10px 0;"> <pre>Ab(Sps)Ab (Pps)Ab(I-frame)Ab(P/B-frame) Ab(P/B-frame)... Ab(Sps)Ab (Pps)Ab(I-frame)Ab(P/B-frame) Ab(P/B-frame)</pre> </div> <p>H.264의 경우 코덱 개인 데이터 (CPD) 는 SPS (시퀀스 매개 변수 세트) 및 PPS (그림 매개 변수 세트) 매개 변수에 포함되며 AVCC 형식에 맞게 조정할 수 있습니다. 미디어 파이프라인이 CPD를 별도로 제공하지 않는 한, 애플리케이션은 프레임에서 CPD를 추출할 수 있습니다. 첫 번째 IDR 프레임 (SPS와 PPS를 포함해야 함) 을 찾아 두 개의 NALU (즉) 를 추출한 다음 CPD 입력에 설정하면 이 작업을 수행할 수 있습니다. Ab(Sps)Ab</p>	

필드	데이터 유형	설명	기본값
		(Pps) StreamDefinition 자세한 정보는 NAL 적응 플래그 를 참조하세요.	
frame_rate	uint32_t	예상 프레임 속도입니다. 이 값은 버퍼링 니즈를 계산하는 데 사용됩니다.	25
avg_bandwidth_bps	uint32_t	스트림용 예상 평균 대역폭. 이 값은 버퍼링 니즈를 계산하는 데 사용됩니다.	4 * 1024 * 1024

필드	데이터 유형	설명	기본값
buffer_duration	duration<uint64_t>	스트림 버퍼 지속 시간(초)입니다. SDK는 프레임을 콘텐츠 저장소에 최대 기간 동안 보관하며buffer_duration , 이 기간이 지나면 창이 앞으로 이동함에 따라 이전 프레임이 삭제됩니다. 드롭되는 프레임이 백엔드로 전송되지 않은 경우 드롭된 프레임 콜백이 호출됩니다. 현재 버퍼 지속 시간이 max_latency 보다 크면 스트림 지연 시간 압력 콜백이 호출됩니다. 조각 지속 ACK가 수신되면 버퍼가 다음 조각 시작에 맞게 잘립니다. 따라서 콘텐츠가 클라우드에 내구성 있게 지속되므로 더 이상 로컬 디바이스에 콘텐츠를 저장할 필요가 없습니다.	120

필드	데이터 유형	설명	기본값
replay_duration	duration< uint64_t>	재시작을 활성화한 경우 오류가 발생하는 동안 현재 리더를 뒤로 돌려 재생하는 데 걸리는 시간 (초) 입니다. 버퍼가 시작되면 롤백이 중지됩니다(스트리밍이 막 시작되었거나 지속적인 ACK가 나타나는 경우). 롤백은 조각화 시작을 나타내는 키 프레임에 랜딩하려고 합니다. '재시작을 초래한다'는 오류가 작동하지 않는 호스트 (호스트가 아직 살아 있고 내부 버퍼에 프레임 데이터가 들어 있음) 를 나타내지 않는 경우 마지막으로 수신한 ACK 프레임에서 롤백이 중지됩니다. 그런 다음 전체 조각이 호스트 메모리에 이미 저장되어 있으므로 다음 키 프레임으로 앞으로 롤합니다.	40

필드	데이터 유형	설명	기본값
connection_staleness	duration< uint64_t>	SDK가 버퍼링 ACK를 수신하지 않을 경우 스트림 비활성 콜백이 호출되기까지 걸리는 시간 (초) 입니다. 이는 디바이스에서 프레임이 전송되고 있지만 백엔드가 프레임을 승인하지 않음을 나타냅니다. 이 조건은 중간 흡 또는 로드 밸런서에서 연결이 끊어졌음을 나타냅니다.	30
codec_id	string	MKV 트랙의 Codec ID	V_MPEG4/ISO/AVC
track_name	string	MKV 트랙 이름	kinesis_video

필드	데이터 유형	설명	기본값
codecPrivateData	unsigned char*	CPD(코덱 프라이빗 데이터) 버퍼입니다. 미디어 파이프라인이 스트림이 시작되기 전 CPD에 대한 정보를 가지고 있다면 미디어 파이프라인은 StreamDefinition.codecPrivateData에 설정됩니다. 비트가 복사되며, 스트림을 생성하기 위한 호출 후에 버퍼를 재사용하거나 비울 수 있습니다. 하지만 스트림을 생성할 때 데이터를 사용할 수 없는 경우 함수의 오버로드 중 하나에 데이터를 설정할 수 있습니다. KinesisVideoStream.start(cpd)	null
codecPrivateData크기	uint32_t	코덱 프라이빗 데이터 버퍼 크기입니다.	0

ClientMetrics

ClientMetrics객체를 호출하면 채워집니다getKinesisVideoMetrics.

멤버 필드

필드	데이터 유형	설명
version	UINT32	구조의 버전은 CLIENT_METRICS_CURRENT_VERSION 매크로에 정의되어 있습니다.
contentStoreSize	UINT64	바이트 단위의 전체 콘텐츠 스토어 크기. DeviceInfo.StorageInfo.storageSize 에서 이 값을 지정합니다.
contentStoreAvailable크기	UINT64	현재 사용 가능한 스토리지 크기 (바이트)
contentStoreAllocated크기	UINT64	현재 할당된 크기. 내부 기록 및 콘텐츠 저장소 구현으로 인해 할당된 크기 더하기 사용 가능한 크기는 전체 스토리지 크기 보다 약간 작아야 합니다.
totalContentViews사이즈	UINT64	모든 스트림에 대한 모든 콘텐츠에 할당된 메모리 크기. 이는 스토리지 크기에 포함되지 않습니다. 이 메모리는 MEMALLOC 매크로를 사용하여 할당되고 덮어 쓰여 사용자 지정 할당기를 제공합니다.
totalFrameRate	UINT64	모든 스트림의 관측된 총 프레임 속도.
totalTransferRate	UINT64	초당 바이트 단위의 모든 스트림의 관측된 총 스트림 속도.

StreamMetrics

StreamMetrics 객체를 호출하면 채워집니다 `getKinesisVideoMetrics`.

멤버 필드

필드	데이터 유형	설명
<code>version</code>	UINT32	구조의 버전은 <code>STREAM_METRICS_CURRENT_VERSION</code> 매크로에 정의되어 있습니다.
<code>currentViewDuration</code>	UINT64	축적된 프레임 지속 시간. 고속 네트워킹의 경우 이 지속 시간은 0 또는 프레임 지속 시간 (프레임 전송 시)입니다. 지속 시간이 <code>max_latency</code> 지정한 기간보다 길어지면 스트림 지연 시간 콜백이 지정된 경우 이 콜백이 호출됩니다. <code>StreamDefinition</code> 지속 시간은 PIC 계층의 기본 시간 단위인 100ns 단위로 지정됩니다.
<code>overallViewDuration</code>	UINT64	총 보기 지속 시간. 스트림이 ACK 또는 지속성 없이 구성된 경우 이 값은 프레임이 Kinesis 비디오 스트림에 추가됨에 따라 증가하고 의 값과 같아집니다. <code>buffer_duration</code> <code>StreamDefinition</code> ACK가 활성화되고 지속형 ACK가 수신되면 버퍼는 다음 키프레임으로 트리밍됩니다. 이는 ACK 타임스탬프가 전체 프래그먼트의 시작을 나타내기 때문입니

필드	데이터 유형	설명
		다. 지속 시간은 PIC 계층의 기본 시간 단위인 100-ns 단위로 지정됩니다.
currentViewSize	UINT64	바이트 단위의 현재 버퍼 크기.
overallViewSize	UINT64	바이트 단위의 전체 보기 크기.
currentFrameRate	UINT64	현재 스트림의 관측된 총 프레임 속도.
currentTransferRate	UINT64	초당 바이트 단위의 현재 스트림의 관측된 전송 속도.

프로듀서 SDK 콜백

Amazon Kinesis Video Streams 프로듀서 SDK의 클래스와 메서드는 자체 프로세스를 유지 관리하지 않습니다. 그 대신, 수신 함수 호출 및 이벤트를 사용하여 애플리케이션과 통신할 콜백을 예약합니다.

애플리케이션이 SDK와 상호 작용하는 데 사용할 수 있는 두 개의 콜백 패턴이 있습니다.

- [CallbackProvider](#)— 이 객체는 플랫폼 독립 코드 (PIC) 구성 요소의 모든 콜백을 애플리케이션에 노출합니다. 이 패턴은 전체 기능을 허용하지만, 이 경우 구현은 C++ 계층에 있는 모든 퍼블릭 API 메서드 및 서명을 처리해야 합니다.
- [StreamCallbackProvider](#) 및 [ClientCallbackProvider](#) — 이러한 객체는 스트림별 콜백과 클라이언트별 콜백을 노출하고 SDK의 C++ 계층은 나머지 콜백을 노출합니다. 이것은 생산자 SDK와 상호 작용하기 위한 기본 콜백 패턴입니다.

다음 다이어그램은 콜백 객체의 객체 모델을 보여줍니다.

위의 다이어그램에서 [DefaultCallbackProvider](#)는 [CallbackProvider](#)(PIC에 있는 모든 콜백을 공개)에서 파생되며 [StreamCallbackProvider](#) 및 [ClientCallbackProvider](#)를 포함합니다.

이 주제는 다음 섹션을 포함하고 있습니다.

- [ClientCallbackProvider](#)

- [StreamCallbackProvider](#)
- [ClientCallbacks 구조](#)
- [스트리밍 재시도를 위한 콜백 구현](#)

ClientCallbackProvider

`ClientCallbackProvider` 객체는 클라이언트 수준 콜백 함수를 공개합니다. 이 함수의 세부 정보는 [ClientCallbacks](#) 단원에서 설명합니다.

콜백 메서드:

- `getClientReadyCallback`— 클라이언트에 준비 상태를 보고합니다.
- `getStorageOverflowPressureCallback`— 스토리지 오버플로우 또는 압력을 보고합니다. 스토리지 활용도가 전체 스토리지 크기의 5%인 `STORAGE_PRESSURE_NOTIFICATION_THRESHOLD` 값 밑으로 떨어지면 이 콜백이 호출됩니다. 자세한 정보는 [StorageInfo](#)을 참조하세요.

StreamCallbackProvider

`StreamCallbackProvider` 객체는 스트림 수준 콜백 함수를 공개합니다.

콜백 메서드:

- `getDroppedFragmentReportCallback`: 드롭된 조각을 보고합니다.
- `getDroppedFrameReportCallback`— 프레임 드롭을 보고합니다.
- `getFragmentAckReceivedCallback`— 스트림에 대해 프래그먼트 ACK가 수신되었음을 보고합니다.
- `getStreamClosedCallback`— 스트림이 종료된 상태를 보고합니다.
- `getStreamConnectionStaleCallback`— 부실 연결 상태를 보고합니다. 이 상태에서는 생산자가 서비스에 데이터를 보내고 있지만 승인을 받지 못하는 것입니다.
- `getStreamDataAvailableCallback`— 스트림에서 데이터를 사용할 수 있다고 보고합니다.
- `getStreamErrorReportCallback`— 스트림 오류 상태를 보고합니다.
- `getStreamLatencyPressureCallback`— 누적 버퍼 크기가 `max_latency` 값보다 큰 경우인 스트림 지연 시간을 보고합니다. 자세한 정보는 [StreamDefinition/StreamInfo](#)을 참조하세요.
- `getStreamReadyCallback`: —스트림 준비 상태를 보고합니다.

- `getStreamUnderflowReportCallback`— 스트림 언더플로우 상태를 보고합니다. 이 함수는 현재 사용되지 않으며 나중에 사용할 수 있도록 예약되어 있습니다.

의 소스 코드는 `StreamCallbackProvider` [StreamCallbackProvider.h](#)를 참조하십시오.

ClientCallbacks 구조

ClientCallbacks 구조에는 특정 이벤트가 발생할 때 PIC가 호출하는 콜백 함수 엔트리 포인트가 포함됩니다. 구조에는 `CALLBACKS_CURRENT_VERSION` 필드에 있는 버전 정보, 및 개별 콜백 함수를 통해 반환되는 사용자 정의 데이터에 대한 `customData` 필드도 포함됩니다.

다음 코드 예와 같이, 클라이언트 애플리케이션은 `this` 포인터를 `custom_data` 필드에 사용하여 멤버 함수를 정적 ClientCallback 함수에 실행 시간에 매핑합니다.

```
STATUS TestStreamCallbackProvider::streamClosedHandler(UINT64 custom_data,
    STREAM_HANDLE stream_handle, UINT64 stream_upload_handle) {
    LOG_INFO("Reporting stream stopped.");

    TestStreamCallbackProvider* streamCallbackProvider =
        reinterpret_cast<TestStreamCallbackProvider*> (custom_data);
    streamCallbackProvider->streamClosedHandler(...);
```

이벤트

함수	설명	유형
CreateDeviceFunc	백엔드에서 현재 구현되지 않습니다. Java 또는 C++에서 호출되면 이 호출이 실패합니다. 다른 클라이언트는 플랫폼별 초기화를 수행합니다.	백엔드 API
CreateStreamFunc	스트림이 생성될 때 호출됩니다.	백엔드 API
DescribeStreamFunc	DescribeStream 이 호출될 때 호출됩니다.	백엔드 API
GetStreamingEndpointFunc	GetStreamingEndpoint 이 호출될 때 호출됩니다.	백엔드 API

함수	설명	유형
GetStreamingTokenFunc	GetStreamingToken 이 호출될 때 호출됩니다.	백엔드 API
PutStreamFunc	PutStream 이 호출될 때 호출됩니다.	백엔드 API
TagResourceFunc	TagResource 이 호출될 때 호출됩니다.	백엔드 API
CreateMutexFunc	동기화 뮤텁스를 생성합니다.	동기화
FreeMutexFunc	뮤텁스를 비웁니다.	동기화
LockMutexFunc	동기화 뮤텁스를 잠급니다.	동기화
TryLockMutexFunc	뮤텁스를 잠그려고 시도합니다. 현재 구현되지 않습니다.	동기화
UnlockMutexFunc	뮤텁스 잠금을 해제합니다.	동기화
ClientReadyFunc	클라이언트가 준비 상태에 들어갈 때 호출됩니다.	Notification
DroppedFrameReportFunc	프레임이 드롭될 때 보고합니다.	Notification
DroppedFragmentReportFunc	조각이 드롭될 때 보고합니다. 이 함수는 현재 사용되지 않으며 나중에 사용할 수 있도록 예약되어 있습니다.	Notification
FragmentAckReceivedFunc	조각 ACK(버퍼링, 수신, 지속 및 오류)가 수신될 때 호출됩니다.	Notification

함수	설명	유형
StorageOverflowPressureFunc	스토리지 활용도가 전체 스토리지 크기의 5%로 정의되는 STORAGE_PRESSURE_NOTIFICATION_THRESHOLD 값 밑으로 떨어지면 호출됩니다.	Notification
StreamClosedFunc	나머지 프레임의 마지막 비트가 스트리밍될 때 호출됩니다.	Notification
StreamConnectionStaleFunc	스트림이 기한 경과 연결 상태에 들어갈 때 호출됩니다. 이 조건에서 생산자는 서비스에 데이터를 전송하지만 승인을 받지 않습니다.	Notification
StreamDataAvailableFunc	스트림 데이터를 사용할 수 있을 때 호출됩니다.	Notification
StreamErrorReportFunc	스트림 오류가 발생할 경우 호출됩니다. 스트림이 이 조건에 있으면 PIC는 스트림을 자동으로 닫습니다.	Notification
StreamLatencyPressureFunc	스트림이 지연 시간 조건에 들어갈 때 호출됩니다. 이 조건은 누적된 버퍼 크기가 max_latency 값보다 큰 경우입니다. 자세한 정보는 StreamDefinition/StreamInfo 를 참조하세요.	Notification
StreamReadyFunc	스트림이 준비 상태에 들어갈 때 호출됩니다.	Notification

함수	설명	유형
StreamUnderflowReportFunc	이 함수는 현재 사용되지 않으며 나중에 사용할 수 있도록 예약되어 있습니다.	Notification
DeviceCertToTokenFunc	연결 인증서를 토큰으로 반환합니다.	플랫폼 통합
GetCurrentTimeFunc	현재 시간을 반환합니다.	플랫폼 통합
GetDeviceCertificateFunc	디바이스 인증서를 반환합니다. 이 함수는 현재 사용되지 않으며 나중에 사용할 수 있도록 예약되어 있습니다.	플랫폼 통합
GetDeviceFingerprintFunc	디바이스 지문을 반환합니다. 이 함수는 현재 사용되지 않으며 나중에 사용할 수 있도록 예약되어 있습니다.	플랫폼 통합
GetRandomNumberFunc	0과 RAND_MAX 사이의 임의 숫자를 반환합니다.	플랫폼 통합
GetSecurityTokenFunc	백엔드 API와 통신하는 함수에 전달된 보안 토큰을 반환합니다. 구현은 직렬화된 AccessKeyId , SecretKeyId 및 세션 토큰을 지정할 수 있습니다.	플랫폼 통합
LogPrintFunc	태그와 로그 수준을 사용하여 텍스트 줄을 기록합니다. 자세한 정보는 PlatformUtils.h 을 참조하세요.	플랫폼 통합

위의 표에 있는 플랫폼 통합 함수의 경우 마지막 파라미터는 ServiceCallContext 구조이며, 이 구조에는 다음 필드가 있습니다.

- version: 구조의 버전입니다.
- callAfter: 함수를 호출할 때까지의 절대 시간입니다.
- timeout: 작업 제한 시간입니다(100나노초 단위).
- customData: 다시 클라이언트에게 전달할 사용자 정의 값입니다.
- pAuthInfo: 호출에 대한 자격 증명입니다. 자세한 내용은 다음(__AuthInfo) 구조를 참조하십시오.

권한 부여 정보는 __AuthInfo 구조를 사용하여 제공되며, 이 구조는 직렬화된 자격 증명 또는 공급자별 인증 토큰일 수 있습니다. 이 구조에는 다음 필드가 있습니다.

- version: __AuthInfo 구조의 버전입니다.
- type: 자격 증명의 유형(인증서 또는 보안 토큰)을 정의하는 AUTH_INFO_TYPE 값입니다.
- data: 인증 정보가 포함된 바이트 배열입니다.
- size: data 파라미터의 크기입니다.
- expiration: 100나노초 단위로 표시되는 자격 증명의 만료입니다.

스트리밍 재시도를 위한 콜백 구현

Kinesis Video Producer SDK는 콜백 기능을 통해 스트리밍 상태를 알려 줍니다. 스트리밍 중에 발생하는 일시적인 네트워크 문제를 복구하려면 다음 콜백 메커니즘을 구현하는 것이 좋습니다.

- 스트림 지연 압력 콜백 - 이 콜백 메커니즘은 SDK에서 스트림 지연 조건이 발생할 때 시작됩니다. 누적된 버퍼 크기가 MAX_LATENCY 값보다 커지면 이러한 현상이 발생합니다. 스트리밍 애플리케이션은 스트림을 생성할 때 MAX_LATENCY를 기본값인 60초로 설정합니다. 이 콜백의 일반적인 구현 방법은 연결을 재설정하는 것입니다. 필요에 따라 [https://github.com/aws-labs/amazon-kinesis-video-streams - producer-sdk-cpp /blob/master/ kinesis-video-c-producer /src/source/](https://github.com/aws-labs/amazon-kinesis-video-streams-producer-sdk-cpp/blob/master/kinesis-video-c-producer/src/source/.c) .c에서 샘플 구현을 사용할 수 있습니다. StreamLatencyStateMachine 단, 네트워크 중단으로 인해 전송되지 않은 프레임을 보조 스토리지에 저장하여 백필할 수 있는 옵션은 없습니다.
- 스트림 부실 콜백 - 이 콜백은 생산자가 Amazon Kinesis Data Streams 서비스 (업링크) 로 데이터를 보낼 수는 있지만 승인 (버퍼링된 ACK) 을 제시기에 가져올 수 없을 때 시작됩니다 (기본값은 60초). 네트워크 설정에 따라 스트림 지연 압력 콜백이나 스트림 지연 콜백 또는 둘 다 시작될 수 있습니다. 스트림 지연 시간 압력 콜백의 재시도 구현과 마찬가지로, 일반적인 구현 방법은 연결을 재설정하고

스트리밍을 위한 연결을 새로 시작하는 것입니다. 필요에 따라 <https://github.com/aws-labs/amazon-kinesis-video-streams-ConnectionStateStateMachine-producer-c/blob/master/src/source/.c>에서 샘플 구현을 사용할 수 있습니다.

- 스트림 오류 콜백 - 이 콜백은 KVS API 서비스 호출 호출 중에 SDK가 네트워크 연결 제한 시간 초과나 기타 오류를 발견하면 시작됩니다.
- 프레임 콜백 삭제 - 이 콜백은 네트워크 속도가 느리거나 스트림 오류로 인해 스토리지 크기가 가득 차면 시작됩니다. 네트워크 속도로 인해 프레임이 손실되는 경우 네트워크 속도에 맞게 스토리지 크기를 늘리거나 비디오 프레임 크기를 줄이거나 프레임 속도를 줄일 수 있습니다.

Kinesis 비디오 스트림 파서 라이브러리

Kinesis 비디오 스트림 파서 라이브러리는 Java 애플리케이션에서 Kinesis 비디오 스트림의 MKV 데이터를 사용하는 데 사용할 수 있는 도구 세트입니다.

이 라이브러리에는 다음 도구가 포함됩니다.

- [StreamingMkvReader](#): 이 클래스는 비디오 스트림에서 지정된 MKV 요소를 읽습니다.
- [FragmentMetadataVisitor](#): 이 클래스는 조각의 메타데이터를 검색하고(미디어 조각) 추적합니다(오디오 또는 자막과 같은 미디어 정보를 포함하는 개별 데이터 스트림).
- [OutputSegmentMerger](#): 이 클래스는 연속 조각 또는 청크를 한 비디오 스트림에 병합합니다.
- [KinesisVideoExample](#): Kinesis 비디오 스트림 파서 라이브러리를 사용하는 방법을 보여주는 샘플 애플리케이션입니다.

이 라이브러리에는 도구가 사용되는 방법을 보여 주는 테스트도 포함됩니다.

절차: Kinesis 비디오 스트림 파서 라이브러리 사용

이 절차에는 다음 단계가 포함됩니다.

- [the section called “1단계: 코드 다운로드 및 구성”](#).
- [the section called “2단계: 코드 작성 및 검토”](#).
- [the section called “3단계: 코드 실행 및 확인”](#).

사전 조건

Kinesis 비디오 스트림 파서 라이브러리를 검토하고 사용하려면 다음이 있어야 합니다.

- Amazon Web Services (AWS) 계정. 아직 계정이 AWS 계정없는 경우 을 참조하십시오 [the section called “가입하여 AWS 계정”](#).
- [이클립스 자바 네온 또는 JetBrains 인텔리제이 아이디어와 같은 자바 통합 개발 환경 \(IDE\)](#)
- 자바 11 (예: [아마존 코레토 11](#))

1단계: 코드 다운로드 및 구성

이 단원에서는 Java 라이브러리 및 테스트 코드를 다운로드하고 프로젝트를 Java IDE로 가져옵니다.

사전 조건 및 이 절차에 관한 기타 세부 정보는 [스트림 구문 분석기 라이브러리](#)를 참조하십시오.

1. 디렉터리를 생성하고 GitHub 리포지토리 (<https://github.com/aws/amazon-kinesis-video-streams-parser-library>) 에서 라이브러리 소스 코드를 복제합니다.

```
git clone https://github.com/aws/amazon-kinesis-video-streams-parser-library
```

2. 사용 중인 자바 IDE (예: [Eclipse](#) 또는 [IntelliJ](#) IDEA) 를 열고 다운로드한 Apache Maven 프로젝트를 가져옵니다.

- Eclipse에서 [File], [Import], [Maven], [Existing Maven Projects]를 차례로 선택하여 kinesis-video-streams-parser-lib 폴더로 이동합니다.
- IntelliJ Idea에서 [Import]를 선택합니다. 다운로드한 패키지의 루트에 있는 pom.xml 파일을 찾습니다.

자세한 내용은 관련 IDE 문서를 참조하십시오.

다음 단계

[the section called “2단계: 코드 작성 및 검토”](#).

2단계: 코드 작성 및 검토

이 단원에서는 Java 라이브러리 및 테스트 코드를 검사하고 자체 코드에 있는 라이브러리에서 도구를 사용하는 방법을 배웁니다.

Kinesis 비디오 스트림 파서 라이브러리에는 다음 도구가 포함되어 있습니다.

- [StreamingMkvReader](#)
- [FragmentMetadataVisitor](#)
- [OutputSegmentMerger](#)
- [KinesisVideoExample](#)

StreamingMkvReader

이 클래스는 비차단 방식으로 스트림에서 지정된 MKV 요소를 읽습니다.

다음 코드 예제(FragmentMetadataVisitorTest)는 Streaming MkvReader를 생성하고 사용하여 입력 스트림(inputStream)에서 MkvElement 객체를 검색하는 방법을 보여 줍니다.

```
StreamingMkvReader mkvStreamReader =
    StreamingMkvReader.createDefault(new
InputStreamParserByteSource(inputStream));
while (mkvStreamReader.mightHaveNext()) {
    Optional<MkvElement> mkvElement = mkvStreamReader.nextIfAvailable();
    if (mkvElement.isPresent()) {
        mkvElement.get().accept(fragmentVisitor);
        ...
    }
}
```

FragmentMetadataVisitor

이 클래스는 프래그먼트 (미디어 요소) 에 대한 메타데이터를 검색하고 코덱 전용 데이터, 픽셀 너비 또는 픽셀 높이와 같은 미디어 정보를 포함하는 개별 데이터 스트림을 추적합니다.

다음 코드 예제(FragmentMetadataVisitorTest 파일)는 FragmentMetadataVisitor를 사용하여 MkvElement 객체로부터 데이터를 검색하는 방법을 보여 줍니다.

```
FragmentMetadataVisitor fragmentVisitor = FragmentMetadataVisitor.create();
StreamingMkvReader mkvStreamReader =
    StreamingMkvReader.createDefault(new InputStreamParserByteSource(in));
int segmentCount = 0;
while(mkvStreamReader.mightHaveNext()) {
    Optional<MkvElement> mkvElement = mkvStreamReader.nextIfAvailable();
    if (mkvElement.isPresent()) {
        mkvElement.get().accept(fragmentVisitor);
        if
(MkvTypeInfos.SIMPLEBLOCK.equals(mkvElement.get().getElementMetaData().getTypeInfo()))
{
            MkvDataElement dataElement = (MkvDataElement) mkvElement.get();
            Frame frame =
((MkvValue<Frame>)dataElement.getValueCopy()).getVal();
```



```

                MkvTrackMetadata trackMetadata =
fragmentVisitor.getMkvTrackMetadata(frame.getTrackNumber());
                assertTrackAndFragmentInfo(fragmentVisitor, frame, trackMetadata);
            }
            if
(MkvTypeInfos.SEGMENT.equals(mkvElement.get().getElementMetadata().getTypeInfo())) {
                if (mkvElement.get() instanceof MkvEndMasterElement) {
                    if (segmentCount < continuationTokens.size()) {
                        Optional<String> continuationToken =
fragmentVisitor.getContinuationToken();
                        Assert.assertTrue(continuationToken.isPresent());
                        Assert.assertEquals(continuationTokens.get(segmentCount),
continuationToken.get());
                    }
                    segmentCount++;
                }
            }
        }
    }
}
}
}
}

```

앞선 예제는 다음과 같은 코딩 패턴을 보입니다.

- 데이터 구문 분석을 위한 `FragmentMetadataVisitor`와 데이터 제공을 위한 [StreamingMkvReader](#)를 생성합니다.
- 스트림에 있는 각 `MkvElement`에 대해 메타데이터가 유형 `SIMPLEBLOCK`인지 테스트합니다.
- 이 경우 `MkvElement`에서 `MkvDataElement`를 검색합니다.
- `MkvDataElement`에서 `Frame`(미디어 데이터)을 검색합니다.
- `FragmentMetadataVisitor`에서 `Frame`의 `MkvTrackMetadata`를 검색합니다.
- `Frame` 및 `MkvTrackMetadata` 객체로부터 다음 데이터를 검색하고 확인합니다.
 - 트랙 번호.
 - 프레임의 픽셀 높이.
 - 프레임의 픽셀 넓이.
 - 프레임 인코딩에 사용되는 코덱의 코덱 ID.
 - 이 프레임이 순서대로 도착했는지 여부. 이전 프레임 (있는 경우) 의 트랙 번호가 현재 프레임의 트랙 번호보다 작은지 확인하십시오.

프로젝트에서 `FragmentMetadataVisitor`를 사용하려면, `MkvElement` 객체를 `accept` 메서드를 사용하여 방문자에 전달합니다

```
mkvElement.get().accept(fragmentVisitor);
```

OutputSegmentMerger

이 클래스는 스트림의 여러 트랙에서 취합되는 메타데이터를 단일 세그먼트로 병합합니다.

다음 코드 예제(`FragmentMetadataVisitorTest` 파일)는 `OutputSegmentMerger`를 사용하여 `inputBytes` 바이트 어레이의 트랙 메타데이터를 병합하는 방법을 보여 줍니다.

```
FragmentMetadataVisitor fragmentVisitor = FragmentMetadataVisitor.create();

ByteArrayOutputStream outputStream = new ByteArrayOutputStream();

OutputSegmentMerger outputSegmentMerger =
    OutputSegmentMerger.createDefault(outputStream);

CompositeMkvElementVisitor compositeVisitor =
    new TestCompositeVisitor(fragmentVisitor, outputSegmentMerger);

final InputStream in = TestResourceUtil.getTestInputStream("output_get_media.mkv");

StreamingMkvReader mkvStreamReader =
    StreamingMkvReader.createDefault(new InputStreamParserByteSource(in));

while (mkvStreamReader.mightHaveNext()) {
    Optional<MkvElement> mkvElement = mkvStreamReader.nextIfAvailable();
    if (mkvElement.isPresent()) {
        mkvElement.get().accept(compositeVisitor);
        if
(MkvTypeInfos.SIMPLEBLOCK.equals(mkvElement.get().getElementMetaData().getTypeInfo()))
        {
            MkvDataElement dataElement = (MkvDataElement) mkvElement.get();
            Frame frame = ((MkvValue<Frame>) dataElement.getValueCopy()).getVal();
            Assert.assertTrue(frame.getFrameData().limit() > 0);
            MkvTrackMetadata trackMetadata =
            fragmentVisitor.getMkvTrackMetadata(frame.getTrackNumber());
            assertTrackAndFragmentInfo(fragmentVisitor, frame, trackMetadata);
        }
    }
}
```

앞선 예제는 다음과 같은 코딩 패턴을 보입니다.

- 스트림에서 메타데이터를 검색하기 위해 [FragmentMetadataVisitor](#)를 생성합니다.
- 출력 스트림을 생성하여 병합된 메타데이터를 수신합니다.
- `OutputSegmentMerger`에 통과하는 `ByteArrayOutputStream`를 생성합니다.
- 두 방문자를 포함한 `CompositeMkvElementVisitor`를 생성합니다.
- 지정 파일을 가리키는 `InputStream`을 생성합니다.
- 입력 데이터에 있는 각 요소를 출력 스트림에 병합합니다.

KinesisVideoExample

Kinesis 비디오 스트림 파서 라이브러리를 사용하는 방법을 보여주는 샘플 애플리케이션입니다.

이 클래스는 다음 작업을 수행합니다.

- Kinesis 비디오 스트림을 생성합니다. 주어진 이름의 스트림이 이미 존재하는 경우, 해당 스트림은 삭제되고 다시 생성됩니다.
- Kinesis 비디오 [PutMedia](#) 스트림으로 비디오 프래그먼트를 스트리밍하기 위해 호출합니다.
- Kinesis 비디오 스트림에서 비디오 프래그먼트를 [GetMedia](#) 스트리밍하도록 호출합니다.
- [StreamingMkvReader](#)을 사용하여 스트림에서 반환되는 조각을 구문 분석하고, [FragmentMetadataVisitor](#)를 사용하여 조각을 로깅합니다.

스트림 삭제 및 재생성

다음 코드 예제 (`StreamOps.java`파일에서) 는 지정된 Kinesis 비디오 스트림을 삭제합니다.

```
//Delete the stream
amazonKinesisVideo.deleteStream(new
    DeleteStreamRequest().withStreamARN(streamInfo.get().getStreamARN()));
```

다음 코드 예제 (`StreamOps.java`파일에서) 는 지정된 이름의 Kinesis 비디오 스트림을 생성합니다.

```
amazonKinesisVideo.createStream(new CreateStreamRequest().withStreamName(streamName)
    .withDataRetentionInHours(DATA_RETENTION_IN_HOURS)
    .withMediaType("video/h264"));
```

호출 PutMedia

다음 코드 예제 (PutMediaWorker.java파일에서 가져온) 는 [PutMedia](#) 스트림에서 호출합니다.

```
putMedia.putMedia(new PutMediaRequest().withStreamName(streamName)
    .withFragmentTimecodeType(FragmentTimecodeType.RELATIVE)
    .withProducerStartTimestamp(new Date())
    .withPayload(inputStream), new PutMediaAckResponseHandler() {
        ...
    });
```

통화 GetMedia

다음 코드 예제 (GetMediaWorker.java파일에서 가져온) 는 [GetMedia](#) 스트림에서 호출합니다.

```
GetMediaResult result = videoMedia.getMedia(new
    GetMediaRequest().withStreamName(streamName).withStartSelector(startSelector));
```

결과 파싱하기 GetMedia

이 단원에서는 [StreamingMkvReader](#), [FragmentMetadataVisitor](#)와 [CompositeMkvElementVisitor](#)를 사용하여 GetMedia에서 반환되는 데이터를 구문 분석하고, 파일에 저장하고, 로깅하는 방법을 설명합니다.

의 GetMedia 출력을 읽어 보세요. [StreamingMkvReader](#)

다음 코드 예제 (GetMediaWorker.java파일로부터) 는 a를 [StreamingMkvReader](#) 만들고 이를 사용하여 [GetMedia](#) 작업 결과를 파싱합니다.

```
StreamingMkvReader mkvStreamReader = StreamingMkvReader.createDefault(new
    InputStreamParserByteSource(result.getPayload()));
log.info("StreamingMkvReader created for stream {}", streamName);
try {
    mkvStreamReader.apply(this.elementVisitor);
} catch (MkvElementVisitException e) {
    log.error("Exception while accepting visitor {}", e);
}
```

앞의 코드 예제에서 [StreamingMkvReader](#)는 GetMedia 결과의 페이로드에서 MKVElement 객체를 검색합니다. 다음 단원에서는 이 요소들이 [FragmentMetadataVisitor](#)로 전달됩니다.

를 사용하여 프래그먼트를 검색하십시오 `FragmentMetadataVisitor`

다음 코드 예제(`KinesisVideoExample.java` 및 `StreamingMkvReader.java` 파일의 예제)는 [FragmentMetadataVisitor](#)를 생성합니다. 그런 다음 [StreamingMkvReader](#)에 의해 반복되는 `MkvElement` 객체가 `accept` 메서드를 사용하여 방문자에게 전달됩니다

`KinesisVideoExample.java`로부터:

```
FragmentMetadataVisitor fragmentMetadataVisitor = FragmentMetadataVisitor.create();
```

`StreamingMkvReader.java`로부터:

```
if (mkvElementOptional.isPresent()) {
    //Apply the MkvElement to the visitor
    mkvElementOptional.get().accept(elementVisitor);
}
```

요소를 로깅하고 파일에 쓰기

다음 코드 예제(`KinesisVideoExample.java` 파일의 예제)는 다음 객체를 생성하여 `GetMediaProcessingArguments` 함수의 반환 값의 일부로 반환합니다.

- 시스템 로그에 쓰는 `LogVisitor`(`MkvElementVisitor`의 확장).
- 수신 데이터를 MKV 파일에 쓰는 `OutputStream`.
- `OutputStream`으로 가는 데이터를 버퍼링하는 `BufferedOutputStream`.
- `GetMedia` 결과의 연속적 요소들을 동일 트랙 및 EBML 데이터와 병합하는 [the section called "OutputSegmentMerger"](#).
- [FragmentMetadataVisitor](#)[the section called "OutputSegmentMerger"](#), 및 를 단일 요소 `LogVisitor` 방문자로 `CompositeMkvElementVisitor` 구성하는 A.

```
//A visitor used to log as the GetMedia stream is processed.
LogVisitor logVisitor = new LogVisitor(fragmentMetadataVisitor);

//An OutputSegmentMerger to combine multiple segments that share track and ebml
metadata into one
//mkv segment.
```

```

OutputStream fileOutputStream =
Files.newOutputStream(Paths.get("kinesis_video_example_merged_output2.mkv"),
    StandardOpenOption.WRITE, StandardOpenOption.CREATE);
BufferedOutputStream outputStream = new BufferedOutputStream(fileOutputStream);
OutputSegmentMerger outputSegmentMerger =
OutputSegmentMerger.createDefault(outputStream);

//A composite visitor to encapsulate the three visitors.
CompositeMkvElementVisitor mkvElementVisitor =
    new CompositeMkvElementVisitor(fragmentMetadataVisitor,
outputSegmentMerger, logVisitor);

return new GetMediaProcessingArguments(outputStream, logVisitor,
mkvElementVisitor);

```

그런 다음 미디어 처리 인수가 로 전달되고 이 인수는 로 전달되며 `GetMediaWorker`, 이 인수는 로 전달되며 `ExecutorService`, 이 인수는 별도의 스레드에서 작업자를 수행합니다.

```

GetMediaWorker getMediaWorker = GetMediaWorker.create(getRegion(),
    getCredentialsProvider(),
    getStreamName(),
    new StartSelector().withStartSelectorType(StartSelectorType.EARLIEST),
    amazonKinesisVideo,
    getMediaProcessingArgumentsLocal.getMkvElementVisitor());
executorService.submit(getMediaWorker);

```

다음 단계

[the section called “3단계: 코드 실행 및 확인”](#)

3단계: 코드 실행 및 확인

Kinesis 비디오 스트림 파서 라이브러리에는 자체 프로젝트에서 사용할 수 있는 도구가 포함되어 있습니다. 프로젝트에는 사용자가 설치를 확인할 때 실행할 수 있는 도구의 단위 테스트가 포함됩니다.

다음 유닛 테스트가 라이브러리에 포함됩니다.

- `mkv`
 - `ElementSizeAndOffsetVisitorTest`
 - `MkvValueTest`

- `StreamingMkvReaderTest`
- 유틸리티
 - `FragmentMetadataVisitorTest`
 - `OutputSegmentMergerTest`

아마존 키네시스 비디오 스트림 예제

다음 코드 예제는 Kinesis Video Streams API를 사용하는 방법을 보여줍니다.

예: Kinesis Video Streams로 데이터 전송

- [예: Kinesis Video Streams 프로듀서 SDK GStreamer 플러그인 - kvssink](#): GStreamer 대상으로 사용할 Kinesis Video Streams 프로듀서 SDK를 빌드하는 방법을 보여 줍니다.
- [도커 컨테이너에서 GStreamer 요소를 실행합니다.](#): 사전 빌드된 Docker 이미지를 사용하여 IP 카메라에서 Kinesis Video Streams로 실시간 스트리밍 프로토콜 (RTSP) 비디오를 전송하는 방법을 보여 줍니다.
- [예: RTSP 소스에서의 스트리밍](#): 자체 Docker 이미지를 구축하고 IP 카메라의 RTSP 비디오를 Kinesis Video Streams로 전송하는 방법을 보여 줍니다.
- [예: API를 사용하여 PutMedia Kinesis Video Streams에 데이터 전송](#): 를 사용하여 API를 사용하여 이미 컨테이너 형식 (MKV) 으로 되어 있는 Kinesis Video Streams로 데이터를 보내는 방법을 보여 줍니다. [Java 생산자 라이브러리 사용하기 PutMedia](#)

예: Kinesis Video Streams에서 데이터 검색

- [KinesisVideoExample](#): Kinesis Video Streams 파서 라이브러리를 사용하여 비디오 프래그먼트를 파싱하고 로깅하는 방법을 보여 줍니다.
- [예: Kinesis Video Streams 프래그먼트의 파싱 및 렌더링: JCodec과 JFrame을 사용하여 Kinesis 비디오 스트림 프래그먼트를 파싱하고 렌더링하는 방법을 보여 줍니다.](#)

예: 비디오 데이터 재생

- [예: HTML에서 HLS를 사용하고 JavaScript](#): Kinesis 비디오 스트림의 HLS 스트리밍 세션을 검색하고 웹 페이지에서 재생하는 방법을 보여 줍니다.

사전 조건

- 샘플 코드에서는 자격 증명 프로필 파일에 설정한 프로필을 AWS 지정하거나 통합 개발 환경 (IDE) 의 Java 시스템 속성에 자격 증명을 제공하여 자격 증명을 제공합니다. 아직 자격 증명을 설정하지

않았다면 먼저 자격 증명을 설정하십시오. 자세한 내용은 [AWS 자격 증명 및 개발 지역 설정을 참조](#) 하십시오.

- Java IDE를 사용하여 코드를 보고 실행하는 것이 좋습니다(예: 다음 중 하나).
 - [Eclipse Java Neon](#)
 - [JetBrains 인텔리제이 아이디어](#)

예: Kinesis Video Streams 프로듀서 SDK GStreamer 플러그인 - kvssink

이 주제에서는 GStreamer 플러그인으로 사용할 Amazon Kinesis Video Streams 프로듀서 SDK를 빌드하는 방법을 설명합니다.

주제

- [GStreamer 요소를 다운로드, 빌드 및 구성합니다.](#)
- [GStreamer 요소를 실행하세요.](#)
- [GStreamer 실행 명령어 예시](#)
- [도커 컨테이너에서 GStreamer 요소를 실행합니다.](#)
- [GStreamer 요소 파라미터 레퍼런스](#)

[GStreamer](#)는 여러 카메라 및 비디오 소스에서 모듈식 플러그인을 결합하여 사용자 지정 미디어 파이프라인을 생성하는 데 사용하는 인기 있는 미디어 프레임워크입니다. Kinesis Video Streams GStreamer 플러그인은 기존 GStreamer 미디어 파이프라인과 Kinesis Video Streams의 통합을 간소화합니다. GStreamer를 통합한 후 웹캠 또는 실시간 스트리밍 프로토콜 (RTSP) 카메라의 비디오를 Kinesis Video Streams로 스트리밍하여 실시간 또는 이후 재생, 저장 및 추가 분석을 수행할 수 있습니다.

GStreamer 플러그인은 Kinesis Video Streams Producer SDK에서 제공하는 기능을 GStreamer 싱크 요소에 캡슐화하여 Kinesis Video Streams로의 비디오 스트림 전송을 자동으로 관리합니다. kvssink GStreamer 프레임워크는 추가 처리, 렌더링 또는 저장을 위해 카메라와 같은 디바이스 또는 기타 비디오 소스의 미디어 흐름을 구성하기 위한 표준 관리형 환경을 제공합니다.

GStreamer 파이프라인은 일반적으로 소스(비디오 카메라)와 싱크 요소(비디오를 렌더링하기 위한 플레이어 또는 오프라인 검색을 위한 스토리지) 간의 링크로 구성됩니다. 이 예제에서는 생산자 SDK 요소를 싱크 또는 미디어 대상으로 비디오 소스(웹캠 또는 IP 카메라)에 사용합니다. 그런 다음 SDK를 캡슐화하는 플러그인 요소가 비디오 스트림을 Kinesis Video Streams로 보냅니다.

이 항목에서는 일반적으로 중간 인코딩 단계 (H.264 인코딩 사용) 를 통해 Kinesis Video Streams 로 연결되는 웹 카메라 또는 RTSP 스트림과 같은 비디오 소스에서 비디오를 스트리밍할 수 있는 GStreamer 미디어 파이프라인을 구성하는 방법에 대해 설명합니다. 비디오 스트림을 Kinesis 비디오 스트림으로 사용할 수 있게 되면 Kinesis [Video Streams Parser 라이브러리](#)를 사용하여 [비디오 스트림](#)의 추가 처리, 재생, 저장 또는 분석을 수행할 수 있습니다.

GStreamer 요소를 다운로드, 빌드 및 구성합니다.

GStreamer 플러그인 예제는 Kinesis Video Streams C++ 프로듀서 SDK에 포함되어 있습니다. SDK 사전 조건 및 다운로드에 대한 자세한 내용은 [1단계: C++ 프로듀서 라이브러리 코드 다운로드 및 구성](#) 단원을 참조하십시오.

macOS, Ubuntu, Raspberry Pi 또는 Windows에서 생산자 SDK GStreamer 싱크를 동적 라이브러리로 빌드할 수 있습니다. GStreamer 플러그인은 build 디렉터리에 있습니다. 이 플러그인을 로드하려면 해당 플러그인이 사용자 버전에 있어야 합니다. GST_PLUGIN_PATH 다음 명령을 실행합니다:

```
export GST_PLUGIN_PATH=`pwd`/build
```

Note

macOS에서는 도커 컨테이너에서 GStreamer를 실행할 때만 네트워크 카메라에서 비디오를 스트리밍할 수 있습니다. 도커 컨테이너에서 macOS의 USB 카메라에서 비디오를 스트리밍하는 작업은 지원되지 않습니다.

GStreamer 요소를 실행하세요.

Kinesis Video Streams Producer SDK 요소를 싱크로 사용하여 GStreamer를 실행하려면 명령을 사용합니다. `gst-launch-1.0` GStreamer 플러그인이 사용하기에 적합한 업스트림 요소를 사용하세요. 예를 들어 Linux 시스템의 v4l2 디바이스에는 [v4l2src](#)를 사용하고, RTSP 디바이스에는 [rtspsrc](#)를 사용합니다. 생산자 SDK에 비디오를 전송하기 위한 싱크(파이프라인의 최종 대상)로 `kvssink`를 지정합니다.

[자격 증명을 제공하고 지역을 제공하는](#) 것 외에도 `kvssink` 요소에는 다음과 같은 필수 매개변수가 있습니다.

- `stream-name`— 대상 Kinesis Video Streams의 이름.

kvssink 선택적 파라미터에 대한 자세한 내용은 [GStreamer 요소 파라미터 레퍼런스](#) 단원을 참조하십시오.

[GStreamer 플러그인 및 파라미터에 대한 최신 정보는 GStreamer 플러그인을 참조하십시오.](#)

GStreamer 요소 또는 플러그인 이름을 `gst-inspect-1.0` 뒤에 붙여 해당 정보를 인쇄하고 기기에서 사용할 수 있는지 확인할 수도 있습니다.

```
gst-inspect-1.0 kvssink
```

빌드에 kvssink 실패하거나 GST_PLUGIN_PATH가 제대로 설정되지 않은 경우 출력은 다음과 비슷하게 표시됩니다.

```
No such element or plugin 'kvssink'
```

GStreamer 실행 명령어 예시

다음 예제는 kvssink GStreamer 플러그인을 사용하여 다양한 유형의 기기에서 비디오를 스트리밍하는 방법을 보여줍니다.

예 1: Ubuntu의 RTSP 카메라에서 비디오 스트리밍

다음 명령은 [rtspsrc](#) GStreamer 플러그인을 사용하여 네트워크 RTSP 카메라에서 스트리밍하는 GStreamer 파이프라인을 Ubuntu에서 생성합니다.

```
gst-launch-1.0 -v rtspsrc location="rtsp://YourCameraRtspUrl" short-header=TRUE !
rtph264depay ! h264parse ! kvssink stream-name="YourStreamName" storage-size=128
```

예 2: Ubuntu의 USB 카메라에서 비디오 인코딩 및 스트리밍

다음 명령은 USB 카메라의 스트림을 H.264 형식으로 인코딩하고 Kinesis Video Streams로 스트리밍하는 GStreamer 파이프라인을 Ubuntu에 생성합니다. [이 예제에서는 v4l2src GStreamer 플러그인을 사용합니다.](#)

```
gst-launch-1.0 v4l2src do-timestamp=TRUE device=/dev/video0 ! videoconvert ! video/x-raw,format=I420,width=640,height=480,framerate=30/1 ! x264enc bframes=0 key-int-max=45
bitrate=500 ! video/x-h264,stream-format=avc,alignment=au,profile=baseline ! kvssink
stream-name="YourStreamName" storage-size=512 access-key="YourAccessKey" secret-key="YourSecretKey" aws-region="YourAWSRegion"
```

예 3: Ubuntu의 USB 카메라에서 사전 인코딩된 비디오를 스트리밍합니다.

다음 명령은 카메라가 이미 H.264 형식으로 인코딩한 비디오를 Kinesis Video Streams로 스트리밍하는 GStreamer 파이프라인을 Ubuntu에 생성합니다. [이 예제에서는 v4l2src GStreamer 플러그인을 사용합니다.](#)

```
gst-launch-1.0 v4l2src do-timestamp=TRUE device=/dev/video0 ! h264parse ! video/x-h264,stream-format=avc,alignment=au ! kvssink stream-name="plugin" storage-size=512 access-key="YourAccessKey" secret-key="YourSecretKey" aws-region="YourAWSRegion"
```

예 4: macOS의 네트워크 카메라에서 비디오 스트리밍

다음 명령은 네트워크 카메라에서 Kinesis Video Streams로 비디오를 스트리밍하는 GStreamer 파이프라인을 macOS에 생성합니다. 이 예제에서는 [rtspsrc](#) GStreamer 플러그인을 사용합니다.

```
gst-launch-1.0 rtspsrc location="rtsp://YourCameraRtspUrl" short-header=TRUE ! rtph264depay ! h264parse ! video/x-h264, format=avc,alignment=au ! kvssink stream-name="YourStreamName" storage-size=512 access-key="YourAccessKey" secret-key="YourSecretKey" aws-region="YourAWSRegion"
```

예 5: Windows의 네트워크 카메라에서 비디오를 스트리밍합니다.

다음 명령은 네트워크 카메라에서 Kinesis Video Streams로 비디오를 스트리밍하는 GStreamer 파이프라인을 Windows에 생성합니다. 이 예제에서는 [rtspsrc](#) GStreamer 플러그인을 사용합니다.

```
gst-launch-1.0 rtspsrc location="rtsp://YourCameraRtspUrl" short-header=TRUE ! rtph264depay ! video/x-h264, format=avc,alignment=au ! kvssink stream-name="YourStreamName" storage-size=512 access-key="YourAccessKey" secret-key="YourSecretKey" aws-region="YourAWSRegion"
```

예 6: 라즈베리 파이의 카메라에서 비디오를 스트리밍합니다.

다음 명령은 비디오를 Kinesis Video Streams로 스트리밍하는 GStreamer 파이프라인을 라즈베리 파이에서 생성합니다. [이 예제에서는 v4l2src GStreamer 플러그인을 사용합니다.](#)

```
gst-launch-1.0 v4l2src do-timestamp=TRUE device=/dev/video0 ! videoconvert ! video/x-raw,format=I420,width=640,height=480,framerate=30/1 ! omxh264enc control-rate=1 target-bitrate=5120000 periodicity-idr=45 inline-header=FALSE ! h264parse ! video/x-h264,stream-format=avc,alignment=au,width=640,height=480,framerate=30/1,profile=baseline ! kvssink
```

```
stream-name="YourStreamName" access-key="YourAccessKey" secret-key="YourSecretKey"
aws-region="YourAWSRegion"
```

예제 7: 라즈베리 파이와 우분투에서 오디오와 비디오를 모두 스트리밍합니다.

[gst-launch-1.0 명령을 실행하여 Raspberry-Pi 및 Ubuntu의 오디오 및 비디오 모두를 스트리밍하기 시작하는 방법을 알아봅니다.](#)

예 8: macOS의 기기 소스에서 오디오와 비디오를 모두 스트리밍

[gst-launch-1.0 명령을 실행하여 MacOS의 오디오 및 비디오 모두를 스트리밍하기 시작하는 방법을 알아봅니다.](#)

예 9: 오디오와 비디오가 모두 포함된 MKV 파일 업로드

[gst-launch-1.0 명령을 실행하여 오디오와 비디오가 모두 포함된 MKV 파일을 업로드하는 방법을 알아봅니다.](#) h.264 및 AAC 인코딩 미디어가 포함된 MKV 테스트 파일이 필요합니다.

도커 컨테이너에서 GStreamer 요소를 실행합니다.

도커는 컨테이너를 사용하여 애플리케이션을 개발, 배포 및 실행하기 위한 플랫폼입니다. Docker를 사용하여 GStreamer 파이프라인을 생성하면 Kinesis Video Streams의 운영 환경이 표준화되어 애플리케이션 구축 및 사용이 간소화됩니다.

도커를 설치하고 구성하려면 다음을 참조하십시오.

- [도커 다운로드 지침](#)
- [도커 시작하기](#)

Docker를 설치한 후 아래에 제공된 명령 중 하나를 사용하여 Amazon Elastic 컨테이너 레지스트리에서 Kinesis Video Streams C++ 프로듀서 SDK (및 GStreamer 플러그인) 를 다운로드할 수 있습니다.

```
docker pull
```

Kinesis Video Streams Producer SDK 요소를 Docker 컨테이너의 싱크로 사용하여 GStreamer를 실행하려면 다음을 수행하십시오.

주제

- [Docker 클라이언트를 인증하십시오.](#)
- [우분투, macOS, 윈도우 또는 라즈베리 파이용 도커 이미지 다운로드](#)
- [도커 이미지 실행](#)

Docker 클라이언트를 인증하십시오.

이미지를 가져오려는 Amazon ECR 레지스트리에 대해 Docker 클라이언트를 인증합니다. 사용된 각 레지스트리에 대한 인증 토큰을 가져와야 합니다. 토큰은 12시간 동안 유효합니다. 자세한 내용은 Amazon Elastic Container Registry 사용 설명서의 [레지스트리 권한](#)을 참조하세요.

Example : Amazon ECR을 사용하여 인증

Amazon ECR로 인증하려면 표시된 대로 다음 명령을 복사하여 붙여넣습니다.

```
sudo aws ecr get-login-password --region us-west-2 | docker login -u AWS --password-stdin https://546150905175.dkr.ecr.us-west-2.amazonaws.com
```

성공하면 Login Succeeded 출력이 인쇄됩니다.

우분투, macOS, 윈도우 또는 라즈베리 파이용 도커 이미지 다운로드

운영 체제에 따라 다음 명령 중 하나를 사용하여 도커 이미지를 도커 환경으로 다운로드합니다.

우분투용 도커 이미지 다운로드

```
sudo docker pull 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-amazon-linux:latest
```

macOS용 도커 이미지 다운로드

```
docker pull 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-amazon-linux:latest
```

윈도우용 도커 이미지 다운로드

```
docker pull 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-amazon-windows:latest
```

라즈베리파이용 도커 이미지 다운로드

```
sudo docker pull 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-raspberry-pi:latest
```

이미지가 성공적으로 추가되었는지 확인하려면 다음 명령을 사용합니다.

```
docker images
```

도커 이미지 실행

운영 체제에 따라 다음 명령 중 하나를 사용하여 도커 이미지를 실행합니다.

우분투에서 도커 이미지 실행

```
sudo docker run -it --network="host" --device=/dev/video0 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-amazon-linux /bin/bash
```

macOS에서 도커 이미지 실행하기

```
sudo docker run -it --network="host" 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-amazon-linux /bin/bash
```

윈도우에서 도커 이미지 실행

```
docker run -it 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-windows AWS_ACCESS_KEY_ID AWS_SECRET_ACCESS_KEY RTSP_URL STREAM_NAME
```

라즈베리파이에서 도커 이미지 실행

```
sudo docker run -it --device=/dev/video0 --device=/dev/vchiq -v /opt/vc:/opt/vc 546150905175.dkr.ecr.us-west-2.amazonaws.com/kinesis-video-producer-sdk-cpp-raspberry-pi /bin/bash
```

Docker는 컨테이너를 시작하고 컨테이너 내에서 명령을 사용하기 위한 명령 프롬프트를 표시합니다.

컨테이너에서 다음 명령을 사용하여 환경 변수를 설정합니다.

```
export LD_LIBRARY_PATH=/opt/awssdk/amazon-kinesis-video-streams-producer-sdk-cpp/kinesis-video-native-build/downloads/local/lib:$LD_LIBRARY_PATH
export PATH=/opt/awssdk/amazon-kinesis-video-streams-producer-sdk-cpp/kinesis-video-native-build/downloads/local/bin:$PATH
export GST_PLUGIN_PATH=/opt/awssdk/amazon-kinesis-video-streams-producer-sdk-cpp/kinesis-video-native-build/downloads/local/lib:$GST_PLUGIN_PATH
```

kvssink를 사용하여 디바이스와 비디오 소스에 적합한 파이프라인을 `gst-launch-1.0` 실행하도록 스트리밍을 시작하세요. 파이프라인의 예는 [을 참조하십시오](#) [GStreamer 실행 명령어 예시](#).

GStreamer 요소 파라미터 레퍼런스

Amazon Kinesis Video Streams 프로듀서 C++ SDK로 비디오를 전송하려면 파이프라인의 싱크 또는 최종 목적지로 *kvssink* 지정합니다. 이 참조는 *kvssink* 필수 및 선택적 파라미터에 대한 정보를 제공합니다. 자세한 정보는 [the section called “스트리머 플러그인 - kvssink”](#)을 참조하세요.

주제

- [the section called “에 자격 증명을 제공하십시오. kvssink”](#)
- [the section called “지역을 제공하십시오. kvssink”](#)
- [the section called “kvssink선택적 파라미터”](#)

에 자격 증명을 제공하십시오. kvssink

kvssinkGStreamer 요소가 요청을 할 수 있도록 하려면 Amazon Kinesis Video Streams 서비스를 호출할 때 사용할 AWS 자격 증명을 제공하십시오. AWS자격 증명 공급자 체인은 다음과 같은 순서로 자격 증명을 찾습니다.

1. AWS IoT 자격 증명

AWS IoT 자격 증명을 설정하려면 [the section called “를 사용하여 Kinesis Video Streams 리소스에 대한 액세스 제어 AWS IoT”](#)을 참조하십시오.

```
iot-credentials##### ## ## # = # ## #### ### ##### iot-certificate, ##### ## ## ###.
```

키	필수	설명
ca-path	예	TLS를 통해 백엔드 서비스와의 신뢰를 구축하는 데 사용되는 CA 인증서의 파일 경로입니다. Example 예: <code>/file/path/to/certificate.pem</code>
cert-path	예	X.509 인증서의 파일 경로.

키	필수	설명
		<p>Example</p> <p>예: <code>/file/path/to/certificateID -certificate.pem.crt</code></p>
endpoint	예	<p>계정의 AWS IoT Core 자격 증명 엔드포인트 공급자 엔드포인트. AWS AWS IoT 개발자 안내서를 참조하십시오.</p> <p>Example</p> <p>예: <code>credential-account-specific-prefix .credentials.iot. aws-region .amazonaws.com</code></p>
key-path	예	<p>공개/개인 키 쌍에서 사용되는 개인 키의 파일 경로입니다.</p> <p>Example</p> <p>예: <code>/file/path/to/certificateID -private.pem.key</code></p>
role-aliases	예	<p>연결할 때 사용할 AWS IAM 역할을 가리키는 역할 별칭의 이름. AWS IoT Core</p> <p>Example</p> <p>예: <code>KvsCameraIoTRoleAlias</code></p>

키	필수	설명
iot-thing-name	아니요	iot-thing-name 은 선택 사항입니다. 제공하지 않으면 iot-thing-name stream-name 파라미터 값이 사용됩니다. Example 예: <i>kvs_example_camera</i>

Example

예:

```
gst-launch-1.0 -v ... ! kvssink stream-name="YourStream" aws-region="YourRegion"
  iot-certificate="iot-certificate,endpoint=credential-account-specific-prefix.credentials.iot.aws-region.amazonaws.com,cert-path=certificateID-certificate.pem.crt,key-path=certificateID-private.pem.key,ca-path=certificate.pem,role-aliases=YourRoleAlias,iot-thing-name=YourThingName"
```

2. 환경 변수

환경의 사용자 자격 증명을 가져오려면 kvssink 다음 환경 변수를 설정하십시오.

환경 변수 이름	필수	설명
AWS_ACCESS_KEY_ID	예	Amazon Kinesis Video Streams에 액세스하는 데 사용되는 액세스 키입니다. AWS
AWS_SECRET_ACCESS_KEY	예	액세스 키와 관련된 AWS 비밀 키.
AWS_SESSION_TOKEN	아니요	AWS STS 운영에서 직접 임시 보안 자격 증명을 사용하는 경우 필요한 세션 토큰 값을 지정합니다.

환경 변수를 설정하면 사용되는 값이 변경되어 웹 세션이 종료될 때까지 또는 변수를 다른 값으로 설정할 때까지 유지됩니다. 향후 세션에서 변수를 계속 사용하려면 셸의 시작 스크립트에서 변수를 설정하세요.

3. access-key, secret-key 매개변수

자격 증명을 kvssink 파라미터로 직접 지정하려면 다음 파라미터를 설정하십시오.

kvssink파라미터 이름	필수	설명
access-key	예	Amazon Kinesis Video Streams에 액세스하는 데 사용되는 액세스 키입니다. AWS
secret-key	예	액세스 키와 관련된 AWS 비밀 키.
session-token	아니요	AWS STS 운영에서 직접 임시 보안 자격 증명을 사용하는 경우 필요한 세션 토큰 값을 지정합니다.

Example

정적 자격 증명 사용:

```
gst-launch-1.0 -v ... ! kvssink stream-name="YourStream" aws-region="YourRegion"
  access-key="AKIDEXAMPLE" secret-key="SKEEXAMPLE"
```

Example

임시 자격 증명 사용:

```
gst-launch-1.0 -v ... ! kvssink stream-name="YourStream" aws-region="YourRegion"
  access-key="AKIDEXAMPLE" secret-key="SKEEXAMPLE" session-token="STEXAMPLE"
```

4. 자격 증명 파일

⚠ Important

이전 방법 중 하나를 선택한 경우 `credential-file kvssink` 파라미터를 사용할 수 없습니다.

kvssink 파라미터 이름	필수	설명
<code>credential-file</code>	예	특정 형식의 자격 증명이 들어 있는 텍스트 파일의 경로입니다.

텍스트 파일에는 다음 형식 중 하나의 자격 증명이 포함되어야 합니다.

- 자격 증명 `YourAccessKeyYourSecretKey`
- 자격 증명 `YourAccessKey## YourSecretKeySessionToken`

Example

예: `credentials.txt` 파일의 위치는 다음과 같습니다. `/home/ubuntu`

```
CREDENTIALS AKIDEXAMPLE 2023-08-10T22:43:00Z SKEXAMPLE STEXAMPLE
```

에서 사용하려면 다음과 `kvssink` 같이 입력합니다.

```
gst-launch-1.0 -v ... ! kvssink stream-name="YourStream" aws-region="YourRegion"
credential-file="/home/ubuntu/credentials.txt"
```

ℹ Note

향후 만료 시간은 $5 + 30 + 3 = 38$ 초 이상이어야 합니다. 유예 기간은 `IOT_CREDENTIAL_FETCH_GRACE_PERIOD` 변수로 정의됩니다 [IotCredentialProvider.h](#). 시작할 `kvssink` 때 자격 증명 만료일에 너무 가까워지면 오류 코드가 나타납니다 `0x52000049 - STATUS_INVALID_TOKEN_EXPIRATION`.

⚠ Important

kvssink 자격 증명 파일을 수정하지 않습니다. 임시 자격 증명을 사용하는 경우 만료 시간에서 유예 기간을 뺀 시간 전에 외부 소스에서 자격 증명 파일을 업데이트해야 합니다.

지역을 제공하십시오. kvssink

지역 조회 순서는 다음과 같습니다.

1. AWS_DEFAULT_REGION 환경 변수를 먼저 검토합니다. 설정된 경우 해당 지역을 사용하여 클라이언트를 구성합니다.
2. aws-region 다음에서 매개변수를 검토합니다. 설정된 경우 해당 지역을 사용하여 클라이언트를 구성합니다.
3. 이전 방법 중 어느 것도 사용하지 않은 경우 kvssink 기본값은 us-west-2입니다.

kvssink 선택적 파라미터

kvssink 요소에는 다음과 같은 선택적 파라미터가 있습니다. 이런 파라미터에 대한 자세한 내용은 [Kinesis 비디오 스트림 구조](#) 섹션을 참조하세요.

파라미터	설명	단위/유형	기본값
stream-name	대상 아마존 Kinesis 비디오 스트림의 이름. ⚠ Important 스트림 이름을 지정하지 않은 경우 기본 스트림 이름인 "DEFAULT_STREAM"이 사용됩니다. 해당 기본 이름을 가진 스		

파라미터	설명	단위/유형	기본값
	<p>트림이 아직 없는 경우 스트림이 생성됩니다.</p>		
absolute-fragment-times	절대 조각 시간을 사용할지 여부입니다.	불	true
access-key	<p>Kinesis Video Streams에 액세스하는 데 사용되는 액세스 키입니다. AWS</p> <p>AWS 자격 증명을 설정하거나 이 파라미터를 제공해야 합니다. 이 정보를 제공하려면 다음을 입력합니다.</p> <pre>export AWS_ACCESS_KEY_ID=</pre>		
avg-bandwidth-bps	스트림용 예상 평균 대역폭.	초당 비트 수	4194304

파라미터	설명	단위/유형	기본값
aws-region	<p>사용 AWS 리전 방법.</p> <div data-bbox="472 302 792 1003" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p>Note</p> <p>지역에 AWS_DEFAULT_REGION 환경 변수를 제공할 수도 있습니다. 환경 변수와 kvssink 매개 변수가 모두 설정된 경우 환경 변수가 우선합니다.</p> </div> <div data-bbox="472 1066 792 1528" style="border: 1px solid #ff9999; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p>Important</p> <p>달리 지정하지 않을 us-west-2 경우 해당 지역의 기본값으로 설정됩니다.</p> </div>	String	"us-west-2"
buffer-duration	스트림 버퍼 지속 시간입니다.	초	120
codec-id	스트림의 코덱 ID입니다.	String	"V_MPEG4/ISO/AVC"

파라미터	설명	단위/유형	기본값
connection-staleness	스트림 비활성 콜백이 호출되는 시점 이후 시간입니다.	초	60
content-type	스트림의 콘텐츠 유형입니다.	String	"video/h264"
fragment-acks	조각 ACK를 사용할지 여부입니다.	불	true
fragment-duration	원하는 조각 지속 시간입니다.	밀리초	2000
framerate	예상 프레임 속도입니다.	초당 프레임	25
frame-timecodes	현재 시간 콜백을 사용하여 프레임 타임코드를 사용할지 타임스탬프를 생성할지 여부.	불	true
key-frame-fragmentation	키 프레임에서 조각을 생성할지 여부입니다.	불	true
log-config	로그 구성 경로입니다.	String	"../kvs_1log_configuration"
max-latency	스트림의 최대 지연 시간입니다.	초	60
recalculate-metrics	지표를 다시 계산할지 여부입니다.	불	true

파라미터	설명	단위/유형	기본값
replay-duration	다시 시작이 활성화된 경우 오류가 발생한 동안 현재 리더를 뒤로 롤하여 다시 재생하는 기간입니다.	초	40
restart-on-error	오류 발생 시 다시 시작할지 여부입니다.	불	true
retention-period	스트림이 보존되는 시간의 길이입니다.	시간	2
rotation-period	키 교체 기간입니다. 자세한 내용은 회전 AWS KMS 키 를 참조하십시오.	초	3600
secret-key	<p>Kinesis Video Streams에 액세스하는 데 사용되는 AWS 비밀 키입니다.</p> <p>AWS 자격 증명을 설정하거나 이 파라미터를 제공해야 합니다.</p> <pre>export AWS_SECRET_ACCESS_KEY=</pre>		
session-token	AWS STS 운영에서 직접 임시 보안 자격 증명을 사용하는 경우 필요한 세션 토큰 값을 지정합니다.		

파라미터	설명	단위/유형	기본값
storage-size	디바이스 스토리지 크기 (MiB) 디바이스 스토리지 구성에 대한 자세한 내용은 StorageInfo 단원을 참조하십시오.	메비바이트 (MiB)	128
streaming-type	스트리밍 유형입니다. 유효한 값으로는 다음이 포함됩니다. <ul style="list-style-type: none"> 0: 실시간 1: 근 실시간(현재 지원되지 않은) 2: 오프라인 	열거형 GstKvsSinkStreamingType	0: 실시간
timecode-scale	MKV 타임코드 척도입니다.	밀리초	1
track-name	MKV 트랙 이름	String	"kinesis_video"

파라미터	설명	단위/유형	기본값
iot-certificate	<p>AWS IoT kvssink요소에 사용할 자격 증명.</p> <p>iot-certificate 다음 키와 값을 허용합니다.</p> <div data-bbox="472 575 792 1226" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> Note</p> <p>iot-thing-name 는 선택 사항입니다. 제공하지 않으면 iot-thing-name 매개변수 값이 사용됩니다.</p> </div> <ul style="list-style-type: none"> • endpoint=iotcredentialsproviderendpoint • cert-path=/localdirectorypath/to/certificate • key-path=/localdirectorypath / 	String	None

파라미터	설명	단위/유형	기본값
	<pre>to/private/ key</pre> <ul style="list-style-type: none"> • <code>ca-path=/localdir/ectorypath/to/ca-cert</code> • <code>role-aliases =role-aliases</code> • <code>iot-thing-name=YourIotThingName</code> 		

예: API를 사용하여 PutMedia Kinesis Video Streams에 데이터 전송

이 예제는 API를 사용하는 방법을 보여줍니다. [PutMedia](#) 이미 컨테이너 형식 (MKV) 으로 되어 있는 데이터를 보내는 방법을 보여줍니다. 전송하기 전에 데이터를 컨테이너 형식으로 조합해야 하는 경우 (예: 카메라 비디오 데이터를 프레임으로 조합하는 경우) 를 참조하십시오. [Kinesis 비디오 스트림 프로듀서 라이브러리](#)

Note

이 PutMedia 작업은 C++ 및 Java SDK에서만 사용할 수 있습니다. 이는 연결, 데이터 흐름 및 승인이 전이중 방식으로 관리되기 때문입니다. 다른 언어에서는 지원되지 않습니다.

이 예제에는 다음 단계가 포함됩니다.

- [1단계: 코드 다운로드 및 구성](#)
- [2단계: 코드 작성 및 검토](#)
- [3단계: 코드 실행 및 확인](#)

1단계: 코드 다운로드 및 구성

단계에 따라 Java 예제 코드를 다운로드하고, 프로젝트를 Java IDE로 가져오고, 라이브러리 위치를 구성하고, AWS 자격 증명을 사용하도록 코드를 구성합니다.

1. 디렉터리를 생성하고 GitHub 리포지토리에서 예제 소스 코드를 복제합니다. PutMedia 예제는 [Java 생산자 라이브러리](#)의 일부입니다.

```
git clone https://github.com/aws-labs/amazon-kinesis-video-streams-producer-sdk-java
```

2. 사용 중인 자바 IDE (예: [Eclipse](#) 또는 [IntelliJ](#) IDEA) 를 열고 다운로드한 Apache Maven 프로젝트를 가져옵니다.
 - Eclipse에서 파일, 가져오기, Maven, Existing Maven Projects(기존 Maven 프로젝트)를 차례로 선택한 후 다운로드한 패키지의 루트로 이동합니다. pom.xml 파일을 선택합니다.
 - IntelliJ Idea에서 [Import]를 선택합니다. 다운로드한 패키지의 루트에 있는 pom.xml 파일을 찾습니다.

자세한 내용은 관련 IDE 문서를 참조하십시오.

3. 가져온 라이브러리를 IDE가 알 수 있도록 프로젝트를 업데이트합니다.
 - IntelliJ IDEA의 경우 다음을 수행합니다.
 - a. 프로젝트의 [lib] 디렉터리의 컨텍스트 메뉴를 열고(오른쪽 버튼 클릭) [Add as library]를 선택합니다.
 - b. 파일을 선택한 다음 프로젝트 구조를 선택합니다.
 - c. [Project Settings]에서 [Modules]를 선택합니다.
 - d. 소스 탭에서 Language Level(언어 레벨)을 7 이상으로 설정합니다.
 - Eclipse의 경우 다음을 수행합니다.
 - a. 프로젝트의 컨텍스트 메뉴를 열고(오른쪽 버튼 클릭) [Properties], [Java Build Path], [Source]를 차례로 선택합니다. 뒤이어 다음과 같이 하십시오.
 1. [Source] 탭에서 [Native library location]을 두 번 클릭합니다.
 2. [Native Library Folder Configuration] 마법사에서, [Workspace]를 선택합니다.
 3. [Native Library Folder] 선택에서 프로젝트에 있는 [lib] 디렉터리를 선택합니다.

- b. 프로젝트에 대한 컨텍스트 메뉴를 열고(마우스 오른쪽 버튼 클릭) [Properties]를 선택합니다. 뒤이어 다음과 같이 하십시오.
 1. [Libraries] 탭에서 [Add Jars]를 선택합니다.
 2. JAR selection(JAR 선택) 마법사에서 프로젝트의 lib 디렉터리에 있는 모든 .jar을 선택합니다.

2단계: 코드 작성 및 검토

PutMedia API 예제()는 다음과 같은 코딩 패턴을 보입니다.PutMediaDemo

주제

- [만들기 PutMediaClient](#)
- [미디어를 스트리밍하고 스레드를 일시 중지합니다.](#)

이 단원의 코드 예제는 PutMediaDemo 클래스가 출처입니다.

만들기 PutMediaClient

PutMediaClient객체를 만들려면 다음 매개변수가 필요합니다.

- PutMedia 엔드포인트의 URI.
- 스트리밍할 MKV 파일로 향하는 InputStream
- 스트림 이름입니다. 이 예제에서는 [Java 생산자 라이브러리 사용하기](#)에서 생성된 스트림을 사용합니다(my-stream). 다른 스트림을 사용하려면 다음 파라미터를 변경합니다.

```
private static final String STREAM_NAME="my-stream";
```

Note

PutMediaAPI 예제는 스트림을 생성하지 않습니다. 스트림을 생성하려면 테스트 애플리케이션 [Java 생산자 라이브러리 사용하기](#), Kinesis Video Streams 콘솔 또는 AWS CLI를 사용해야 합니다.

- 현재 타임스탬프.

- 타임 코드 유형. 예제에서는 RELATIVE를 사용하는데, 타임스탬프가 컨테이너 시작 기준이라는 것을 가리킵니다.
- 수신된 패킷이 인가된 발신자에 의해 전송되었음을 확인하는 `AWSKinesisVideoV4Signer` 객체.
- 최대 업스트림 대역폭(단위: Kbps)
- 패킷을 받았다는 인정을 수신하는 `AckConsumer` 객체.

다음 코드는 `PutMediaClient` 객체를 생성합니다.

```

/* actually URI to send PutMedia request */
final URI uri = URI.create(KINESIS_VIDEO_DATA_ENDPOINT + PUT_MEDIA_API);

/* input stream for sample MKV file */
final InputStream inputStream = new FileInputStream(MKV_FILE_PATH);

/* use a latch for main thread to wait for response to complete */
final CountDownLatch latch = new CountDownLatch(1);

/* a consumer for PutMedia ACK events */
final AckConsumer ackConsumer = new AckConsumer(latch);

/* client configuration used for AWS SigV4 signer */
final ClientConfiguration configuration = getClientConfiguration(uri);

/* PutMedia client */
final PutMediaClient client = PutMediaClient.builder()
    .putMediaDestinationUri(uri)
    .mkvStream(inputStream)
    .streamName(STREAM_NAME)
    .timestamp(System.currentTimeMillis())
    .fragmentTimeCodeType("RELATIVE")
    .signWith(getKinesisVideoSigner(configuration))
    .upstreamKbps(MAX_BANDWIDTH_KBPS)
    .receiveAcks(ackConsumer)
    .build();

```

미디어를 스트리밍하고 스레드를 일시 중지합니다.

클라이언트가 생성된 후에는 샘플이 `putMediaInBackground`로 비동기식 스트리밍을 시작합니다. 그러면 메인 스레드가 `AckConsumer`가 반환될 때 까지, 즉 클라이언트가 닫히는 시점까지 `latch.await`에 의해 일시 중지됩니다.

```

/* start streaming video in a background thread */
    client.putMediaInBackground();

    /* wait for request/response to complete */
    latch.await();

    /* close the client */
    client.close();

```

3단계: 코드 실행 및 확인

PutMedia API 예제를 실행하려면 다음을 수행합니다.

1. Kinesis Video Streams my-stream 콘솔에서 또는 `awscli`를 사용하여 AWS CLI이름이 지정된 스트림을 생성합니다.
2. 작업 디렉터리를 Java 생산자 SDK 디렉터리로 변경합니다.

```

cd /<YOUR_FOLDER_PATH_WHERE_SDK_IS_DOWNLOADED>/amazon-kinesis-video-streams-
producer-sdk-java/

```

3. Java SDK와 데모 애플리케이션을 컴파일합니다.

```

mvn package

```

4. `/tmp` 디렉터리에 임시 파일 이름을 생성합니다.

```

jar_files=$(mktemp)

```

5. 로컬 리포지토리에서 파일로 종속성의 classpath 문자열을 생성합니다.

```

mvn -Dmdep.outputFile=$jar_files dependency:build-classpath

```

6. 다음과 같이 `LD_LIBRARY_PATH` 환경 변수의 값을 설정합니다.

```

export LD_LIBRARY_PATH=/<YOUR_FOLDER_PATH_WHERE_SDK_IS_DOWNLOADED>/amazon-kinesis-
video-streams-producer-sdk-cpp/kinesis-video-native-build/downloads/local/lib:
$LD_LIBRARY_PATH
$ classpath_values=$(cat $jar_files)

```

7. 다음과 같이 명령줄에서 AWS 자격 증명을 입력하여 데모를 실행합니다.


```
java -classpath target/kinesisvideo-java-demo-1.0-SNAPSHOT.jar:$classpath_values -
Daws.accessKeyId=${ACCESS_KEY} -Daws.secretKey=${SECRET_KEY} -Djava.library.path=/
opt/amazon-kinesis-video-streams-producer-sdk-cpp/kinesis-video-native-build
com.amazonaws.kinesisvideo.demoapp.DemoAppMain
```

8. [Kinesis Video Streams](#) 콘솔을 열고 스트림 관리 페이지에서 스트림을 선택합니다. 비디오가 [Video Preview] 창에서 재생됩니다.

예: RTSP 소스에서의 스트리밍

예는 실시간 스트리밍 프로토콜 (RTSP) 네트워크 카메라에 연결되는 [Docker](#) 컨테이너에 대한 정의가 [C++ 생산자 라이브러리](#) 포함되어 있습니다. Docker를 사용하면 Kinesis Video Streams의 운영 환경이 표준화되어 애플리케이션 구축 및 사용이 간소화됩니다.

다음 절차에서는 RTSP 데모 애플리케이션을 설정하고 사용하는 방법을 보여 줍니다.

주제

- [자습서 비디오](#)
- [사전 조건](#)
- [도커 이미지를 빌드하십시오.](#)
- [RTSP 예제 애플리케이션을 실행합니다.](#)

자습서 비디오

이 동영상은 RTSP 피드를 AWS 클라우드와 Amazon Kinesis Video Streams로 전송하도록 Raspberry Pi를 설정하는 방법을 보여줍니다. 이 데모는 데모입니다. end-to-end

이 동영상은 컴퓨터 비전과 Amazon Rekognition을 사용하여 피드에서 이미지를 캡처하여 이미지를 처리하고 알림을 보내는 방법을 보여줍니다.

사전 조건

Kinesis Video Streams RTSP 예제 애플리케이션을 실행하려면 다음이 있어야 합니다.

- 도커: 도커 설치 및 사용에 대한 자세한 내용은 다음 링크를 참조하십시오.
 - [도커 다운로드 지침](#)
 - [도커 시작하기](#)

- RTSP 네트워크 카메라 소스: 권장 카메라에 대한 자세한 내용은 [시스템 사양](#) 단원을 참조하십시오.

도커 이미지를 빌드하십시오.

먼저 데모 애플리케이션이 내부에서 실행될 Docker 이미지를 빌드합니다.

1. Amazon Kinesis Video Streams 데모 리포지토리를 복제하십시오.

```
git clone https://github.com/aws-samples/amazon-kinesis-video-streams-demos.git
```

2. Dockerfile이 들어 있는 디렉터리로 변경합니다. 이 경우에는 [docker-rtsp](#) 디렉터리입니다.

```
cd amazon-kinesis-video-streams-demos/producer-cpp/docker-rtsp/
```

3. 다음 명령을 사용하여 Docker 이미지를 빌드합니다. 이 명령어는 이미지를 만들고 rtspdockertest 로 태그를 지정합니다.

```
docker build -t rtspdockertest .
```

4. 태그가 docker images 지정된 이미지 ID를 실행하고 검색합니다. rtspdockertest

예를 들어, 아래 샘플 출력에서는 IMAGE ID is가 54f0d65f69b2 표시됩니다.

REPOSITORY	TAG	IMAGE ID	CREATED	PLATFORM	SIZE
rtspdockertest	latest	54f0d65f69b2	10 minutes ago	linux/arm64	653.1 MiB
	BLOB SIZE				292.4 MiB

이는 이후 단계에서 필요합니다.

RTSP 예제 애플리케이션을 실행합니다.

Docker 컨테이너 내부 또는 외부에서 RTSP 예제 애플리케이션을 실행할 수 있습니다. 아래의 해당 지침을 따르세요.

주제

- [Docker 컨테이너 내에서](#)
- [Docker 컨테이너 외부](#)

Docker 컨테이너 내에서

RTSP 예제 애플리케이션을 실행합니다.

1. 다음 명령을 사용하여 Amazon Kinesis Video Streams Docker 컨테이너를 시작합니다.

```
docker run -it YourImageId /bin/bash
```

2. 샘플 애플리케이션을 시작하려면 AWS 자격 증명, Amazon Kinesis 비디오 스트림의 이름, RTSP 네트워크 카메라의 URL을 제공하십시오.

⚠ Important

임시 자격 증명을 사용하는 경우 자격 증명도 제공해야 합니다. `AWS_SESSION_TOKEN` 아래 두 번째 예를 참조하십시오.

```
export AWS_ACCESS_KEY_ID=YourAccessKeyId
export AWS_SECRET_ACCESS_KEY=YourSecretKeyId
export AWS_DEFAULT_REGION=YourAWSRegion
./kvs_gstreamer_sample YourStreamName YourRtspUrl
```

임시 자격 증명:

```
export AWS_ACCESS_KEY_ID=YourAccessKeyId
export AWS_SECRET_ACCESS_KEY=YourSecretKeyId
export AWS_SESSION_TOKEN=YourSessionToken
export AWS_DEFAULT_REGION=YourAWSRegion
./kvs_gstreamer_sample YourStreamName YourRtspUrl
```

3. AWS Management Console 로그인하고 [Kinesis Video Streams](#) 콘솔을 엽니다.

스트림을 보십시오.

4. Docker 컨테이너를 종료하려면 터미널 창을 닫거나 다음을 입력합니다 `exit`.

Docker 컨테이너 외부

Docker 컨테이너 외부에서 다음 명령을 사용합니다.

```
docker run -it YourImageId /bin/bash -c "export AWS_ACCESS_KEY_ID=YourAccessKeyId;  
export AWS_SECRET_ACCESS_KEY=YourSecretKeyId; export  
AWS_SESSION_TOKEN=YourSessionToken; export AWS_DEFAULT_REGION=Your AWS Region; ./  
kvs_gstreamer_sample YourStreamName YourRtspUrl"
```

예: Kinesis Video Streams 프래그먼트의 파싱 및 렌더링

예는 Amazon Kinesis 비디오 스트림 KinesisVideoRendererExample 프래그먼트의 구문 분석 및 렌더링을 보여주는 이름의 데모 애플리케이션이 [스트림 구문 분석기 라이브러리](#) 포함되어 있습니다. 이 예제에서는 [JCodec](#)을 사용하여 [예: Kinesis Video Streams 프로듀서 SDK GStreamer 플러그인 - kvssink](#) 애플리케이션을 사용해 수집된 H.264 인코딩 프레임을 디코딩합니다. JCodec을 사용하여 프레임을 디코딩한 후에는 [JFrame](#)을 사용하여 표시된 이미지가 렌더링됩니다.

이 예에서는 다음 작업을 수행하는 방법을 보여줍니다.

- GetMediaAPI를 사용하여 Kinesis 비디오 스트림에서 프레임을 검색하고 볼 수 있도록 스트림을 렌더링합니다.
- Kinesis Video Streams 콘솔을 사용하는 대신 사용자 지정 애플리케이션에서 스트림의 비디오 콘텐츠를 볼 수 있습니다.

또한 이 예제의 클래스를 사용하여 표시되기 전에 디코딩할 필요가 없는 JPEG 파일 스트림과 같이 H.264로 인코딩되지 않은 Kinesis 비디오 스트림 콘텐츠를 볼 수 있습니다.

다음 절차에서는 렌더러 데모 애플리케이션을 설정하고 사용하는 방법을 소개합니다.

사전 조건

렌더러 예제 라이브러리를 검사하고 사용하려면 다음이 필요합니다.

- Amazon Web Services (AWS) 계정. 아직 AWS 계정이 없는 경우 [Kinesis Video Streams 시작하기](#)를 참조하십시오.
- [이클립스 자바 네온 또는 JetBrains 인텔리제이 아이디어와 같은 자바 통합 개발 환경 \(IDE\)](#)

렌더러 예제 실행

1. 디렉토리를 만든 다음 리포지토리에서 예제 소스 코드를 복제합니다 GitHub.

```
git clone https://github.com/aws/amazon-kinesis-video-streams-parser-library
```

2. 사용 중인 Java IDE(예: [Eclipse](#) 또는 [IntelliJ IDEA](#))를 열고 다운로드한 Apache Maven 프로젝트를 가져옵니다.
 - Eclipse: [File], [Import], [Maven], [Existing Maven Projects]를 차례로 선택합니다. kinesis-video-streams-parser-lib 디렉터리로 이동합니다.
 - IntelliJ Idea에서 [Import]를 선택합니다. 다운로드한 패키지의 루트에 있는 pom.xml 파일을 찾습니다.

Note

IntelliJ가 종속성을 찾을 수 없는 경우 다음을 수행해야 할 수 있습니다.

- 빌드 정리: File(파일), Settings(설정), Build, Execution, Deployment(구축, 실행, 배포), Compiler(컴파일러)를 선택합니다. 재빌드 시 출력 디렉토리 지우기가 선택되어 있는지 확인한 다음 Build, Build Project를 선택합니다.
- 프로젝트 다시 가져오기: 프로젝트의 컨텍스트(오른쪽 클릭) 메뉴를 열고 Maven, Reimport(다시 가져오기)를 선택합니다.

자세한 내용은 관련 IDE 문서를 참조하십시오.

3. Java IDE에서 src/test/java/com.amazonaws.kinesisvideo.parser/examples/KinesisVideoRendererExampleTest를 엽니다.
4. 파일에서 @Ignore 명령을 제거합니다.
5. Kinesis 비디오 스트림의 이름으로 .stream 파라미터를 업데이트합니다.
6. KinesisVideoRendererExample 테스트를 실행합니다.

작동 방식

이 예제 애플리케이션은 다음과 같은 방법을 보여줍니다.

- [MKV 데이터 전송](#)
- [MKV 프래그먼트를 프레임으로 파싱하기](#)
- [프레임 디코딩 및 표시](#)

MKV 데이터 전송

이 예제에서는 `rendering_example_video.mkv` 파일에서 샘플 MKV 데이터를 전송하고, `PutMedia` 사용하여 이라는 스트림으로 비디오 데이터를 전송합니다. `render-example-stream`

이 애플리케이션에서는 `PutMediaWorker`를 생성합니다.

```
PutMediaWorker putMediaWorker = PutMediaWorker.create(getRegion(),
    getCredentialsProvider(),
    getStreamName(),
    inputStream,
    streamOps.amazonKinesisVideo);
executorService.submit(putMediaWorker);
```

`PutMediaWorker` 클래스에 대한 자세한 내용은 [스트림 구문 분석기 라이브러리](#) 설명서의 [호출 PutMedia](#) 단원을 참조하십시오.

MKV 프래그먼트를 프레임으로 파싱하기

그런 다음 이 예에서는 `GetMediaWorker`를 사용하여 스트림에서 MKV 조각을 검색해 구문 분석합니다.

```
GetMediaWorker getMediaWorker = GetMediaWorker.create(getRegion(),
    getCredentialsProvider(),
    getStreamName(),
    new StartSelector().withStartSelectorType(StartSelectorType.EARLIEST),
    streamOps.amazonKinesisVideo,
    getMediaProcessingArgumentsLocal().getFrameVisitor());
executorService.submit(getMediaWorker);
```

`GetMediaWorker` 클래스에 대한 자세한 내용은 [스트림 구문 분석기 라이브러리](#) 설명서의 [통화 GetMedia](#) 단원을 참조하십시오.

프레임 디코딩 및 표시

이 예에서는 [JFrame](#)을 사용하여 프레임을 디코딩하여 표시합니다.

다음 코드 예제는 `KinesisVideoFrameViewer` 클래스에서 가져온 것으로, `JFrame`을 확장합니다.

```
public void setImage(BufferedImage bufferedImage) {
    image = bufferedImage;
```

```
repaint();  
}
```

이미지는 [java.awt.image](#) 인스턴스로 표시됩니다. [BufferedImage](#). [BufferedImage](#) 작업 방법을 보여주는 예제는 [이미지 읽기/로드](#)를 참조하십시오.

아마존 키네시스 비디오 스트림 모니터링

모니터링은 Amazon Kinesis Video Streams와 솔루션의 안정성, 가용성 및 성능을 유지하는 데 있어 중요한 부분입니다. AWS 다중 지점 장애가 발생할 경우 이를 디버깅하는 데 도움이 되도록 AWS 솔루션의 모든 부분에서 모니터링 데이터를 수집하는 것이 좋습니다. Amazon Kinesis Video Streams 모니터링을 시작하기 전에 다음 질문에 대한 답변이 포함된 모니터링 계획을 작성하는 것이 좋습니다.

- 모니터링의 목표
- 모니터링할 리소스
- 이러한 리소스를 모니터링하는 빈도
- 사용할 모니터링 도구
- 모니터링 작업을 수행할 사람
- 문제 발생 시 알려야 할 대상

모니터링 목표를 정의하고 모니터링 계획을 수립한 후, 다음 단계는 사용자 환경에서 정상적인 Amazon Kinesis Video Streams 성능에 대한 기준을 설정하는 것입니다. 다양한 시간과 다양한 로드 조건에서 Amazon Kinesis Video Streams 성능을 측정해야 합니다. Amazon Kinesis Video Streams를 모니터링할 때는 수집한 모니터링 데이터 기록을 저장하십시오. 현재 Amazon Kinesis Video Streams 성능을 이 기록 데이터와 비교하여 정상적인 성능 패턴 및 성능 이상을 식별하고 발생할 수 있는 문제를 해결하는 방법을 고안할 수 있습니다.

주제

- [를 사용하여 Amazon Kinesis Video Streams 지표를 모니터링합니다. CloudWatch](#)
- [를 사용하여 Amazon Kinesis Video Streams 엣지 에이전트를 모니터링하기 CloudWatch](#)
- [를 사용하여 Amazon Kinesis Video Streams API 호출을 로깅합니다. AWS CloudTrail](#)

를 사용하여 Amazon Kinesis Video Streams 지표를 모니터링합니다. CloudWatch

Amazon CloudWatch Kinesis Video Streams의 원시 데이터를 수집하여 읽기 쉬운 거의 실시간 지표로 처리하는 Amazon을 사용하여 Kinesis 비디오 스트림을 모니터링할 수 있습니다. 이러한 통계는 15개월 동안 기록되므로 기록 정보에 액세스하고 웹 애플리케이션 또는 서비스 성능을 더 잘 파악할 수 있습니다.

[Amazon Kinesis Video Streams](#) 콘솔에서는 다음 두 가지 방법으로 Amazon Kinesis 비디오 스트림에 대한 지표를 CloudWatch 볼 수 있습니다.

- 대시보드 페이지에서 현재 지역의 계정 수준 지표 섹션에서 비디오 스트림 탭을 선택합니다.
- 비디오 스트림의 세부 정보 페이지에서 모니터링 탭을 선택합니다.


Amazon Kinesis Video Streams는 다음과 같은 지표를 제공합니다.

지표	설명
ArchivedFragmentsConsumed.Media	모든 API에서 사용한 프래그먼트 미디어 할당량 포인트 수입니다. 쿼터 포인트의 개념에 대한 설명은 을 참조하십시오. the section called “프래그먼트 메타데이터 및 프래그먼트 미디어 할당량” 단위: 개
ArchivedFragmentsConsumed.Metadata	모든 API에서 사용한 프래그먼트 메타데이터 쿼터 포인트의 수입니다. 쿼터 포인트의 개념에 대한 설명은 을 참조하십시오. the section called “프래그먼트 메타데이터 및 프래그먼트 미디어 할당량” 단위: 개
PutMedia.Requests	지정된 스트림에 대한 PutMedia API 요청 수 단위: 개
PutMedia.IncomingBytes	스트림의 PutMedia 일부로 수신된 바이트 수입니다. 단위: 바이트
PutMedia.IncomingFragments	스트림의 PutMedia 일부로 수신된 전체 프래그먼트 수입니다. 단위: 개
PutMedia.IncomingFrames	스트림의 PutMedia 일부로 수신된 전체 프레임 수입니다.


지표	설명
	단위: 개
PutMedia.ActiveConnections	서비스 호스트에 대한 총 연결 수입니다. 단위: 개
PutMedia.ConnectionErrors	스트림에 대한 PutMedia 연결을 설정하는 중 오류가 발생했습니다. 단위: 개
PutMedia.FragmentIngestionLatency	Amazon Kinesis Video Streams에서 프래그먼트의 첫 바이트와 마지막 바이트를 수신하는 시점 사이의 시간 차이입니다. 단위: 밀리초
PutMedia.FragmentPersistLatency	전체 프래그먼트 데이터가 수신되고 보관된 시점부터 소요된 시간입니다. 단위: 개
PutMedia.Latency	연결을 설정하는 InletService 동안 요청과 HTTP 응답 사이의 시간 차이. 단위: 개
PutMedia.BufferingAckLatency	Amazon Kinesis Video Streams가 새 프래그먼트의 첫 바이트를 수신한 시점과 프래그먼트에 대해 버퍼링 ACK를 전송하는 시점 사이의 시간 차이입니다. 단위: 밀리초
PutMedia.ReceivedAckLatency	Amazon Kinesis Video Streams가 새 프래그먼트의 마지막 바이트를 수신한 시점과 해당 프래그먼트에 대해 수신된 ACK를 보낸 시점 사이의 시간 차이입니다. 단위: 밀리초

지표	설명
PutMedia.PersistedAckLatency	Amazon Kinesis Video Streams가 새 프래그먼트의 마지막 바이트를 수신한 시점과 해당 프래그먼트에 대해 지속적 ACK를 전송하는 시점 사이의 시간 차이입니다. 단위: 밀리초
PutMedia.ErrorAckCount	스트림에 대해 작업을 수행하는 동안 전송된 오류 ACK 수입니다. PutMedia 단위: 개
PutMedia.Success	조각이 성공적으로 기록될 때마다 1, 조각이 실패할 때마다 0이 계산됩니다. 이 지표의 평균 값으로 완전하고 유효한 조각의 전송 수를 나타냅니다. 단위: 개
GetMedia.Requests	특정 스트림에 대한 GetMedia API 요청 수. 단위: 개
GetMedia.OutgoingBytes	지정된 스트림에 대해 GetMedia API의 일부로 서비스에서 전송한 총 바이트 수입니다. 단위: 바이트
GetMedia.OutgoingFragments	스트림에 대해 작업을 수행하는 GetMedia 동안 전송된 프래그먼트 수입니다. 단위: 개
GetMedia.OutgoingFrames	지정된 GetMedia 스트림에서 전송된 프레임 수입니다. 단위: 개

지표	설명
<code>GetMedia.MillisBehindNow</code>	현재 서버 타임스탬프와 마지막으로 전송된 프래그먼트의 서버 타임스탬프 간의 시간 차이 단위: 밀리초
<code>GetMedia.ConnectionErrors</code>	설정되지 않은 연결 수입입니다. 단위: 개
<code>GetMedia.Success</code>	조각이 성공적으로 전송될 때마다 1, 실패할 때마다 0 이 계산됩니다. 평균 값으로 성공률이 표시됩니다.
	<div data-bbox="777 753 899 791" data-label="Section-Header">Note</div> <div data-bbox="820 808 1482 1039" data-label="Text"> <p>400 (사용자) 에러와 500 (시스템) 에러가 모두 발생했습니다. 요청 ID를 포함하여 요청 및 응답 요약을 활성화하는 방법에 대한 자세한 내용은 AWS 요청/응답 요약 로깅을 참조하십시오.</p> </div>
	단위: 개
<code>GetMediaForFragmentList.OutgoingBytes</code>	지정된 스트림에 대해 <code>GetMediaForFragmentList</code> API의 일부로 서비스에서 전송한 총 바이트 수입입니다. 단위: 바이트
<code>GetMediaForFragmentList.OutgoingFragments</code>	지정된 스트림에 대해 <code>GetMediaForFragmentList</code> API의 일부로 서비스에서 전송된 총 프래그먼트 수입입니다. 단위: 개

지표	설명
GetMediaForFragmentList.OutgoingFrames	지정된 스트림에 대해 GetMediaForFragmentList API의 일부로 서비스에서 전송된 총 프레임 수입입니다. 단위: 개
GetMediaForFragmentList.Requests	특정 스트림에 대한 GetMediaForFragmentList API 요청 수. 단위: 개
GetMediaForFragmentList.Success	조각이 성공적으로 전송될 때마다 1, 실패할 때마다 0이 계산됩니다. 평균 값으로 성공률이 표시됩니다. <div data-bbox="748 814 1507 1178" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"> <p> Note</p> <p>400 (사용자) 에러와 500 (시스템) 에러가 모두 발생했습니다. 요청 ID를 포함한 요청 및 응답 요약을 활성화하는 방법에 대한 자세한 내용은 AWS 요청/응답 요약 로깅을 참조하십시오.</p> </div> 단위: 개
ListFragments.Latency	지정된 스트림 이름에 대한 ListFragments API 호출의 지연 시간. 단위: 밀리초
ListFragments.Requests	특정 스트림에 대한 ListFragments API 요청 수. 단위: 개

지표	설명
ListFragments.Success	<p>요청이 성공할 때마다 1, 실패할 때마다 0 평균 값으로 성공률이 표시됩니다.</p> <div data-bbox="748 352 1507 709" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"> <p>Note</p> <p>400 (사용자) 에러와 500 (시스템) 에러가 모두 발생했습니다. 요청 ID를 포함한 요청 및 응답 요약을 활성화하는 방법에 대한 자세한 내용은 AWS 요청/응답 요약 로깅을 참조하십시오.</p> </div> <p>단위: 개</p>
GetHLSStreamingSessionURL.Latency	<p>지정된 스트림 이름에 대한 GetHLSStreamingSessionURL API 호출의 지연 시간.</p> <p>단위: 밀리초</p>
GetHLSStreamingSessionURL.Requests	<p>특정 스트림에 대한 GetHLSStreamingSessionURL API 요청 수.</p> <p>단위: 개</p>

지표	설명
GetHLSStreamingSessionURL.Success	<p>요청이 성공할 때마다 1, 실패할 때마다 0 평균 값으로 성공률이 표시됩니다.</p> <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> Note</p> <p>400 (사용자) 에러와 500 (시스템) 에러가 모두 발생했습니다. 요청 ID를 포함한 요청 및 응답 요약을 활성화하는 방법에 대한 자세한 내용은 AWS 요청/응답 요약 로깅을 참조하십시오.</p> </div> <p>단위: 개</p>
GetHLSMasterPlaylist.Latency	<p>지정된 스트림 이름에 대한 GetHLSMasterPlaylist API 호출의 지연 시간.</p> <p>단위: 밀리초</p>
GetHLSMasterPlaylist.Requests	<p>특정 스트림에 대한 GetHLSMasterPlaylist API 요청 수.</p> <p>단위: 개</p>


지표	설명
GetHLSMasterPlaylist.Success	<p>요청이 성공할 때마다 1, 실패할 때마다 0 평균 값으로 성공률이 표시됩니다.</p> <div data-bbox="748 352 1507 709" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p>Note</p> <p>400 (사용자) 에러와 500 (시스템) 에러가 모두 발생했습니다. 요청 ID를 포함한 요청 및 응답 요약을 활성화하는 방법에 대한 자세한 내용은 AWS 요청/응답 요약 로깅을 참조하십시오.</p> </div> <p>단위: 개</p>
GetHLSMediaPlaylist.Latency	<p>지정된 스트림 이름에 대한 GetHLSMediaPlaylist API 호출의 지연 시간.</p> <p>단위: 밀리초</p>
GetHLSMediaPlaylist.Requests	<p>특정 스트림에 대한 GetHLSMediaPlaylist API 요청 수.</p> <p>단위: 개</p>

지표	설명
GetHLSMediaPlaylist.Success	<p>요청이 성공할 때마다 1, 실패할 때마다 0 평균 값으로 성공률이 표시됩니다.</p> <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>400 (사용자) 에러와 500 (시스템) 에러가 모두 발생했습니다. 요청 ID를 포함한 요청 및 응답 요약을 활성화하는 방법에 대한 자세한 내용은 AWS 요청/응답 요약 로깅을 참조하십시오.</p> </div> <p>단위: 개</p>
GetMP4InitFragment.Latency	<p>지정된 스트림 이름에 대한 GetMP4InitFragment API 호출의 지연 시간.</p> <p>단위: 밀리초</p>
GetMP4InitFragment.Requests	<p>특정 스트림에 대한 GetMP4InitFragment API 요청 수.</p> <p>단위: 개</p>

지표	설명
GetMP4InitFragment.Success	<p>요청이 성공할 때마다 1, 실패할 때마다 0 평균 값으로 성공률이 표시됩니다.</p> <div style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>400 (사용자) 에러와 500 (시스템) 에러가 모두 발생했습니다. 요청 ID를 포함한 요청 및 응답 요약을 활성화하는 방법에 대한 자세한 내용은 AWS 요청/응답 요약 로깅을 참조하십시오.</p> </div> <p>단위: 개</p>
GetMP4MediaFragment.Latency	<p>지정된 스트림 이름에 대한 GetMP4MediaFragment API 호출의 지연 시간.</p> <p>단위: 밀리초</p>
GetMP4MediaFragment.Requests	<p>특정 스트림에 대한 GetMP4MediaFragment API 요청 수.</p> <p>단위: 개</p>

지표	설명
GetMP4MediaFragment.Success	<p>요청이 성공할 때마다 1, 실패할 때마다 0 평균 값으로 성공률이 표시됩니다.</p> <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> Note</p> <p>400 (사용자) 에러와 500 (시스템) 에러가 모두 발생했습니다. 요청 ID를 포함한 요청 및 응답 요약을 활성화하는 방법에 대한 자세한 내용은 AWS 요청/응답 요약 로깅을 참조하십시오.</p> </div> <p>단위: 개</p>
GetMP4MediaFragment.OutgoingBytes	<p>지정된 스트림에 대해 GetMP4MediaFragment API의 일부로 서비스에서 전송한 총 바이트 수입니다.</p> <p>단위: 바이트</p>
GetTSFragment.Latency	<p>지정된 스트림 이름에 대한 GetTSFragment API 호출 지연 시간.</p> <p>단위: 밀리초</p>
GetTSFragment.Requests	<p>특정 스트림에 대한 GetTSFragment API 요청 수.</p> <p>단위: 개</p>

지표	설명
GetTSFragment.Success	<p>요청이 성공할 때마다 1, 실패할 때마다 0 평균 값으로 성공률이 표시됩니다.</p> <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p>Note</p> <p>400 (사용자) 에러와 500 (시스템) 에러가 모두 발생했습니다. 요청 ID를 포함한 요청 및 응답 요약을 활성화하는 방법에 대한 자세한 내용은 AWS 요청/응답 요약 로깅을 참조하십시오.</p> </div> <p>단위: 개</p>
GetTSFragment.OutgoingBytes	<p>지정된 스트림에 대해 GetTSFragment API의 일부로 서비스에서 전송한 총 바이트 수입니다.</p> <p>단위: 바이트</p>
GetDASHStreamingSessionURL.Latency	<p>지정된 스트림 이름에 대한 GetDASHStreamingSessionURL API 호출 지연 시간.</p> <p>단위: 밀리초</p>
GetDASHStreamingSessionURL.Requests	<p>특정 스트림에 대한 GetDASHStreamingSessionURL API 요청 수.</p> <p>단위: 개</p>

지표	설명
GetDASHStreamingSessionURL.Success	<p>요청이 성공할 때마다 1, 실패할 때마다 0 평균 값으로 성공률이 표시됩니다.</p> <div data-bbox="748 352 1507 716" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> Note</p> <p>400 (사용자) 에러와 500 (시스템) 에러가 모두 발생했습니다. 요청 ID를 포함한 요청 및 응답 요약을 활성화하는 방법에 대한 자세한 내용은 AWS 요청/응답 요약 로깅을 참조하십시오.</p> </div> <p>단위: 개</p>
GetDASHManifest.Latency	<p>지정된 스트림 이름에 대한 GetDASHManifest API 호출의 지연 시간.</p> <p>단위: 밀리초</p>
GetDASHManifest.Requests	<p>특정 스트림에 대한 GetDASHManifest API 요청 수.</p> <p>단위: 개</p>

지표	설명
GetDASHManifest.Success	<p>요청이 성공할 때마다 1, 실패할 때마다 0 평균 값으로 성공률이 표시됩니다.</p> <div data-bbox="748 352 1507 709" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p>Note</p> <p>400 (사용자) 에러와 500 (시스템) 에러가 모두 발생했습니다. 요청 ID를 포함한 요청 및 응답 요약을 활성화하는 방법에 대한 자세한 내용은 AWS 요청/응답 요약 로깅을 참조하십시오.</p> </div> <p>단위: 개</p>
GetClip.Latency	<p>지정된 비디오 스트림 이름에 대한 GetClip API 호출의 지연 시간.</p> <p>단위: 밀리초</p>
GetClip.Requests	<p>특정 비디오 스트림에 대한 GetClip API 요청 수.</p> <p>단위: 개</p>
GetClip.Success	<p>요청이 성공할 때마다 1, 실패할 때마다 0 평균 값으로 성공률이 표시됩니다.</p> <div data-bbox="748 1360 1507 1717" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p>Note</p> <p>400 (사용자) 에러와 500 (시스템) 에러가 모두 발생했습니다. 요청 ID를 포함하여 요청 및 응답 요약을 활성화하는 방법에 대한 자세한 내용은 AWS 요청/응답 요약 로깅을 참조하십시오.</p> </div> <p>단위: 개</p>

지표	설명
GetClip.OutgoingBytes	지정된 비디오 스트림에 대해 GetClip API의 일부로 서비스에서 전송한 총 바이트 수입니다. 단위: 바이트

CloudWatch 지표 지침

CloudWatch 메트릭은 다음 질문에 대한 답을 찾는 데 도움이 될 수 있습니다.

주제

- [데이터가 Amazon Kinesis Video Streams 서비스에 도달하고 있습니까?](#)
- [Amazon Kinesis Video Streams 서비스에서 데이터를 제대로 수집하지 못하는 이유는 무엇입니까?](#)
- [Amazon Kinesis Video Streams 서비스에서 생산자가 전송한 것과 동일한 속도로 데이터를 읽을 수 없는 이유는 무엇입니까?](#)
- [콘솔에 비디오가 없거나 비디오가 지연되어 재생되는 이유는 무엇입니까?](#)
- [실시간 데이터 읽기가 지연되는 이유는 무엇이며 클라이언트가 스트림의 헤드보다 지연되는 이유는 무엇입니까?](#)
- [클라이언트가 Kinesis 비디오 스트림에서 데이터를 읽고 있습니까? 속도는 어느 정도입니까?](#)
- [클라이언트가 Kinesis 비디오 스트림에서 데이터를 읽을 수 없는 이유는 무엇입니까?](#)

데이터가 Amazon Kinesis Video Streams 서비스에 도달하고 있습니까?

관련 측정치:

- PutMedia.IncomingBytes
- PutMedia.IncomingFragments
- PutMedia.IncomingFrames

작업 항목:

- 이러한 지표가 떨어지면 애플리케이션이 여전히 서비스에 데이터를 보내고 있는지 확인하십시오.
- 네트워크 대역폭을 확인합니다. 네트워크 대역폭이 충분하지 않으면 서비스에서 데이터를 수신하는 속도가 느려질 수 있습니다.

Amazon Kinesis Video Streams 서비스에서 데이터를 제대로 수집하지 못하는 이유는 무엇입니까?

관련 측정치:

- PutMedia.Requests
- PutMedia.ConnectionErrors
- PutMedia.Success
- PutMedia.ErrorAckCount

작업 항목:

- 증가하면 프로듀서 클라이언트가 수신한 HTTP 응답 및 오류 코드를 살펴보고 연결을 설정하는 동안 어떤 오류가 발생하는지 확인하십시오. PutMedia.ConnectionErrors
- PutMedia.Success 감소하거나 증가하는 경우 PutMedia.ErrorAckCount 서비스에서 보낸 ack 응답의 ack 오류 코드를 살펴보고 데이터 수집이 실패하는 이유를 확인하세요. [자세한 내용은 .Values를 참조하십시오. AckErrorCode](#)

Amazon Kinesis Video Streams 서비스에서 생산자가 전송한 것과 동일한 속도로 데이터를 읽을 수 없는 이유는 무엇입니까?

관련 측정치:

- PutMedia.FragmentIngestionLatency
- PutMedia.IncomingBytes

작업 항목:

- 이러한 지표가 떨어지면 연결의 네트워크 대역폭을 확인하십시오. 낮은 대역폭 연결로 인해 데이터가 낮은 속도로 서비스에 도달할 수 있습니다.

콘솔에 비디오가 없거나 비디오가 지연되어 재생되는 이유는 무엇입니까?

관련 측정치:

- PutMedia.FragmentIngestionLatency

- `PutMedia.FragmentPersistLatency`
- `PutMedia.Success`
- `ListFragments.Latency`
- `PutMedia.IncomingFragments`

작업 항목:

- `PutMedia.FragmentIngestionLatency` 증가하거나 감소하는 경우 네트워크 대역폭과 데이터가 계속 전송되고 있는지 확인하세요. `PutMedia.IncomingFragments`
- 드롭인이 발생한 경우 ack 오류 코드를 확인하세요. `PutMedia.Success` 자세한 내용은 [AckErrorCode.Values](#)를 참조하십시오.
- OR가 `PutMedia.FragmentPersistLatency` `ListFragments.Latency` 증가하면 서비스 문제가 발생할 가능성이 큼니다. 상태가 장기간 지속되면 고객 서비스 담당자에게 문의하여 서비스에 문제가 있는지 확인하세요.

실시간 데이터 읽기가 지연되는 이유는 무엇이며 클라이언트가 스트림의 헤드보다 지연되는 이유는 무엇입니까?

관련 측정치:

- `GetMedia.MillisBehindNow`
- `GetMedia.ConnectionErrors`
- `GetMedia.Success`

작업 항목:

- 스트림의 양이 증가하면 스트림에 `GetMedia.ConnectionErrors` 다시 연결하려는 잦은 시도 때문에 소비자가 스트림을 제대로 읽지 못하는 것일 수 있습니다. `GetMedia` 요청에 반환된 HTTP 응답/오류 코드를 확인합니다.
- 시청률이 떨어진다면 서비스가 소비자에게 데이터를 전송하지 못해 연결이 끊기고 소비자와 다시 연결되어 소비자가 스트림의 선두주자보다 뒤처지게 되기 때문일 수 있습니다. `GetMedia.Success`
- 대역폭이 증가하면 대역폭 한도를 살펴보고 낮은 대역폭으로 인해 더 느린 속도로 데이터를 수신하고 있는지 확인하세요. `GetMedia.MillisBehindNow`

클라이언트가 Kinesis 비디오 스트림에서 데이터를 읽고 있습니까? 속도는 어느 정도입니까?

관련 측정치:

- `GetMedia.OutgoingBytes`
- `GetMedia.OutgoingFragments`
- `GetMedia.OutgoingFrames`
- `GetMediaForFragmentList.OutgoingBytes`
- `GetMediaForFragmentList.OutgoingFragments`
- `GetMediaForFragmentList.OutgoingFrames`

작업 항목:

- 이러한 지표는 실시간 및 보관된 데이터를 읽는 속도를 나타냅니다.

클라이언트가 Kinesis 비디오 스트림에서 데이터를 읽을 수 없는 이유는 무엇입니까?

관련 측정치:

- `GetMedia.ConnectionErrors`
- `GetMedia.Success`
- `GetMediaForFragmentList.Success`
- `PutMedia.IncomingBytes`

작업 항목:

- 증가분이 있는 경우 `GetMedia` 요청에서 `GetMedia.ConnectionErrors` 반환된 HTTP 응답과 오류 코드를 살펴보세요. 자세한 내용은 [AckErrorCode.Values](#)를 참조하십시오.
- 최신 또는 실시간 데이터를 읽으려는 경우 서비스가 소비자에게 전송할 데이터가 스트림으로 들어 오는지 확인하세요 `PutMedia.IncomingBytes`.
- OR가 `GetMedia.Success` 감소한다면 서비스가 소비자에게 데이터를 전송할 수 없기 때문일 수 있습니다. `GetMediaForFragmentList.Success` 상태가 장기간 지속되면 고객 서비스 담당자에게 문의하여 서비스에 문제가 있는지 확인하세요.


를 사용하여 Amazon Kinesis Video Streams 엣지 에이전트를 모니터링하기 CloudWatch

Amazon을 사용하여 Amazon Kinesis Video Streams Edge Agent를 모니터링할 수 있습니다. CloudWatch Amazon은 원시 데이터를 수집하여 읽을 수 있는 거의 실시간 지표로 처리합니다. 이러한 통계는 15개월 동안 기록됩니다. 이 기록 정보를 통해 웹 애플리케이션 또는 Amazon Kinesis Video Streams Edge Agent 서비스가 어떻게 수행되고 있는지 더 잘 파악할 수 있습니다.

지표를 보려면 다음과 같이 하십시오.

1. <https://console.aws.amazon.com/cloudwatch/> 에서 AWS Management Console 로그인하고 CloudWatch 콘솔을 엽니다.
2. 왼쪽 탐색창의 지표 아래에서 모든 지표를 선택합니다.
3. 찾아보기 탭을 선택한 다음 EdgeRuntimeAgent사용자 지정 네임스페이스를 선택합니다.

Amazon Kinesis Video Streams Edge Agent는 네임스페이스 아래에 다음과 같은 지표를 게시합니다. EdgeRuntimeAgent

차원	State	설명
스트림 이름, RecordJob	실행 중	RecordJob 가 실행 중일 때 계속 게시합니다. 단위: 없음. 이 상태인 한 "RecordJob 1"은 게시됩니다.
	FatalError	RecordJob 치명적인 오류가 발생한 경우 게시합니다. 단위: 없음. 이 이벤트가 발생하면 "1"이 한 번 게시됩니다. <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; background-color: #e6f2ff;"> <p> Note 자세한 내용은 로그를 참조하십시오.</p> </div>
	완료됨	a가 완료되면 RecordJob 게시됩니다. 단위: 없음. 이 이벤트가 발생하면 "1"이 한 번 게시됩니다.
스트림 이름,	실행 중	UploadJob 가 실행 중일 때 계속 게시합니다.

차원	State	설명
UploadJob		단위: 없음. 이 상태인 한 "UploadJob 1"은 게시됩니다.
	FatalError	심각한 오류가 발생한 경우 UploadJob 게시합니다. 단위: 없음. 이 이벤트가 발생하면 "1"이 한 번 게시됩니다. <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; background-color: #e6f2ff;"> <p> Note 자세한 내용은 로그를 참조하십시오.</p> </div>
	완료됨	완료되면 UploadJob 게시됩니다. 단위: 없음. 이 이벤트가 발생하면 "1"이 한 번 게시됩니다.
스트림 이름	PercentageSpaceUsed	이는 Amazon Kinesis Video Streams Edge Agent 구성에 할당된 총 공간 중 녹화 미디어에 사용되는 비율입니다. 자세한 정보는 the section called "LocalSizeConfig" 을 참조하세요. 단위: 백분율 (스케일 0—1).
사물 이름	얼라이브	Amazon Kinesis Video Streams Edge 에이전트에서 실행 중인 구성에 관계없이 1분마다 게시합니다. 이를 통해 Amazon Kinesis Video Streams Edge 에이전트가 활성 상태이고 구성을 수락할 준비가 되었는지 파악할 수 있습니다. 단위: 없음. "1"은 1분마다 게시됩니다.
	RecordJobs.HealthyJobCount	Amazon Kinesis Video Streams Edge 에이전트에서 실행 중인 녹화 작업 및 예약된 녹화 작업의 총 수입니다. 단위: 개수.
	UploadJobs.HealthyJobCount	Amazon Kinesis Video Streams Edge Agent에서 실행 중인 업로드 작업 및 예약된 업로드 작업의 총 수입니다. 단위: 개수.

차원	State	설명
	RecordJobs.UnhealthyJobCount	현재 오류가 발생한 레코드 작업의 총 수입니다. 단위: 개수.
	UploadJobs.UnhealthyJobCount	현재 오류가 발생한 업로드 작업의 총 수입니다. 단위: 개수.
	RecordJobs.RunningJobCount	현재 실행 중인 레코드 작업의 총 수입니다. 단위: 개수.
	UploadJobs.RunningJobCount	현재 실행 중인 업로드 작업의 총 수입니다. 단위: 개수.
	RecordJobs.EdgeConfigCount	Amazon Kinesis Video Streams Edge 에이전트에서 처리 중인 총 레코드 구성 수입니다. 단위: 개수.
	UploadJobs.EdgeConfigCount	Amazon Kinesis Video Streams Edge 에이전트에서 처리 중인 총 업로드 구성 수입니다. 단위: 개수.

CloudWatch Amazon Kinesis Video Streams 엣지 에이전트에 대한 메트릭 지침

CloudWatch 지표는 다음 질문에 대한 답을 찾는 데 유용할 수 있습니다.

주제

- [Amazon Kinesis Video Streams Edge 에이전트에 녹화할 공간이 충분합니까?](#)
- [Amazon Kinesis Video Streams Edge 에이전트는 살아있나요?](#)
- [건강에 좋지 않은 작업이 있나요?](#)

- [외부 개입이 필요한 작업이 있나요?](#)

Amazon Kinesis Video Streams Edge 에이전트에 녹화할 공간이 충분합니까?

관련 지표: PercentageSpaceUsed

조치: 별도의 조치가 필요하지 않습니다.

Amazon Kinesis Video Streams Edge 에이전트는 살아있나요?

관련 지표: Alive

조치: 어느 시점에서든 이 지표 수신을 중단하면 Amazon Kinesis Video Streams Edge 에이전트에서 다음 중 하나 이상이 발생한 것입니다.

- 애플리케이션 런타임 문제: 메모리 또는 기타 리소스 제약, 버그 등
- 종료, 충돌 또는 종료 시 에이전트가 실행 중인 AWS IoT 장치
- AWS IoT 장치가 네트워크에 연결되어 있지 않습니다.

건강에 좋지 않은 직업이 있나요?

관련 측정치:

- RecordJobs.UnhealthyJobCount
- UploadJobs.UnhealthyJobCount

조치: 로그를 검사하고 FatalError 지표를 찾아보십시오.

- FatalError지표가 있는 경우 치명적 오류가 발생한 것이므로 작업을 수동으로 다시 시작해야 합니다. 를 사용하여 StartEdgeConfigurationUpdate 작업을 수동으로 다시 시작하기 전에 로그를 검사하고 문제를 해결하십시오.
- FatalError지표가 없는 경우 일시적 (치명적이지 않은) 오류가 발견되어 Amazon Kinesis Video Streams Edge Agent가 작업을 재시도하고 있는 것입니다.

Note

오류가 발생한 작업을 에이전트가 다시 시도하도록 하려면 `를` 사용하십시오. [the section called “StartEdgeConfigurationUpdate”](#)

외부 개입이 필요한 작업이 있나요?

관련 측정치:

- `PercentageSpaceUsed`— 이 값이 특정 값을 초과하는 경우 기록 작업은 일시 중지되고 공간이 남아 있을 때 (미디어 보존 기간이 만료될 때)에만 재개됩니다. 더 높은 값이 포함된 업데이트된 구성을 `MaxLocalMediaSizeInMB` 전송하여 작업을 즉시 업데이트할 수 있습니다.
- `RecordJob.FatalError/UploadJob.FatalError`— 에이전트의 로그를 조사하고 작업을 재개할 수 있도록 구성을 다시 보내십시오.

조치: 구성과 함께 API를 호출하여 이 문제가 발생한 작업을 다시 시작하십시오.

를 사용하여 Amazon Kinesis Video Streams API 호출을 로깅합니다. AWS CloudTrail

Amazon Kinesis Video Streams는 Amazon Kinesis Video Streams에서 사용자, 역할 또는 사용자가 취한 작업에 AWS 서비스 대한 기록을 제공하는 서비스인 서비스와 AWS CloudTrail 함께 작동합니다. CloudTrail Amazon Kinesis Video Streams에 대한 모든 API 호출을 이벤트로 캡처합니다. 캡처된 호출에는 Amazon Kinesis Video Streams 콘솔에서의 통화 및 Amazon Kinesis Video Streams API 작업에 대한 코드 호출이 포함됩니다. 트레일을 생성하면 Amazon Kinesis Video Streams에 대한 CloudTrail 이벤트를 포함하여 Amazon S3 버킷으로 이벤트를 지속적으로 전송할 수 있습니다. 트레일을 구성하지 않아도 CloudTrail 콘솔의 이벤트 기록에서 최신 이벤트를 계속 볼 수 있습니다. 에서 수집한 CloudTrail 정보를 사용하여 Amazon Kinesis Video Streams에 이루어진 요청, 요청이 이루어진 IP 주소, 요청한 사람, 요청 시기 및 추가 세부 정보를 확인할 수 있습니다.

구성 및 활성화 방법을 CloudTrail 포함하여 자세한 내용은 [사용 AWS CloudTrail 설명서를](#) 참조하십시오.

아마존 키네시스 비디오 스트림 및 CloudTrail

CloudTrail 계정을 생성하면 AWS 계정에서 활성화됩니다. Amazon Kinesis Video Streams에서 지원되는 이벤트 활동이 발생하면 해당 활동이 이벤트 기록의 CloudTrail AWS 다른 서비스 이벤트와 함께 이벤트에 기록됩니다. AWS 계정에서 최근 이벤트를 보고, 검색하고, 다운로드할 수 있습니다. 자세한 내용은 이벤트 [기록으로 CloudTrail 이벤트 보기를](#) 참조하십시오.

Amazon Kinesis Video Streams의 이벤트를 포함하여 AWS 계정에서 진행 중인 이벤트 기록을 보려면 트레일을 생성하십시오. 트레일을 사용하면 CloudTrail Amazon S3 버킷으로 로그 파일을 전송할 수 있습니다. 콘솔에서 추적을 생성하면 기본적으로 모든 AWS 리전에 추적이 적용됩니다. 트레일은 AWS 파티션에 있는 모든 지역의 이벤트를 기록하고 지정한 Amazon S3 버킷으로 로그 파일을 전송합니다. 또한 CloudTrail 로그에서 수집된 이벤트 데이터를 추가로 분석하고 이에 따라 조치를 AWS 서비스 취하도록 기타를 구성할 수 있습니다. 자세한 내용은 다음 자료를 참조하십시오.

- [추적 생성 개요](#)
- [CloudTrail 지원되는 서비스 및 통합](#)
- [에 대한 Amazon SNS 알림 구성 CloudTrail](#)
- [여러 지역에서 CloudTrail 로그 파일 수신 및 여러 계정으로부터 CloudTrail 로그 파일 수신](#)

Amazon Kinesis Video Streams는 다음과 같은 작업을 로그 파일에 이벤트로 기록할 CloudTrail 수 있도록 지원합니다.

- [CreateStream](#)
- [DeleteStream](#)
- [DescribeStream](#)
- [GetDataEndpoint](#)
- [ListStreams](#)
- [ListTagsForStream](#)
- [TagStream](#)
- [UntagStream](#)
- [UpdateDataRetention](#)
- [UpdateStream](#)

모든 이벤트 및 로그 항목에는 요청을 생성한 사용자에게 대한 정보가 들어 있습니다. 보안 인증 정보를 이용하면 다음을 쉽게 판단할 수 있습니다.

- 요청을 루트로 했는지 아니면 사용자 보안 인증으로 했는지 여부
- 역할 또는 연합된 사용자에게 대한 임시 보안 인증을 사용하여 요청이 생성되었는지 여부
- 다른 AWS 서비스에서 요청했는지 여부.

자세한 내용은 [CloudTrail사용자 ID 요소를 참조하십시오.](#)

예: Amazon Kinesis Video Streams 로그 파일 항목

트레일은 지정한 Amazon S3 버킷에 이벤트를 로그 파일로 전송할 수 있는 구성입니다. CloudTrail 로그 파일에는 하나 이상의 로그 항목이 포함되어 있습니다. 이벤트는 모든 소스로부터의 단일 요청을 나타내며 요청 작업, 작업 날짜와 시간, 요청 파라미터 등에 대한 정보가 들어 있습니다. CloudTrail 로그 파일은 공개 API 호출의 정렬된 스택 추적이 아니므로 특정 순서로 표시되지 않습니다.

다음 예제는 [CreateStream](#) 작업을 보여주는 CloudTrail 로그 항목을 보여줍니다.

```
{
  "Records": [
    {
      "eventVersion": "1.05",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:user/Alice",
        "accountId": "123456789012",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
      },
      "eventTime": "2018-05-25T00:16:31Z",
      "eventSource": "kinesisvideo.amazonaws.com",
      "eventName": "CreateStream",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "127.0.0.1",
      "userAgent": "aws-sdk-java/unknown-version Linux/x.xx",
      "requestParameters": {
        "streamName": "VideoStream",
        "dataRetentionInHours": 2,
        "mediaType": "mediaType",
        "kmsKeyId": "arn:aws:kms::us-east-1:123456789012:alias",
      },
      "deviceName": "my-device"
    },
    {
      "responseElements": {
```

```

"streamARN":arn:aws:kinesisvideo:us-east-1:123456789012:stream/VideoStream/12345"
  },
  "requestID": "db6c59f8-c757-11e3-bc3b-57923b443c1c",
  "eventID": "b7acfd0-6ca9-4ee1-a3d7-c4e8d420d99b"
},
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::123456789012:user/Alice",
    "accountId": "123456789012",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2018-05-25:17:06Z",
  "eventSource": "kinesisvideo.amazonaws.com",
  "eventName": "DeleteStream",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "aws-sdk-java/unknown-version Linux/x.xx",
  "requestParameters": {
    "streamARN": "arn:aws:kinesisvideo:us-east-1:012345678910:stream/
VideoStream/12345",
    "currentVersion": "keqrjeqkj9"
  },
  "responseElements": null,
  "requestID": "f0944d86-c757-11e3-b4ae-25654b1d3136",
  "eventID": "0b2f1396-88af-4561-b16f-398f8eaea596"
},
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::123456789012:user/Alice",
    "accountId": "123456789012",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "Alice"
  },
  "eventTime": "2014-04-19T00:15:02Z",
  "eventSource": "kinesisvideo.amazonaws.com",
  "eventName": "DescribeStream",
  "awsRegion": "us-east-1",

```

```

    "sourceIPAddress": "127.0.0.1",
    "userAgent": "aws-sdk-java/unknown-version Linux/x.xx",
    "requestParameters": {
      "streamName": "VideoStream"
    },
    "responseElements": null,
    "requestID": "a68541ca-c757-11e3-901b-cbcfe5b3677a",
    "eventID": "22a5fb8f-4e61-4bee-a8ad-3b72046b4c4d"
  },
  {
    "eventVersion": "1.05",
    "userIdentity": {
      "type": "IAMUser",
      "principalId": "EX_PRINCIPAL_ID",
      "arn": "arn:aws:iam::123456789012:user/Alice",
      "accountId": "123456789012",
      "accessKeyId": "EXAMPLE_KEY_ID",
      "userName": "Alice"
    },
    "eventTime": "2014-04-19T00:15:03Z",
    "eventSource": "kinesisvideo.amazonaws.com",
    "eventName": "GetDataEndpoint",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "aws-sdk-java/unknown-version Linux/x.xx",
    "requestParameters": {
      "streamName": "VideoStream",
      "apiName": "LIST_FRAGMENTS"
    }
  },
  {
    "responseElements": null,
    "requestID": "a6e6e9cd-c757-11e3-901b-cbcfe5b3677a",
    "eventID": "dcd2126f-c8d2-4186-b32a-192dd48d7e33"
  },
  {
    "eventVersion": "1.05",
    "userIdentity": {
      "type": "IAMUser",
      "principalId": "EX_PRINCIPAL_ID",
      "arn": "arn:aws:iam::123456789012:user/Alice",
      "accountId": "123456789012",
      "accessKeyId": "EXAMPLE_KEY_ID",
      "userName": "Alice"
    }
  },

```

```
    "eventTime": "2018-05-25T00:16:56Z",
    "eventSource": "kinesisvideo.amazonaws.com",
    "eventName": "ListStreams",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "aws-sdk-java/unknown-version Linux/x.xx",
    "requestParameters": {
      "maxResults": 100,
      "streamNameCondition": {"comparisonValue": "MyVideoStream"
comparisonOperator": "BEGINS_WITH"}}
    },
    "responseElements": null,
    "requestID": "e9f9c8eb-c757-11e3-bf1d-6948db3cd570",
    "eventID": "77cf0d06-ce90-42da-9576-71986fec411f"
  }
]
}
```

Kinesis Video Streams 서비스 할당량

Kinesis Video Streams의 서비스 할당량은 다음과 같습니다.

Important

다음 서비스 할당량은 지원 티켓을 제출하여 업그레이드할 수 있는 소프트 [s] 와 늘릴 수 없는 하드 [h] 입니다. 아래 표의 개별 서비스 할당량 옆에 [s] 와 [h] 가 표시됩니다.

컨트롤 플레인 API 서비스 할당량


다음 섹션에서는 컨트롤 플레인 API의 서비스 할당량을 설명합니다. TPS는 초당 트랜잭션을 나타냅니다.

계정 수준 또는 리소스 수준 요청 한도에 도달하면 a가 발생합니다.

`ClientLimitExceededException`

컨트롤 플레인 API 서비스 할당량

API	계정 한도: 요청	계정 한도: 스트림	스트림 레벨 한도	관련 예외 및 참고 사항
the section called “CreateStream”	50TPS[s]	미국 동부 (버지니아 북부) 및 미국 서부 (오레곤) 지역에서는 계정당 10000개의 스트림이, 지원되는 다른 모든 지역에서는 계정당 5000개의 스트림이 있습니다.	N/A	디바이스, CLI, SDK 구동 액세스 및 콘솔은 모두 이 API를 호출할 수 있습니다. 스트림이 이미 존재하지 않는 경우 하나의 API 호출만 성공합니다.

API	계정 한도: 요청	계정 한도: 스트림	스트림 레벨 한도	관련 예외 및 참고 사항
		<p> Note 계정 당 100,000 개 이상의 스트림으로 최대 한도를 증가할 수 있습니다. 로그인하여 Kinesis Video Streams에 대한 서비스 한도 증가 사례를 제출하여 이 한도를 증가를 요</p>		

API	계정 한도: 요청	계정 한도: 스트림	스트림 레벨 한도	관련 예외 및 참고 사항
		<p>청하 실행 오. AWS Managem t Console https:// console.a ws.amazo .com/</p>		
the section called “DeleteEdgeConfiguration”	10TPS[h]	N/A	1 TPS [h]	
the section called “DeleteStream”	50TPS[h]	N/A	5TPS[h]	
the section called “DescribeEdgeConfiguration”	50TPS[h]	N/A	5TPS[h]	

API	계정 한도: 요청	계정 한도: 스트림	스트림 레벨 한도	관련 예외 및 참고 사항
the section called “DescribeImageGenerationConfiguration”	50TPS[h]	N/A	5TPS[h]	
the section called “DescribeMappedResourceConfiguration”	50TPS[h]	N/A	5TPS[h]	
the section called “DescribeNotificationConfiguration”	50TPS[h]	N/A	5TPS[h]	
the section called “DescribeStream”	300TPS[h]	N/A	5TPS[h]	
the section called “GetDataEndpoint”	300TPS[h]	N/A	5TPS[h]	대부분의 PutMedia/GetMedia 사용 사례에서 스트리밍 토큰을 새로 고치기 위해 45분마다 호출됩니다. 애플리케이션이 실패 시 다시 로드하는 경우 데이터 엔드포인트를 캐시하는 것이 안전합니다.

API	계정 한도: 요청	계정 한도: 스트림	스트림 레벨 한도	관련 예외 및 참고 사항
the section called “ListEdgeAgentConfigurations”	50TPS[h]	N/A	N/A	
the section called “ListStreams”	50TPS[h]	N/A	N/A	
the section called “ListTagsForStream”	50TPS[h]	N/A	5TPS[h]	
the section called “StartEdgeConfigurationUpdate”	10TPS[h]	N/A	1 TPS [h]	
the section called “TagStream”	50TPS[h]	N/A	5TPS[h]	
the section called “UntagStream”	50TPS[h]	N/A	5TPS[h]	
the section called “UpdateDataRetention”	50TPS[h]	N/A	5TPS[h]	

API	계정 한도: 요청	계정 한도: 스트림	스트림 레벨 한도	관련 예외 및 참고 사항
the section called “UpdateImageGenerationConfiguration”	50TPS[h]	N/A	5TPS[h]	
the section called “UpdateNotificationConfiguration”	50TPS[h]	N/A	5TPS[h]	
the section called “UpdateStream”	50TPS[h]	N/A	5TPS[h]	

미디어 및 아카이브 미디어 API 서비스 할당량

다음 섹션에서는 미디어 및 아카이브 미디어 API의 서비스 할당량을 설명합니다.

계정 수준 또는 리소스 수준 요청 한도에 도달하면 a가 발생합니다.

`ClientLimitExceededException`

연결 레벨 한도에 도달하면 `ConnectionLimitExceededException`이 발생합니다.

조각 레벨 한도에 도달하면 다음 오류 또는 ack가 발생합니다.


- 최소 기간 미만의 조각에 대해서는 `MIN_FRAGMENT_DURATION_REACHED` ack가 반환됩니다.
- 최대 기간을 초과하는 조각에 대해서는 `MAX_FRAGMENT_DURATION_REACHED` ack가 반환됩니다.
- 최대 데이터 크기를 초과하는 조각에 대해서는 `MAX_FRAGMENT_SIZE` ack가 반환됩니다.

- GetMediaForFragmentList 작업에서 조각 한도에 도달하면 FragmentLimitExceeded 예외가 발생합니다.

데이터 플레인 API 서비스 할당량

API	스트림 레벨 한도	연결 레벨 한도	대역폭 한도	조각 레벨 제한	관련 예외 및 참고 사항
the section called "PutMedia"	5TPS[h]	1[s]	초당 12.5 메가바이트 또는 스트림당 100Mbps	<ul style="list-style-type: none"> • 최소 조각 기간: 1초[h] • 최대 프레임 지속 시간: 20초[h] • 최대 조각 크기: 50MB[h] • 최대 트랙 수: 3[s] • 초당 전송되는 최대 조각 수: 5[h] • 최대 프레임 메타데이터 제한: 10개 태그[h] 	일반적인 PutMedia 요청에는 여러 초 동안 데이터가 포함되므로 스트림당 TPS가 더 낮습니다. 할당량을 초과하는 동시 연결이 여러 개 있는 경우 마지막 연결이 허용됩니다.

API	스트림 레벨 한도	연결 레벨 한도	대역폭 한도	조각 레벨 제한	관련 예외 및 참고 사항
the section called “GetClip”	N/A	N/A	100MB 크기 제한[h]	최대 조각 수: 200[h]	
the section called “GetDASHStreamingSessionURL”	25 TPS [h]	N/A	해당 사항 없음	N/A	
the section called “GetHLSStreamingSessionURL”	25티스폰 [h]	N/A	해당 사항 없음	해당 사항 없음	
the section called “GetImages”	해당 사항 없음	N/A	100 메가바이트 [시간]	N/A	요청당 최대 이미지 수는 100 [h] 입니다. <div data-bbox="1133 1165 1507 1577" style="border: 1px solid #add8e6; border-radius: 15px; padding: 10px;"> <p>Note 의 최소값은 200 밀리초 (ms) 이며, 이는 초당 5 개 이미지입니다. SamplingInterval</p> </div>

API	스트림 레벨 한도	연결 레벨 한도	대역폭 한도	조각 레벨 제한	관련 예외 및 참고 사항
the section called “GetMedia”	5TPS[h]	3[s]	25MB/s 또는 200Mbps[s]	초당 전송되는 최대 프래그먼트 수: 6 [h]	<p>고유 소비 클라이언트는 TPS가 2~3개 이상 필요하지 않아야 합니다. 연결이 설정된 후에는 애플리케이션이 지속적으로 읽어야 하기 때문입니다.</p> <p>일반적인 프래그먼트가 약 5MB인 경우 이 한도는 Kinesis 비디오 스트림당 최대 75MBps를 의미합니다. 이러한 스트림의 송신 비트 속도는 최대 수신 비트 속도의 2배입니다.</p> <div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>GetMediaHLS/DASH 재생에는 사용되지 않습니다.</p> </div>
the section called “GetMediaForFragmentList”	N/A	5[s]	25MB/s 또는 200Mbps[s]	최대 조각 수: 1000[h]	<p>프래그먼트 기반 소비 애플리케이션 5개를 동시에 호출할 수 있습니다. GetMediaForFragmentList 그 이상의 연결은 거부됩니다.</p>

동영상 재생 프로토콜 API 서비스 할당량

API	세션 레벨 제한	조각 레벨 제한
GetDash ManifestPlaylist	5TPS[h]	플레이리스트당 최대 프래그먼트 수: 5000 [h]
GetHLS MasterPlaylist	5TPS[h]	N/A
GETHLS MediaPlaylist	5TPS[h]	플레이리스트당 최대 프래그먼트 수: 5000 [h]
GetMP4 InitFragment	5TPS[h]	N/A
MP4를 받으세요 MediaFragment	20 틱 [h]	N/A
GetTSFragment	20티스폰 [h]	N/A

프래그먼트 메타데이터 및 프래그먼트 미디어 할당량

[보관된 미디어에 액세스하기 위한 Kinesis Video Streams API는 API](#) 호출 수가 아닌 요청된 프래그먼트 수에 따라 조절됩니다. API는 프래그먼트 메타데이터 수와 요청된 프래그먼트 미디어 수에 따라 속도가 제한됩니다. 프래그먼트 메타데이터와 프래그먼트 미디어 할당량은 스트림별로 적용됩니다. 즉, 한 스트림의 프래그먼트 메타데이터 또는 미디어에 대한 요청은 다른 스트림의 할당량에 적용되지 않습니다. 하지만 특정 스트림 내에서는 각 할당량이 여러 API에서 공유됩니다. 즉, 특정 스트림의 경우 여러 API에서 프래그먼트를 요청하면 동일한 할당량을 소비합니다. 스트림의 프래그먼트 메타데이터 또는 프래그먼트 미디어 할당량을 초과하면 API는 `a`를 반환합니다. `ClientLimitExceededException` 다음 표는 API가 두 가지 유형의 할당량 각각을 어떻게 소비하는지 보여줍니다. 이 표의 두 번째 열에서는 스트림의 할당량이 N이면 API가 해당 스트림의 할당량 유형에서 소비할 포인트가 N개라는 것을 의미합니다. `GetClipAPI`는 두 표에 모두 표시됩니다.

프래그먼트 메타데이터 할당량 소비

API	요청당 사용된 할당량 포인트 수	공유 할당량 (N)
the section called “ListFragments”	MaxResults 파라미터 값	스트림당 초당 10000개의 쿼터 포인트 [h]

API	요청당 사용된 할당량 포인트 수	공유 할당량 (N)
the section called “GetClip”	결과 클립의 프래그먼트 수	
GetHLSMediaPlaylist	파라미터 값입니다. MaxMediaPlaylistFragmentResults	
GetDASHManifest	MaxManifestFragmentResults 파라미터 값.	
the section called “GetImages”	값: 요청된 최대 이미지 수 400개 이상	

프래그먼트 미디어 할당량 소비

API	요청당 소비된 할당량 포인트 수	공유 할당량 (N)
the section called “GetMediaForFragmentList”	프래그먼트 파라미터의 프래그먼트 수	스트림당 초당 쿼터 포인트 500개 [h]
the section called “GetClip”	결과 클립의 프래그먼트 수	
GetMP4MediaFragment	1	
GetTSFragment	1	
the section called “GetImages”	요청된 최대 이미지 수	

예를 들어 초당 500개의 프래그먼트 미디어 할당량이 있는 경우 특정 스트림에 대해 다음과 같은 호출 패턴이 지원됩니다.

- 초당 요청 5개, 각 클립에 프래그먼트 100개 GetClip 포함
- 초당 요청 100개, 각 클립에 프래그먼트 5개 GetClip 포함.
- 초당 요청 2개 - 각 클립에 프래그먼트 100개, 각 GetMediaForFragmentList 클립에 초당 요청 3개 GetClip

- ~로 초당 요청 400개 GetMP4MediaFragment, GetTSFragment ~로 초당 요청 100개

이러한 할당량은 스트림당 지원할 수 있는 HLS 및 MPEG-DASH 세션 수와 관련하여 중요한 영향을 미칩니다. 미디어 플레이어가 특정 시간에 사용할 수 있는 HLS 및 DASH 세션 수에는 제한이 없습니다. 따라서 재생 응용 프로그램에서 너무 많은 세션을 동시에 사용할 수 없도록 하는 것이 중요합니다. 다음 두 예에서는 지원할 수 있는 동시 재생 세션 수를 결정하는 방법을 설명합니다.

예 1: 라이브 스트리밍

지속 시간이 1초이고 오디오 및 비디오 트랙이 5로 MaxMediaPlaylistFragmentResults 설정된 HLS가 있는 라이브 스트리밍 시나리오에서 미디어 플레이어는 일반적으로 GetHLSMediaPlaylist 초당 2회의 통화를 전송합니다. 한 호출은 최신 비디오 메타데이터에 대한 호출이고 다른 호출은 해당 오디오 메타데이터에 대한 호출입니다. 두 번의 호출은 각각 5개의 프래그먼트 메타데이터 할당량을 소비합니다. 또한 GetMP4MediaFragment 초당 두 번의 호출이 발생합니다. 한 번은 최신 비디오를 호출하고 다른 호출은 해당 오디오를 호출합니다. 각 호출은 하나의 프래그먼트 미디어 토큰을 소비하므로 총 두 개의 토큰이 소비됩니다.

이 시나리오에서는 최대 250개의 동시 재생 세션을 지원할 수 있습니다. 250개 세션에서 이 시나리오는 초당 2,500개의 프래그먼트 메타데이터 할당량 포인트 (할당량 10,000개보다 훨씬 적음)와 초당 500개의 프래그먼트 미디어 쿼터 포인트를 소비합니다.

예 2: 온디맨드 재생

오디오 및 비디오 트랙을 1,000으로 MaxManifestFragmentResults 설정한 MPEG-DASH를 사용한 과거 이벤트의 온디맨드 재생 시나리오에서 미디어 플레이어는 일반적으로 세션 시작 시 GetDASHManifest 한 번 호출 (1,000개의 프래그먼트 메타데이터 쿼터 포인트 사용)하고 모든 프래그먼트가 로드될 때까지 초당 최대 5회 (프래그먼트 미디어 쿼터 포인트 5개 사용)의 GetMP4MediaFragment 속도로 호출합니다. 이 시나리오에서는 초당 최대 10개의 새 세션 (초당 10,000개의 프래그먼트 메타데이터 할당량)을 시작할 수 있으며, 초당 5개의 속도 (초당 500개의 프래그먼트 미디어 할당량 기준)로 최대 100개의 세션에서 프래그먼트 미디어를 활발하게 로드할 수 있습니다.

ArchivedFragmentsConsumed.Metadata 및 ArchivedFragmentsConsumed.Media를 사용하여 프래그먼트 메타데이터와 프래그먼트 미디어 쿼터 포인트의 사용량을 각각 모니터링할 수 있습니다. 모니터링에 대한 자세한 내용은 [여기](#)를 참조하십시오. [모니터링](#)

프래그먼트 메타데이터 할당량

Kinesis 비디오 스트림의 프래그먼트에 프래그먼트 메타데이터를 추가할 때는 다음과 같은 서비스 할당량이 적용됩니다.

- 1개의 조각에 최대 10개의 메타데이터 항목을 추가할 수 있습니다.
- 조각 메타데이터 이름의 길이는 최대 128바이트입니다.
- 조각 메타데이터 값의 길이는 최대 256바이트입니다.
- 프래그먼트 메타데이터 이름은 문자열 ""로 시작할 수 없습니다. AWS 그러한 메타데이터 항목이 추가된 경우 PIC의 putFragmentMetadata 메서드에서 STATUS_INVALID_METADATA_NAME 오류를 반환합니다(오류 코드 0x52000077). 이때 애플리케이션이 이 오류를 무시할 수도 있고(PIC가 메타데이터 항목을 추가하지 않음) 이 오류에 반응할 수도 있습니다.

스트림 태그

이러한 메타데이터 카-값 쌍은 Kinesis 비디오 스트림에 포함된 개별 프래그먼트가 아니라 전체 Kinesis Video Streams 리소스에 적용됩니다.

각 Kinesis 비디오 스트림은 최대 50개의 태그를 지원합니다.

스트림 태그의 키 및 값에 [the section called "TagStream"](#) 대한 제한은 을 참조하십시오.

Kinesis Video Streams 문제 해결

다음 정보를 사용하여 Amazon Kinesis Video Streams에서 발생하는 일반적인 문제를 해결할 수 있습니다.

주제

- [일반 문제](#)
- [API 문제](#)
- [HLS 문제](#)
- [Java 문제](#)
- [프로듀서 라이브러리 문제](#)
- [스트림 파서 라이브러리 문제](#)
- [네트워크 문제](#)

일반 문제

이 섹션에서는 Kinesis Video Streams를 사용할 때 발생할 수 있는 일반적인 문제에 대해 설명합니다.

문제

- [너무 긴 지연 시간](#)

너무 긴 지연 시간

Kinesis Video Streams 서비스로 전송되는 프래그먼트의 지속 시간으로 인해 지연 시간이 발생할 수 있습니다. 생산자와 서비스 사이의 지연 시간을 줄이는 한 가지 방법은 조각 지속 시간을 단축하도록 미디어 파이프라인을 구성하는 것입니다.

각 프래그먼트에서 전송되는 프레임 수를 줄이려면 다음 값을 줄여야 합니다.

kinesis_video_gstreamer_sample_app.cpp

```
g_object_set(G_OBJECT (data.encoder), "bframes", 0, "key-int-max", 45, "bitrate", 512, NULL);
```

Note

비디오 렌더링의 내부 구현으로 인해 Mozilla Firefox 브라우저의 지연 시간이 더 길니다.

API 문제

이 섹션에서는 Kinesis Video Streams를 사용할 때 발생할 수 있는 API 문제를 설명합니다.

문제

- 오류: "[알 수 없는 옵션](#)"
- 오류: "[Unable to determine service/operation name to be authorized](#)"(권한을 부여할 서비스/작업 이름을 확인할 수 없음)
- 오류: "[Failed to put a frame in the stream](#)"(스트림에 프레임을 넣지 못했음)
- 오류: "[최종 연결이 수신되기 전에 서비스가 연결을 AckEvent 끊었습니다.](#)"
- 오류: "[STATUS_STORE_OUT_OF_MEMORY](#)"

오류: "알 수 없는 옵션"

GetMedia 및 GetMediaForFragmentList가 다음 오류로 인해 실패할 수 있습니다.

```
Unknown options: <filename>.mkv
```

이 오류는 `output` 유형으로 AWS CLI 구성한 경우 발생합니다. `json` 를 기본 출력 유형 (`none`) AWS CLI 으로 재구성하십시오. 구성에 대한 자세한 내용은 AWS CLI 명령 참조의 [구성](#)을 참조하십시오. AWS CLI

오류: "Unable to determine service/operation name to be authorized"(권한을 부여할 서비스/작업 이름을 확인할 수 없음)

GetMedia가 다음 오류로 인해 실패할 수 있습니다.

```
Unable to determine service/operation name to be authorized
```

이 오류는 엔드포인트가 제대로 지정되지 않은 경우 발생할 수 있습니다. 엔드포인트를 가져올 때는 호출 API에 따라 `GetDataEndpoint` 호출에 다음 파라미터를 포함해야 합니다.

```
--api-name GET_MEDIA
--api-name PUT_MEDIA
--api-name GET_MEDIA_FOR_FRAGMENT_LIST
--api-name LIST_FRAGMENTS
```

오류: "Failed to put a frame in the stream"(스트림에 프레임을 넣지 못했음)

PutMedia가 다음 오류로 인해 실패할 수 있습니다.

```
Failed to put a frame in the stream
```

이 오류는 서비스에서 연결 또는 권한을 사용할 수 없는 경우 발생할 수 있습니다. 에서 AWS CLI 다음을 실행하고 스트림 정보를 검색할 수 있는지 확인합니다.

```
aws kinesismedia describe-stream --stream-name StreamName --endpoint https://ServiceEndpoint.kinesismedia.region.amazonaws.com
```

호출이 실패할 경우 자세한 내용은 [AWS CLI 오류 문제 해결](#)을 참조하십시오.

오류: “최종 연결이 수신되기 전에 서비스가 연결을 AckEvent 끝냈습니다.”

PutMedia가 다음 오류로 인해 실패할 수 있습니다.

```
com.amazonaws.SdkClientException: Service closed connection before final AckEvent was received
```

이 오류는 PushbackInputStream이 올바르게 구현되지 않을 경우 발생할 수 있습니다. unread() 메서드가 올바르게 구현되었는지 확인하십시오.

오류: "STATUS_STORE_OUT_OF_MEMORY"

PutMedia가 다음 오류로 인해 실패할 수 있습니다.

```
The content store is out of memory.
```

이 오류는 콘텐츠 스토어가 충분한 크기로 할당되지 않을 때 나타납니다. 콘텐츠 스토어 크기를 늘리려면 StorageInfo.storageSize 값을 증가시킵니다. 자세한 정보는 [StorageInfo](#)을 참조하세요.

HLS 문제

비디오 스트림이 제대로 재생되지 않는 경우 [이 섹션을 참조하십시오](#) [the section called “HLS 문제 해결”](#).

Java 문제

이 섹션에서는 Kinesis Video Streams를 사용할 때 발생하는 일반적인 Java 문제를 해결하는 방법을 설명합니다.

문제

- [Java 로그 활성화](#)

Java 로그 활성화

Java 샘플 및 라이브러리 문제를 해결하려면 디버그 로그를 활성화하고 검사하는 것이 좋습니다. 디버그 로그를 활성화하려면 다음을 수행합니다.

1. log4j을 pom.xml 노드에 있는 dependencies 파일에 추가합니다.

```
<dependency>
  <groupId>log4j</groupId>
  <artifactId>log4j</artifactId>
  <version>1.2.17</version>
</dependency>
```

2. target/classes 디렉터리에 다음 콘텐츠로 log4j.properties라는 파일을 생성합니다.

```
# Root logger option
log4j.rootLogger=DEBUG, stdout

# Redirect log messages to console
log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.Target=System.out
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %-5p %c{1}:
%L - %m%n

log4j.logger.org.apache.http.wire=DEBUG
```

그러면 디버그 로그가 IDE 콘솔로 인쇄됩니다.

프로듀서 라이브러리 문제

이 단원에서는 [생산자 라이브러리](#) 사용 시 발생할 수 있는 문제에 대해 설명합니다.

문제

- [생산자 SDK를 컴파일할 수 없음](#)
- [비디오 스트림이 콘솔에 나타나지 않음](#)
- [오류: "GStreamer 데모 애플리케이션을 사용하여 데이터를 스트리밍할 때 "요청에 포함된 보안 토큰이 잘못되었음"](#)
- [오류: "Kinesis 비디오 클라이언트에 프레임을 제출하지 못 함"](#)
- [OS X에 "스트리밍 중지됨, 이유가 협상되지 않음" 메시지와 함께 GStreamer 애플리케이션이 중지됨](#)
- [오류: Raspberry Pi 기반으로 GStreamer 데모에서 Kinesis 비디오 클라이언트를 만들 때 "힙 할당에 실패했음"](#)
- [오류: Raspberry Pi 기반으로 GStreamer 데모를 실행하는 동안 "잘못된 명령"](#)
- [카메라가 Raspberry Pi에서 로드하지 못함](#)
- [macOS High Sierra에서 카메라를 찾을 수 없음](#)
- [macOS High Sierra에서 컴파일 시 jni.h 파일을 찾을 수 없음](#)
- [GStreamer 데모 애플리케이션 실행 시 Curl 오류 발생](#)
- [Raspberry Pi에서 런타임 시 타임스탬프/범위 어설션](#)
- [Raspberry Pi에서 gst_value_set_fraction_range_full에 어설션](#)
- [Android에서 STATUS_MKV_INVALID_ANNEXB_NALU_IN_FRAME_DATA\(0x3200000d\) 오류 발생](#)
- [최대 프래그먼트 지속 시간에 도달했습니다. 오류](#)
- [IoT 권한 부여를 사용할 때 "Invalid thing name passed\(잘못된 이름이 전달되었습니다\)" 오류 발생](#)

생산자 SDK를 컴파일할 수 없음

필요한 라이브러리가 경로에 있는지 확인합니다. 이를 확인하려면 다음 명령을 사용합니다.

```
env | grep LD_LIBRARY_PATH
LD_LIBRARY_PATH=/home/local/awslabs/amazon-kinesis-video-streams-producer-sdk-cpp/
kinesis-video-native-build/downloads/local/lib
```

비디오 스트림이 콘솔에 나타나지 않음

콘솔에 비디오 스트림을 표시하려면 AvCC 형식으로 H.264를 사용하여 인코딩해야 합니다. 스트림이 보이지 않으면 다음을 확인합니다.

- 기존 스트림이 Annex-B 형식에 있으면 [NAL 적응 플래그](#)는 NAL_ADAPTATION_ANNEXB_NALS | NAL_ADAPTATION_ANNEXB_CPD_NALS로 설정됩니다. 이는 StreamDefinition 생성자의 기본 값입니다.
- 코덱 프라이빗 데이터를 올바르게 제공하고 있습니다. H.264의 경우 SPS(Sequence Parameter Set) 및 PPS(Picture Parameter Set)입니다. 미디어 원본에 따라 이 데이터를 별도로 미디어 원본에서 검색하거나 프레임에 인코딩할 수 있습니다.

초기 스트림은 다음 형식이며 Ab가 Annex-B 시작 코드(001 혹은 0001)입니다.

```
Ab(Sps)Ab(Pps)Ab(I-frame)Ab(P/B-frame) Ab(P/B-frame)... Ab(Sps)Ab(Pps)Ab(I-frame)Ab(P/B-frame) Ab(P/B-frame)
```

H.264가 SPS 및 PPS로 스트림에 있는 경우 CPD (코덱 개인 데이터) 를 avCC 형식에 맞게 조정할 수 있습니다. 미디어 파이프라인이 CPD를 별도로 제공하지 않는 한, 애플리케이션은 첫 번째 Idr 프레임 (SPS와 PPS를 포함해야 함) 을 찾아 프레임에서 CPD를 추출하고 두 개의 NALU (Ab (Sps) Ab (Pps)) 를 추출한 다음 이를 CPD에 설정할 수 StreamDefinition 있습니다.

오류: "GStreamer 데모 애플리케이션을 사용하여 데이터를 스트리밍할 때 "요청에 포함된 보안 토큰이 잘못되었음"

이 오류가 발생하면 자격 증명에 문제가 있는 것입니다. 다음을 확인합니다.

- 임시 자격 증명을 사용하는 경우 세션 토큰을 지정해야 합니다.
- 임시 자격 증명 만료되지 않았는지 확인합니다.
- 적절한 권한이 설정되어 있는지 확인합니다.
- macOS에서 키체인에 자격 증명 이 캐시되어 있지 않은지 확인합니다.

오류: "Kinesis 비디오 클라이언트에 프레임을 제출하지 못 함"

이 오류가 발생하면 타임스탬프가 소스 스트림에 제대로 설정되지 않은 것입니다. 다음을 시도해 보십시오.

- 최신 SDK 샘플을 사용합니다. 문제를 해결하는 업데이트가 있을 수 있습니다.
- 고품질 스트림을 더 높은 비트레이트로 설정하고 카메라가 지원하는 경우 소스 스트림의 지터를 수정합니다.

OS X에 "스트리밍 중지됨, 이유가 협상되지 않음" 메시지와 함께 GStreamer 애플리케이션이 중지됨

다음 메시지와 함께 OS X에서 스트리밍이 중단될 수 있음:

```
Debugging information: gstbasesrc.c(2939): void gst_base_src_loop(GstPad *) (): /
GstPipeline:test-pipeline/GstAutoVideoSrc:source/GstAVFVideoSrc:source-actual-src-
avfvide:
streaming stopped, reason not-negotiated (-4)
```

이 문제를 해결할 수 있는 방법은 호출에서 프레임 속도 매개변수를 제거하는 것입니다.

gst_caps_new_simple kinesis_video_gstreamer_sample_app.cpp

```
GstCaps *h264_caps = gst_caps_new_simple("video/x-h264",
                                         "profile", G_TYPE_STRING, "baseline",
                                         "stream-format", G_TYPE_STRING, "avc",
                                         "alignment", G_TYPE_STRING, "au",
                                         "width", GST_TYPE_INT_RANGE, 320, 1920,
                                         "height", GST_TYPE_INT_RANGE, 240, 1080,
                                         "framerate", GST_TYPE_FRACTION_RANGE, 0,
                                         1, 30, 1,
                                         NULL);
```

오류: Raspberry Pi 기반으로 GStreamer 데모에서 Kinesis 비디오 클라이언트를 만들 때 "힙 할당에 실패했음"

GStreamer 샘플 애플리케이션이 시스템에서 사용할 수 없는 512MB RAM을 할당하려고 합니다. KinesisVideoProducer.cpp에서 다음 값을 줄임으로써 이 할당을 줄일 수 있습니다.

```
device_info.storageInfo.storageSize = 512 * 1024 * 1024;
```


오류: Raspberry Pi 기반으로 GStreamer 데모를 실행하는 동안 "잘못된 명령"

GStreamer 데모를 실행할 때 다음 오류가 발생하는 경우 애플리케이션이 올바른 기기 버전에 맞게 컴파일되었는지 확인하세요. (예를 들어, 라즈베리 파이 2에서 실행 중일 때는 라즈베리 파이 3용으로 컴파일하고 있지 않은지 확인하세요.)

```
INFO - Initializing curl.  
Illegal instruction
```

카메라가 Raspberry Pi에서 로드하지 못함

카메라가 로드되었는지 확인하려면 다음을 실행합니다.

```
ls /dev/video*
```

아무 것도 없으면 다음을 실행합니다.

```
vcgencmd get_camera
```

다음과 같이 출력됩니다

```
supported=1 detected=1
```

드라이버가 카메라를 감지하지 못하면 다음을 수행합니다.

1. 실제 카메라 설정을 확인하고 올바르게 연결되었는지 확인하십시오.
2. 다음을 실행하여 펌웨어를 업그레이드합니다.

```
sudo rpi-update
```

3. 디바이스를 다시 시작합니다.
4. 다음을 실행하여 드라이버를 로드합니다.

```
sudo modprobe bcm2835-v4l2
```

5. 카메라가 감지되었는지 확인합니다.

```
ls /dev/video*
```

macOS High Sierra에서 카메라를 찾을 수 없음

macOS High Sierra에서 두 대 이상의 카메라를 사용할 수 있는 경우 데모 애플리케이션이 카메라를 찾을 수 없습니다.

macOS High Sierra에서 컴파일 시 jni.h 파일을 찾을 수 없음

이 오류를 해결하려면 설치한 Xcode를 최신 버전으로 업데이트합니다.

GStreamer 데모 애플리케이션 실행 시 Curl 오류 발생

GStreamer 데모 애플리케이션 실행 시 발생한 curl 오류를 실행하려면 [이 인증서 파일](#)을 /etc/ssl/cert.pem으로 복사합니다.

Raspberry Pi에서 런타임 시 타임스탬프/범위 어설션

런타임 시 타임스탬프 범위 어설션이 발생하면 펌웨어를 업데이트하고 디바이스를 다시 시작합니다.

```
sudo rpi-update
$ sudo reboot
```

Raspberry Pi에서 gst_value_set_fraction_range_full에 어설션

uv4l 서비스가 실행 중인 경우 다음 어설션이 나타납니다.

```
gst_util_fraction_compare (numerator_start, denominator_start, numerator_end,
denominator_end) < 0' failed
```

이 경우 uv4l 서비스를 중지하고 애플리케이션을 다시 시작합니다.

Android에서

STATUS_MKV_INVALID_ANNEXB_NALU_IN_FRAME_DATA(0x3200000d) 오류 발생

[NAL 적응 플래그](#)가 미디어 스트림에 대해 부정확할 경우 다음 오류가 나타납니다.

```
putKinesisVideoFrame(): Failed to put a frame with status code 0x3200000d
```

이 오류가 발생하면 해당 미디어(예: NAL_ADAPTATION_ANNEXB_CPD_NALS)에 정확한 `.withNalAdaptationFlags` 플래그를 제공하십시오. 이 플래그를 [Android 생산자 라이브러리](#)의 다음 행에 제공합니다.

<https://github.com/aws-labs/aws-sdk-android-samples/blob/master/src/main/java/com/amazonaws/kinesisvideo/demoapp/fragment/.java#L169> `AmazonKinesisVideoDemoAppStreamConfigurationFragment`

최대 프래그먼트 지속 시간에 도달했습니다. 오류

이 오류는 스트림의 미디어 조각이 최대 조각 기간 한도를 초과할 때 발생합니다. 섹션의 최대 프래그먼트 지속 시간 제한을 참조하십시오. [the section called “미디어 및 아카이브 미디어 API 서비스 할당량”](#)

이 문제를 해결하려면 다음과 같이 해 보십시오.

- 웹캠/USB 카메라를 사용하는 경우, 다음 중 하나를 수행하십시오.
 - 키프레임 기반 프래그먼트화를 사용하는 경우 10초 내에 키프레임을 제공하도록 인코더를 설정하십시오.
 - 키프레임 기반 프래그먼트화를 사용하지 않는 경우에서 [2단계: 코드 작성 및 검토](#) 스트림을 정의할 때 최대 프래그먼트 지속 시간 제한을 10초 미만의 값으로 설정하십시오.
 - GStreamer 파이프라인에서 소프트웨어 인코더 (예: x264) 를 사용하는 경우 10초 이내에 속성을 값으로 설정할 수 있습니다. `key-int-max` 예를 들어 60으로 설정하고 `key-int-max fps`를 30으로 설정하면 2초마다 키 프레임이 활성화됩니다.
- RPI 카메라를 사용하는 경우 키프레임 간격 속성을 10초 미만으로 설정하십시오.
- IP (RTSP) 카메라를 사용하는 경우 GOP 크기를 60으로 설정하십시오.

IoT 권한 부여를 사용할 때 "Invalid thing name passed(잘못된 이름이 전달되었습니다)" 오류 발생

권한 부여에 IoT 자격 증명을 사용할 때 이 오류 (HTTP Error 403: Response: {"message": "Invalid thing name passed"}) 를 방지하려면 `stream-name` (kvssink요소의 필수 매개변수) 값이 의 값과 동일한지 확인하십시오 `iot-thingname`. 자세한 정보는 [GStreamer 요소 파라미터 레퍼런스](#)을 참조하세요.

스트림 파서 라이브러리 문제

이 단원에서는 [스트림 구문 분석기 라이브러리](#) 사용 시 발생할 수 있는 문제에 대해 설명합니다.

문제

- [스트림에서 단일 프레임에 액세스할 수 없음](#)
- [조각 디코딩 오류](#)

스트림에서 단일 프레임에 액세스할 수 없음

소비자 애플리케이션의 스트리밍 소스에서 단일 프레임에 액세스하려면 스트림에 올바른 코덱 프라이빗 데이터가 포함되어 있는지 확인하세요. 스트림의 데이터 형식에 대한 자세한 내용은 [데이터 모델](#) 단원을 참조하십시오.

[코덱 프라이빗 데이터를 사용하여 프레임에 액세스하는 방법을 알아보려면 GitHub 웹 사이트에 있는 다음 테스트 파일을 참조하십시오. `.java KinesisVideoRendererExampleTest`](#)

조각 디코딩 오류

조각이 브라우저에서 지원하는 H.264 형식 및 레벨로 올바르게 인코딩되지 않은 경우 콘솔에서 스트림을 재생할 때 다음 오류가 표시될 수 있습니다.

```
Fragment Decoding Error
There was an error decoding the video data. Verify that the stream contains valid H.264 content
```

이 경우 다음을 확인합니다.

- 프레임의 해상도는 코덱 프라이빗 데이터에 지정된 해상도와 일치합니다.
- 인코딩 프레임의 H.264 프로파일 및 레벨은 코덱 프라이빗 데이터에 지정된 프로파일 및 레벨과 일치합니다.
- 브라우저는 프로파일/레벨 조합을 지원합니다. 최신 브라우저는 모든 프로파일 및 레벨 조합을 지원합니다.
- 타임스탬프는 정확하고 순서가 올바르며 중복 타임스탬프가 생성되지 않습니다.
- 애플리케이션이 H.264 형식을 사용하여 프레임 데이터를 인코딩하고 있습니다.

네트워크 문제

Kinesis Video Streams에 연결하려고 할 때 “연결 시간 초과” 또는 “연결 실패”와 같은 연결 오류가 표시되면 네트워킹 설정의 IP 주소 범위 제한 때문일 수 있습니다.

[설정에 Kinesis Video Streams에 대한 IP 주소 범위 제한이 있는 경우 Kinesis Video Streams IP 주소 범위를 허용 목록에 포함하도록 네트워크 구성을 업데이트하십시오.](#)

[자세한 내용은 IP 범위를 참조하십시오AWS](#). IP 범위가 변경될 때 알림을 받으려면 [구독 절차를](#) 따르십시오.

Amazon Kinesis Video Streams 문서 기록

다음 표에는 Amazon Kinesis Video Streams의 마지막 릴리스 이후 설명서에 대한 중요한 변경 사항이 설명되어 있습니다.

- 최신 API 버전: 2017-11-29
- 최신 설명서 업데이트: 2023년 6월 27일

변경 사항	설명	날짜
Amazon Kinesis Video Streams 엣지 에이전트 엣지 투 클라우드 연결	새 기능 출시. 자세한 정보는 엣지 에이전트 을 참조하세요.	2023년 6월 27일
시작하기: Kinesis 비디오 스트림으로 데이터 전송	카메라에서 Kinesis 비디오 스트림으로 미디어 데이터를 전송하는 방법에 대한 기본 자습서입니다. 자세한 정보는 Amazon Kinesis 비디오 스트림으로 데이터 전송 을 참조하세요.	2019년 1월 21일
스트리밍 메타데이터	Producer SDK를 사용하여 Kinesis 비디오 스트림에 메타데이터를 포함할 수 있습니다. 자세한 정보는 Kinesis Video Streams에서 스트리밍 메타데이터 사용 을 참조하세요.	2018년 9월 28일
C++ 생산자 SDK 로깅	C++ 생산자 SDK 애플리케이션에 대한 로깅을 구성할 수 있습니다. 자세한 정보는 C++ 프로듀서 SDK를 통한 로깅 사용 을 참조하세요.	2018년 7월 18일
HLS 비디오 스트리밍	이제 HTTP 라이브 스트리밍을 사용하여 Kinesis 비디오 스트	2018년 7월 13일

변경 사항	설명	날짜
	<p>림을 볼 수 있습니다. 자세한 정보는 Kinesis Video Streams 재생을 참조하세요.</p>	
<p>RTSP 소스에서 스트리밍</p>	<p>Docker 컨테이너에서 실행되고 RTSP 소스에서 비디오를 스트리밍하는 Kinesis Video Streams용 샘플 애플리케이션입니다. 자세한 정보는 RTSP 및 도커을 참조하세요.</p>	<p>2018년 6월 20일</p>
<p>C++ 생산자 SDK GStreamer 플러그인</p>	<p>C++ 생산자 라이브러리를 구축하여 GStreamer 대상으로 사용하는 방법을 보여 줍니다. 자세한 정보는 스트리머 플러그인 - kvssink을 참조하세요.</p>	<p>2018년 15월 6일</p>
<p>생산자 SDK 콜백 참조 설명서</p>	<p>Kinesis 비디오 스트림 프로듀서 라이브러리에 사용되는 콜백에 대한 참조 문서입니다. 자세한 정보는 프로듀서 SDK 콜백을 참조하세요.</p>	<p>2018년 6월 12일</p>
<p>시스템 요구 사항</p>	<p>생산자 디바이스와 SDK의 메모리 및 스토리지 요구 사항 설명서. 자세한 정보는 Kinesis Video Streams 시스템 요구 사항을 참조하세요.</p>	<p>2018년 5월 30일</p>
<p>CloudTrail 지원</p>	<p>API 사용을 모니터링하는 CloudTrail 데 사용하는 설명서. 자세한 정보는 클 사용하여 Amazon Kinesis Video Streams API 호출을 로깅합니다. AWS CloudTrail을 참조하세요.</p>	<p>2018년 5월 24일</p>

변경 사항	설명	날짜
생산자 SDK 구조 참조 설명서	Kinesis 비디오 스트림 프로듀서 라이브러리 에 사용된 구조 참조 문서 자세한 내용은 프로듀서 SDK 구조 및 Kinesis 비디오 스트림 구조 섹션을 참조하세요.	2018년 5월 7일
렌더러 예제 설명서	Kinesis 비디오 스트림에서 프레임을 디코딩하고 표시하는 방법을 보여주는 렌더러 예제 애플리케이션 설명서. 자세한 정보는 예: Kinesis Video Streams 프래그먼트의 파싱 및 렌더링 을 참조하세요.	2018년 3월 15일
생산자 SDK 제한 사항 참조 설명서	Information about limits for operations in the C++ 생산자 라이브러리 에는 작업에 대한 제한 사항 정보가 수록되어 있습니다. 자세한 정보는 프로듀서 SDK 한도 을 참조하세요.	2018년 3월 13일
모니터링	CloudWatch AWS CloudTrail Amazon을 사용한 Kinesis Video Streams 지표 및 API 호출 모니터링에 대한 정보 및 자세한 정보는 아마존 키네시스 비디오 스트림 모니터링 을 참조하세요.	2018년 2월 5일
NAL(Network Abstraction Layer) 적응 플래그 참조	스트리밍 비디오 이용 시 NAL 적응 플래그 설정에 대한 정보입니다. 자세한 정보는 NAL 적응 플래그 을 참조하세요.	2018년 1월 15일

변경 사항	설명	날짜
스트리밍 비디오에 대한 Android 지원	Kinesis Video Streams는 이제 안드로이드 디바이스에서의 스트리밍 비디오를 지원합니다. 자세한 정보는 Android 생산자 라이브러리 을 참조하세요.	2018년 1월 12일
Kinesis 비디오 예제 문서	애플리케이션에서 를 사용하는 방법을 보여주는 Kinesis Video 예제 애플리케이션 설명서. Kinesis 비디오 스트림 파서 라이브러리 자세한 정보는 KinesisVideoExample 을 참조하세요.	2018년 1월 9일
Kinesis Video Streams 설명서 공개	이것은 Amazon Kinesis Video Streams 개발자 안내서의 최초 릴리스입니다.	2017년 11월 29일

API 참조

이 노트 아래의 섹션에는 API 참조 설명서가 포함되어 있습니다. 왼쪽 창의 목차를 사용하여 다른 API 참조 섹션으로 이동할 수 있습니다.

작업

Amazon Kinesis 비디오 스트림은 다음 작업을 지원합니다.

- [CreateSignalingChannel](#)
- [CreateStream](#)
- [DeleteEdgeConfiguration](#)
- [DeleteSignalingChannel](#)
- [DeleteStream](#)
- [DescribeEdgeConfiguration](#)
- [DescribeImageGenerationConfiguration](#)
- [DescribeMappedResourceConfiguration](#)
- [DescribeMediaStorageConfiguration](#)
- [DescribeNotificationConfiguration](#)
- [DescribeSignalingChannel](#)
- [DescribeStream](#)
- [GetDataEndpoint](#)
- [GetSignalingChannelEndpoint](#)
- [ListEdgeAgentConfigurations](#)
- [ListSignalingChannels](#)
- [ListStreams](#)
- [ListTagsForResource](#)
- [ListTagsForStream](#)
- [StartEdgeConfigurationUpdate](#)
- [TagResource](#)
- [TagStream](#)
- [UntagResource](#)

- [UntagStream](#)
- [UpdateDataRetention](#)
- [UpdateImageGenerationConfiguration](#)
- [UpdateMediaStorageConfiguration](#)
- [UpdateNotificationConfiguration](#)
- [UpdateSignalingChannel](#)
- [UpdateStream](#)

Amazon Kinesis 비디오 스트림 미디어는 다음 작업을 지원합니다.

- [GetMedia](#)
- [PutMedia](#)

Amazon Kinesis 비디오 스트림 아카이브 미디어는 다음 작업을 지원합니다.

- [GetClip](#)
- [GetDASHStreamingSessionURL](#)
- [GetHLSStreamingSessionURL](#)
- [GetImages](#)
- [GetMediaForFragmentList](#)
- [ListFragments](#)

Amazon Kinesis 비디오 시그널링 채널에서는 다음 작업을 지원합니다.

- [GetIceServerConfig](#)
- [SendAlexaOfferToMaster](#)

Amazon Kinesis Video WebRTC 스토리지는 다음 작업을 지원합니다.

- [JoinStorageSession](#)

Amazon Kinesis Video Streams

Amazon Kinesis 비디오 스트림은 다음 작업을 지원합니다.

- [CreateSignalingChannel](#)
- [CreateStream](#)
- [DeleteEdgeConfiguration](#)
- [DeleteSignalingChannel](#)
- [DeleteStream](#)
- [DescribeEdgeConfiguration](#)
- [DescribeImageGenerationConfiguration](#)
- [DescribeMappedResourceConfiguration](#)
- [DescribeMediaStorageConfiguration](#)
- [DescribeNotificationConfiguration](#)
- [DescribeSignalingChannel](#)
- [DescribeStream](#)
- [GetDataEndpoint](#)
- [GetSignalingChannelEndpoint](#)
- [ListEdgeAgentConfigurations](#)
- [ListSignalingChannels](#)
- [ListStreams](#)
- [ListTagsForResource](#)
- [ListTagsForStream](#)
- [StartEdgeConfigurationUpdate](#)
- [TagResource](#)
- [TagStream](#)
- [UntagResource](#)
- [UntagStream](#)
- [UpdateDataRetention](#)
- [UpdateImageGenerationConfiguration](#)
- [UpdateMediaStorageConfiguration](#)
- [UpdateNotificationConfiguration](#)
- [UpdateSignalingChannel](#)
- [UpdateStream](#)

CreateSignalingChannel

서비스: Amazon Kinesis Video Streams

신호 채널을 만듭니다.

CreateSignalingChannel은 비동기식 작업입니다.

Request Syntax

```
POST /createSignalingChannel HTTP/1.1
Content-type: application/json

{
  "ChannelName": "string",
  "ChannelType": "string",
  "SingleMasterConfiguration": {
    "MessageTtlSeconds": number
  },
  "Tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ]
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

ChannelName

생성하고 있는 신호 채널의 이름입니다. 각 AND마다 고유해야 합니다 AWS 계정 . AWS 리전

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256입니다.

패턴: [a-zA-Z0-9_.-]+

필수 사항 여부: Yes

ChannelType

생성하고 있는 신호 채널의 유형입니다. 현재는 SINGLE_MASTER 채널 유형만 지원됩니다.

타입: 문자열

유효 값: SINGLE_MASTER | FULL_MESH

필수 여부: 아니요

SingleMasterConfiguration

SINGLE_MASTER채널 유형에 대한 구성이 포함된 구조입니다.

유형: [SingleMasterConfiguration](#)객체

필수 항목 여부: 아니요

Tags

이 채널과 연결하려는 태그 세트 (키-값 쌍).

유형: [Tag](#)객체 어레이

배열 멤버: 최소수는 0개입니다. 최대수 50개.

필수 여부: 아니요

응답 구문

```
HTTP/1.1 200
Content-type: application/json

{
  "ChannelARN": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

ChannelARN

생성된 채널의 Amazon 리소스 이름 (ARN).

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 [일반적인 오류](#) 섹션을 참조하세요.

AccessDeniedException

이 작업을 수행하는 데 필요한 권한은 없습니다.

HTTP 상태 코드: 401

AccountChannelLimitExceededException

이 AWS 계정 지역에서 이에 대한 활성 신호 채널의 최대 한도에 도달했습니다.

HTTP 상태 코드: 400

ClientLimitExceededException

허용된 클라이언트 호출 한도를 초과했기 때문에 Kinesis Video Streams가 요청 병목 현상을 일으켰습니다. 나중에 전화를 걸어 보세요.

HTTP 상태 코드: 400

InvalidArgumentException

이 입력 매개변수의 값이 유효하지 않습니다.

HTTP 상태 코드: 400

ResourceInUseException

입력 StreamARN 또는 ChannelARN CLOUD_STORAGE_MODE 입력이 이미 다른 Kinesis Video Stream 리소스에 매핑되어 있거나 제공된 StreamARN 입력 ChannelARN 또는 입력이 활성 상태가 아닌 경우 다음 중 하나를 시도해 보십시오.

1. 해당 스트림의 채널이 매핑되는 대상을 결정하는 DescribeMediaStorageConfiguration API입니다.
2. 지정된 스트림이 매핑되는 채널을 결정하는 DescribeMappedResourceConfiguration API입니다.
3. 리소스 상태를 확인하기 위한 DescribeStream 또는 DescribeSignalingChannel API입니다.

HTTP 상태 코드: 400

TagsPerResourceExceededLimitException

리소스에 연결할 수 있는 태그의 제한을 초과했습니다. Kinesis 비디오 스트림은 최대 50개의 태그를 지원할 수 있습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS 파이썬용 SDK](#)
- [AWS 루비 V3용 SDK](#)

CreateStream

서비스: Amazon Kinesis Video Streams

새 Kinesis 비디오 스트림을 생성합니다.

새 스트림을 생성하면 Kinesis Video Streams에서 스트림에 버전 번호를 할당합니다. 스트림의 메타데이터를 변경하면 Kinesis Video Streams가 버전을 업데이트합니다.

CreateStream은 비동기식 작업입니다.

서비스 작동 방식에 대한 자세한 내용을 알아보려면 [작동 방식](#)을 참조하세요.

KinesisVideo:CreateStream 작업에 대한 권한이 있어야 합니다.

Request Syntax

```
POST /createStream HTTP/1.1
Content-type: application/json

{
  "DataRetentionInHours": number,
  "DeviceName": "string",
  "KmsKeyId": "string",
  "MediaType": "string",
  "StreamName": "string",
  "Tags": {
    "string" : "string"
  }
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

[DataRetentionInHours](#)

스트림에 데이터를 보존하려는 시간. Kinesis Video Streams는 스트림과 연결된 데이터 스토어에 데이터를 보관합니다.

기본값은 0이며, 스트림이 데이터를 유지하지 않음을 나타냅니다.

DataRetentionInHours값이 0인 경우에도 소비자는 보존 시간 제한이 5분이고 보존 메모리 제한이 200MB인 서비스 호스트 버퍼에 남아 있는 프래그먼트를 계속 사용할 수 있습니다. 두 한도 중 하나에 도달하면 프래그먼트가 버퍼에서 제거됩니다.

유형: 정수

유효 범위: 최소값 0.

필수 여부: 아니요

DeviceName

스트림에 쓰고 있는 디바이스의 이름.

Note

현재 구현에서는 Kinesis Video Streams가 이 이름을 사용하지 않습니다.

유형: 문자열

길이 제약: 최소 길이는 1. 최대 길이 128.

패턴: [a-zA-Z0-9_.-]+

Required: No

KmsKeyId

Kinesis Video Streams가 스트림 데이터를 암호화하는 데 사용할 AWS Key Management Service (AWS KMS) 키의 ID입니다.

키 ID가 지정되지 않은 경우 기본값인 Kinesis 비디오 관리 키 () AWS/kinesisvideo 가 사용됩니다.

자세한 내용은 을 참조하십시오. [DescribeKey](#)

유형: 문자열

길이 제약: 최소 길이는 1. 최대 길이는 2,048.

패턴: .+

Required: No

MediaType

스트림의 미디어 유형. 스트림 소비자는 스트림을 처리할 때 이 정보를 사용할 수 있습니다. 미디어 유형에 대한 자세한 내용은 [미디어 유형을 참조하십시오](#). 를 지정하기로 선택한 경우 지침은 [이름 지정 요구 사항을 참조하십시오](#). MediaType

유효한 값의 예로는 "비디오/h264" 및 "비디오/h264, 오디오/aac"가 있습니다.

이 매개변수는 선택 사항이며 기본값은 (JSON에서는 비어 있음) 입니다. null

유형: 문자열

길이 제약: 최소 길이는 1. 최대 길이 128.

패턴: `[\w\-\.\.]+/[\w\-\.\.]+(,[\w\-\.\.]+/[\w\-\.\.]+)*`

Required: No

StreamName

생성 중인 스트림의 이름.

스트림 이름은 스트림의 식별자이며 각 계정 및 지역별로 고유해야 합니다.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256입니다.

패턴: `[a-zA-Z0-9_.-]+`

필수 사항 여부: Yes

Tags

지정된 스트림과 연결할 태그 목록입니다. 각 태그는 키-값 쌍입니다 (값은 선택 사항).

유형: 문자열-문자열 맵

맵 항목: 최대 항목 수는 50입니다.

키 길이 제약 조건: 최소 길이는 1. 최대 길이 128.

키 패턴: `^([\p{L}\p{Z}\p{N}_.:/+@-]*)$`

값 길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 256.

값 패턴: `[\p{L}\p{Z}\p{N}_.:/+@-]*`

필수 여부: 아니요

응답 구문

```
HTTP/1.1 200
Content-type: application/json

{
  "StreamARN": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

StreamARN

스트림의 Amazon 리소스 이름(ARN)입니다.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 [일반적인 오류](#) 섹션을 참조하세요.

AccountStreamLimitExceededException

계정에 생성된 스트림 수가 너무 많습니다.

HTTP 상태 코드: 400

ClientLimitExceededException

허용된 클라이언트 호출 한도를 초과했기 때문에 Kinesis Video Streams가 요청 병목 현상을 일으켰습니다. 나중에 전화를 걸어 보세요.

HTTP 상태 코드: 400

DeviceStreamLimitExceededException

구현되지 않음.

HTTP 상태 코드: 400

InvalidArgumentException

이 입력 매개변수의 값이 유효하지 않습니다.

HTTP 상태 코드: 400

InvalidDeviceException

구현되지 않음.

HTTP 상태 코드: 400

ResourceInUseException

입력 StreamARN 또는 ChannelARN CLOUD_STORAGE_MODE 입력이 이미 다른 Kinesis Video Stream 리소스에 매핑되어 있거나 제공된 StreamARN 입력 ChannelARN 또는 입력이 활성 상태가 아닌 경우 다음 중 하나를 시도해 보십시오.

1. 해당 스트림의 채널이 매핑되는 대상을 결정하는 DescribeMediaStorageConfiguration API입니다.
2. 지정된 스트림이 매핑되는 채널을 결정하는 DescribeMappedResourceConfiguration API입니다.
3. 리소스 상태를 확인하기 위한 DescribeStream 또는 DescribeSignalingChannel API입니다.

HTTP 상태 코드: 400

TagsPerResourceExceededLimitException

리소스에 연결할 수 있는 태그의 제한을 초과했습니다. Kinesis 비디오 스트림은 최대 50개의 태그를 지원할 수 있습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DeleteEdgeConfiguration

서비스: Amazon Kinesis Video Streams

스트림의 기존 에지 구성과 Edge Agent의 해당 미디어를 삭제하는 비동기 API입니다.

이 API를 호출하면 동기화 상태가 `DELETING` 로 설정됩니다. `DELETING` 활성 에지 작업이 중지되고 에지 디바이스에서 모든 미디어가 삭제되는 삭제 프로세스가 시작됩니다. 삭제 시간은 저장된 미디어의 총량에 따라 달라집니다. 삭제 프로세스가 실패하면 동기화 상태가 `DELETE_FAILED` 로 바뀝니다. 삭제를 다시 시도해야 합니다.

삭제 프로세스가 성공적으로 완료되면 에지 구성에 더 이상 액세스할 수 없습니다.

Note

이 API는 AWS 아프리카 (케이프타운) 지역인 `af-south-1`에서는 사용할 수 없습니다.

Request Syntax

```
POST /deleteEdgeConfiguration HTTP/1.1
Content-type: application/json

{
  "StreamARN": "string",
  "StreamName": "string"
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

StreamARN

스트림의 Amazon 리소스 이름(ARN)입니다. 또는 `StreamName` StreamARN

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

StreamName

에지 구성을 삭제할 스트림의 이름. StreamName 또는 를 지정합니다 StreamARN.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256입니다.

패턴: `[a-zA-Z0-9_.-]+`

필수 여부: 아니요

응답 구분

```
HTTP/1.1 200
```

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 반환합니다.

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 [일반적인 오류](#) 섹션을 참조하세요.

AccessDeniedException

이 작업을 수행하는 데 필요한 권한은 없습니다.

HTTP 상태 코드: 401

ClientLimitExceededException

허용된 클라이언트 호출 한도를 초과했기 때문에 Kinesis Video Streams가 요청 병목 현상을 일으켰습니다. 나중에 전화를 걸어 보세요.

HTTP 상태 코드: 400

InvalidArgumentException

이 입력 매개변수의 값이 유효하지 않습니다.

HTTP 상태 코드: 400

ResourceNotFoundException

Amazon Kinesis Video Streams에서 지정한 스트림을 찾을 수 없습니다.

HTTP 상태 코드: 404

StreamEdgeConfigurationNotFoundException

Amazon Kinesis Video Stream이 지정한 스트림의 엣지 구성을 찾을 수 없는 경우 예외가 렌더링됩니다.

HTTP 상태 코드: 404

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DeleteSignalingChannel

서비스: Amazon Kinesis Video Streams

지정된 신호 채널을 삭제합니다. DeleteSignalingChannel 비동기 작업입니다. 채널의 현재 버전을 지정하지 않으면 가장 최신 버전이 삭제됩니다.

Request Syntax

```
POST /deleteSignalingChannel HTTP/1.1
Content-type: application/json

{
  "ChannelARN": "string",
  "CurrentVersion": "string"
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

ChannelARN

삭제하려는 신호 채널의 Amazon 리소스 이름 (ARN)

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+

필수 사항 여부: Yes

CurrentVersion

삭제하려는 신호 채널의 현재 버전. DescribeSignalingChannel 또는 ListSignalingChannels API 작업을 호출하여 현재 버전을 가져올 수 있습니다.

유형: 문자열

길이 제한: 최소 길이는 1. 최대 길이는 64.

패턴: [a-zA-Z0-9]+

필수 여부: 아니요

응답 구문

```
HTTP/1.1 200
```

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 반환합니다.

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 [일반적인 오류](#) 섹션을 참조하세요.

AccessDeniedException

이 작업을 수행하는 데 필요한 권한은 없습니다.

HTTP 상태 코드: 401

ClientLimitExceededException

허용된 클라이언트 호출 한도를 초과했기 때문에 Kinesis Video Streams가 요청 병목 현상을 일으켰습니다. 나중에 전화를 걸어 보세요.

HTTP 상태 코드: 400

InvalidArgumentException

이 입력 매개변수의 값이 유효하지 않습니다.

HTTP 상태 코드: 400

ResourceInUseException

입력 StreamARN 또는 ChannelARN CLOUD_STORAGE_MODE 입력이 이미 다른 Kinesis Video Stream 리소스에 매핑되어 있거나 제공된 StreamARN 입력 ChannelARN 또는 입력이 활성 상태가 아닌 경우 다음 중 하나를 시도해 보십시오.

1. 해당 스트림의 채널이 매핑되는 대상을 결정하는 DescribeMediaStorageConfiguration API입니다.

- 지정된 스트림이 매핑되는 채널을 결정하는 `DescribeMappedResourceConfiguration` API입니다.
- 리소스의 상태를 확인하기 위한 `DescribeStream` 또는 `DescribeSignalingChannel` API입니다.

HTTP 상태 코드: 400

ResourceNotFoundException

Amazon Kinesis Video Streams에서 지정한 스트림을 찾을 수 없습니다.

HTTP 상태 코드: 404

VersionMismatchException

지정한 스트림 버전은 최신 버전이 아닙니다. 최신 버전을 가져오려면 [DescribeStream](#) API를 사용하세요.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DeleteStream

서비스: Amazon Kinesis Video Streams

Kinesis 비디오 스트림과 스트림에 포함된 데이터를 삭제합니다.

이 메서드는 스트림을 삭제 대상으로 표시하고 스트림의 데이터에 즉시 액세스할 수 없게 만듭니다.

스트림을 삭제하기 전에 스트림의 최신 버전을 사용할 수 있도록 스트림 버전을 지정할 수 있습니다. Kinesis Video Streams는 각 스트림에 버전을 할당합니다. 스트림을 업데이트하면 Kinesis Video Streams가 새 버전 번호를 할당합니다. 최신 스트림 버전을 가져오려면 API를 사용하십시오.

DescribeStream

이 작업에는 KinesisVideo:DeleteStream 작업에 대한 권한이 필요합니다.

Request Syntax

```
POST /deleteStream HTTP/1.1
Content-type: application/json

{
  "CurrentVersion": "string",
  "StreamARN": "string"
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

CurrentVersion

선택 사항: 삭제하려는 스트림의 버전.

올바른 스트림을 삭제하고 있는지 확인할 수 있는 보호 수단으로 버전을 지정하십시오. 스트림 버전을 가져오려면 DescribeStream API를 사용하세요.

지정하지 않으면 스트림을 삭제하기 전에 에만 체크합니다. CreationTime

유형: 문자열

길이 제한: 최소 길이는 1. 최대 길이는 64.

패턴: [a-zA-Z0-9]+

Required: No

StreamARN

삭제하려는 스트림의 Amazon 리소스 이름 (ARN)

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+

필수 항목 여부: 예

응답 구문

```
HTTP/1.1 200
```

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 반환합니다.

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 [일반적인 오류](#) 섹션을 참조하세요.

ClientLimitExceededException

허용된 클라이언트 호출 한도를 초과했기 때문에 Kinesis Video Streams가 요청 병목 현상을 일으켰습니다. 나중에 전화를 걸어 보세요.

HTTP 상태 코드: 400

InvalidArgumentException

이 입력 매개변수의 값이 유효하지 않습니다.

HTTP 상태 코드: 400

NotAuthorizedException

호출자는 이 작업을 수행할 권한이 없습니다.

HTTP 상태 코드: 401

ResourceInUseException

입력 StreamARN 또는 ChannelARN CLOUD_STORAGE_MODE 입력이 이미 다른 Kinesis Video Stream 리소스에 매핑되어 있거나 제공된 StreamARN 입력 ChannelARN 또는 입력이 활성 상태가 아닌 경우 다음 중 하나를 시도해 보십시오.

1. 해당 스트림의 채널이 매핑되는 대상을 결정하는 DescribeMediaStorageConfiguration API입니다.
2. 지정된 스트림이 매핑되는 채널을 결정하는 DescribeMappedResourceConfiguration API입니다.
3. 리소스의 상태를 확인하기 위한 DescribeStream 또는 DescribeSignalingChannel API입니다.

HTTP 상태 코드: 400

ResourceNotFoundException

Amazon Kinesis Video Streams에서 지정한 스트림을 찾을 수 없습니다.

HTTP 상태 코드: 404

VersionMismatchException

지정한 스트림 버전은 최신 버전이 아닙니다. 최신 버전을 가져오려면 [DescribeStream](#) API를 사용하세요.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)

- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DescribeEdgeConfiguration

서비스: Amazon Kinesis Video Streams

StartEdgeConfigurationUpdateAPI를 사용하여 설정된 스트림의 에지 구성과 Edge Agent의 레코더 및 업로더 작업의 최신 상태를 설명합니다. 이 API를 사용하여 구성 상태를 가져와 구성이 Edge Agent와 동기화되어 있는지 확인할 수 있습니다. 이 API를 사용하여 에지 에이전트의 상태를 평가하십시오.

Note

이 API는 AWS 아프리카 (케이프타운) 지역인 af-south-1에서는 사용할 수 없습니다.

Request Syntax

```
POST /describeEdgeConfiguration HTTP/1.1
Content-type: application/json
```

```
{
  "StreamARN": "string",
  "StreamName": "string"
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

StreamARN

스트림의 Amazon 리소스 이름(ARN)입니다. 또는 를 지정하십시오. StreamName StreamARN

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+

Required: No

StreamName

에지 구성을 업데이트하려는 스트림의 이름. StreamName 또는 를 지정합니다 StreamARN.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256입니다.

패턴: [a-zA-Z0-9_.-]+

필수 여부: 아니요

응답 구문

```
HTTP/1.1 200
Content-type: application/json

{
  "CreationTime": number,
  "EdgeAgentStatus": {
    "LastRecorderStatus": {
      "JobStatusDetails": "string",
      "LastCollectedTime": number,
      "LastUpdatedTime": number,
      "RecorderStatus": "string"
    },
    "LastUploaderStatus": {
      "JobStatusDetails": "string",
      "LastCollectedTime": number,
      "LastUpdatedTime": number,
      "UploaderStatus": "string"
    }
  },
  "EdgeConfig": {
    "DeletionConfig": {
      "DeleteAfterUpload": boolean,
      "EdgeRetentionInHours": number,
      "LocalSizeConfig": {
        "MaxLocalMediaSizeInMB": number,
        "StrategyOnFullSize": "string"
      }
    }
  }
}
```

```

    },
    "HubDeviceArn": "string",
    "RecorderConfig": {
      "MediaSourceConfig": {
        "MediaUriSecretArn": "string",
        "MediaUriType": "string"
      },
      "ScheduleConfig": {
        "DurationInSeconds": number,
        "ScheduleExpression": "string"
      }
    },
    "UploaderConfig": {
      "ScheduleConfig": {
        "DurationInSeconds": number,
        "ScheduleExpression": "string"
      }
    }
  },
  "FailedStatusDetails": "string",
  "LastUpdatedTime": number,
  "StreamARN": "string",
  "StreamName": "string",
  "SyncStatus": "string"
}

```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

CreationTime

스트림의 에지 구성이 처음 생성된 시점의 타임스탬프입니다.

유형: 타임스탬프

EdgeAgentStatus

Edge Agent의 레코더 및 업로더 작업에 대한 최신 상태 세부 정보가 포함된 객체입니다. 이 정보를 사용하여 Edge Agent의 현재 상태를 확인할 수 있습니다.

유형: EdgeAgentStatus 객체

[EdgeConfig](#)

Edge Agent IoT Greengrass 구성 요소와 동기화하는 데 사용할 스트림의 에지 구성에 대한 설명입니다. Edge Agent 구성 요소는 온프레미스의 IoT Hub 장치 설정에서 실행됩니다.

유형: [EdgeConfig](#) 객체

[FailedStatusDetails](#)

생성된 장애 상태에 대한 설명.

타입: 문자열

[LastUpdatedTime](#)

스트림의 에지 구성이 마지막으로 업데이트된 시점의 타임스탬프입니다.

유형: 타임스탬프

[StreamARN](#)

스트림의 Amazon 리소스 이름(ARN)입니다.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

[StreamName](#)

에지 구성이 업데이트된 스트림의 이름.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256입니다.

패턴: `[a-zA-Z0-9_.-]+`

[SyncStatus](#)

에지 구성 업데이트의 최신 상태.

타입: 문자열

유효 값: SYNCING | ACKNOWLEDGED | IN_SYNC | SYNC_FAILED | DELETING | DELETE_FAILED | DELETING_ACKNOWLEDGED

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 [일반적인 오류](#) 섹션을 참조하세요.

AccessDeniedException

이 작업을 수행하는 데 필요한 권한이 없습니다.

HTTP 상태 코드: 401

ClientLimitExceededException

허용된 클라이언트 호출 한도를 초과했기 때문에 Kinesis Video Streams가 요청 병목 현상을 일으켰습니다. 나중에 전화를 걸어 보세요.

HTTP 상태 코드: 400

InvalidArgumentException

이 입력 매개변수의 값이 유효하지 않습니다.

HTTP 상태 코드: 400

ResourceNotFoundException

Amazon Kinesis Video Streams에서 지정한 스트림을 찾을 수 없습니다.

HTTP 상태 코드: 404

StreamEdgeConfigurationNotFoundExcpion

Amazon Kinesis Video Stream이 지정한 스트림의 엣지 구성을 찾을 수 없는 경우 예외가 렌더링됩니다.

HTTP 상태 코드: 404

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)

- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DescribeImageGenerationConfiguration

서비스: Amazon Kinesis Video Streams

지정된 Kinesis 비디오 ImageGenerationConfiguration 스트림에 대한 을 가져옵니다.

Request Syntax

```
POST /describeImageGenerationConfiguration HTTP/1.1
Content-type: application/json
```

```
{
  "StreamARN": "string",
  "StreamName": "string"
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

[StreamARN](#)

이미지 생성 구성을 검색하는 데 사용할 Kinesis 비디오 스트림의 Amazon 리소스 이름 (ARN)입니다. 또는 를 지정해야 합니다. StreamName StreamARN

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+

Required: No

[StreamName](#)

이미지 생성 구성을 검색할 스트림의 이름. StreamName또는 를 지정해야 합니다StreamARN.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256입니다.

패턴: [a-zA-Z0-9_.-]+

필수 여부: 아니요

응답 구문

```
HTTP/1.1 200
Content-type: application/json

{
  "ImageGenerationConfiguration": {
    "DestinationConfig": {
      "DestinationRegion": "string",
      "Uri": "string"
    },
    "Format": "string",
    "FormatConfig": {
      "string": "string"
    },
    "HeightPixels": number,
    "ImageSelectorType": "string",
    "SamplingInterval": number,
    "Status": "string",
    "WidthPixels": number
  }
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

[ImageGenerationConfiguration](#)

Kinesis 비디오 스트림 (KVS) 이미지 전송에 필요한 정보가 포함된 구조입니다. 이 구조가 null인 경우 구성이 스트림에서 삭제됩니다.

유형: [ImageGenerationConfiguration](#) 객체

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 [일반적인 오류](#) 섹션을 참조하세요.

AccessDeniedException

이 작업을 수행하는 데 필요한 권한은 없습니다.

HTTP 상태 코드: 401

ClientLimitExceededException

허용된 클라이언트 호출 한도를 초과했기 때문에 Kinesis Video Streams가 요청 병목 현상을 일으켰습니다. 나중에 전화를 걸어 보세요.

HTTP 상태 코드: 400

InvalidArgumentException

이 입력 매개변수의 값이 유효하지 않습니다.

HTTP 상태 코드: 400

ResourceNotFoundException

Amazon Kinesis Video Streams에서 지정한 스트림을 찾을 수 없습니다.

HTTP 상태 코드: 404

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)

- [AWS 루비 V3용 SDK](#)

DescribeMappedResourceConfiguration

서비스: Amazon Kinesis Video Streams

스트림에 대한 최신 정보를 반환합니다. 입력에 streamName or streamARN 입력해야 합니다.

Request Syntax

```
POST /describeMappedResourceConfiguration HTTP/1.1
Content-type: application/json
```

```
{
  "MaxResults": number,
  "NextToken": "string",
  "StreamARN": "string",
  "StreamName": "string"
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

MaxResults

응답으로 반환되는 최대 결과 수입니다.

유형: 정수

유효 범위: 고정값 1.

필수 여부: 아니요

NextToken

다른 일괄 결과를 얻기 위해 다음 요청 시 제공할 토큰입니다.

타입: 문자열

길이 제한: 최소 길이는 0. 최대 길이는 512입니다.

패턴: [a-zA-Z0-9+/=]*

Required: No

StreamARN

스트림의 Amazon 리소스 이름(ARN)입니다.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+

Required: No

StreamName

스트림의 이름입니다.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256입니다.

패턴: [a-zA-Z0-9_.-]+

필수 여부: 아니요

응답 구문

```
HTTP/1.1 200
Content-type: application/json

{
  "MappedResourceConfigurationList": [
    {
      "ARN": "string",
      "Type": "string"
    }
  ],
  "NextToken": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

[MappedResourceConfigurationList](#)

미디어 스토리지 구성 속성을 캡슐화하거나 포함하는 구조입니다.

유형: [MappedResourceConfigurationListItem](#) 객체 어레이

어레이 멤버: 최소 항목 수 0개. 최대 항목 수는 1개입니다.

[NextToken](#)

다음 결과 세트를 가져오기 위한 NextToken 요청에 사용된 토큰입니다.

타입: 문자열

길이 제한: 최소 길이는 0. 최대 길이는 512입니다.

패턴: [a-zA-Z0-9+/=]*

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 [일반적인 오류](#) 섹션을 참조하세요.

AccessDeniedException

이 작업을 수행하는 데 필요한 권한이 없습니다.

HTTP 상태 코드: 401

ClientLimitExceededException

허용된 클라이언트 호출 한도를 초과했기 때문에 Kinesis Video Streams가 요청 병목 현상을 일으켰습니다. 나중에 전화를 걸어 보세요.

HTTP 상태 코드: 400

InvalidArgumentException

이 입력 매개변수의 값이 유효하지 않습니다.

HTTP 상태 코드: 400

ResourceNotFoundException

Amazon Kinesis Video Streams에서 지정한 스트림을 찾을 수 없습니다.

HTTP 상태 코드: 404

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DescribeMediaStorageConfiguration

서비스: Amazon Kinesis Video Streams

채널에 대한 최신 정보를 반환합니다. ChannelARN 입력에 ChannelName or 를 지정합니다.

Request Syntax

```
POST /describeMediaStorageConfiguration HTTP/1.1
Content-type: application/json

{
  "ChannelARN": "string",
  "ChannelName": "string"
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

[ChannelARN](#)

채널의 Amazon 리소스 이름 (ARN).

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+

Required: No

[ChannelName](#)

채널의 이름입니다.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256입니다.

패턴: [a-zA-Z0-9_.-]+

필수 여부: 아니요

응답 구문

```
HTTP/1.1 200
Content-type: application/json

{
  "MediaStorageConfiguration": {
    "Status": "string",
    "StreamARN": "string"
  }
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

MediaStorageConfiguration

미디어 스토리지 구성 속성을 캡슐화하거나 포함하는 구조입니다.

유형: MediaStorageConfiguration 객체

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 [일반적인 오류](#) 섹션을 참조하세요.

AccessDeniedException

이 작업을 수행하는 데 필요한 권한은 없습니다.

HTTP 상태 코드: 401

ClientLimitExceededException

허용된 클라이언트 호출 한도를 초과했기 때문에 Kinesis Video Streams가 요청 병목 현상을 일으켰습니다. 나중에 전화를 걸어 보세요.

HTTP 상태 코드: 400

InvalidArgumentException

이 입력 매개변수의 값이 유효하지 않습니다.

HTTP 상태 코드: 400

ResourceNotFoundException

Amazon Kinesis Video Streams에서 지정한 스트림을 찾을 수 없습니다.

HTTP 상태 코드: 404

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS 파이썬용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DescribeNotificationConfiguration

서비스: Amazon Kinesis Video Streams

지정된 Kinesis 비디오 NotificationConfiguration 스트림에 대한 을 가져옵니다.

Request Syntax

```
POST /describeNotificationConfiguration HTTP/1.1
Content-type: application/json
```

```
{
  "StreamARN": "string",
  "StreamName": "string"
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

[StreamARN](#)

알림 구성을 검색하려는 Kinesis 비디오 스트림의 Amazon 리소스 이름 (ARN). 또는 StreamARN을 StreamName 지정해야 합니다.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+

Required: No

[StreamName](#)

알림 구성을 검색할 스트림의 이름. StreamName 또는 를 지정해야 합니다 StreamARN.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256입니다.

패턴: [a-zA-Z0-9_.-]+

필수 여부: 아니요

응답 구문

```
HTTP/1.1 200
Content-type: application/json

{
  "NotificationConfiguration": {
    "DestinationConfig": {
      "Uri": "string"
    },
    "Status": "string"
  }
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

NotificationConfiguration

알림에 필요한 정보가 포함된 구조입니다. 구조가 null인 경우 구성이 스트림에서 삭제됩니다.

유형: NotificationConfiguration 객체

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 [일반적인 오류](#) 섹션을 참조하세요.

AccessDeniedException

이 작업을 수행하는 데 필요한 권한이 없습니다.

HTTP 상태 코드: 401

ClientLimitExceededException

허용된 클라이언트 호출 한도를 초과했기 때문에 Kinesis Video Streams가 요청 병목 현상을 일으켰습니다. 나중에 전화를 걸어 보세요.

HTTP 상태 코드: 400

InvalidArgumentException

이 입력 매개변수의 값이 유효하지 않습니다.

HTTP 상태 코드: 400

ResourceNotFoundException

Amazon Kinesis Video Streams에서 지정한 스트림을 찾을 수 없습니다.

HTTP 상태 코드: 404

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DescribeSignalingChannel

서비스: Amazon Kinesis Video Streams

신호 채널에 대한 최신 정보를 반환합니다. 설명하려는 채널의 이름 또는 Amazon 리소스 이름 (ARN) 을 지정해야 합니다.

Request Syntax

```
POST /describeSignalingChannel HTTP/1.1
Content-type: application/json
```

```
{
  "ChannelARN": "string",
  "ChannelName": "string"
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

[ChannelARN](#)

설명하려는 신호 채널의 ARN입니다.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

[ChannelName](#)

설명하려는 신호 채널의 이름.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256입니다.

패턴: [a-zA-Z0-9_.-]+

필수 여부: 아니요

응답 구문

```
HTTP/1.1 200
Content-type: application/json

{
  "ChannelInfo": {
    "ChannelARN": "string",
    "ChannelName": "string",
    "ChannelStatus": "string",
    "ChannelType": "string",
    "CreationTime": number,
    "SingleMasterConfiguration": {
      "MessageTtlSeconds": number
    },
    "Version": "string"
  }
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

ChannelInfo

지정된 신호 채널의 메타데이터와 속성을 캡슐화하는 구조입니다.

유형: [ChannelInfo](#) 객체

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 [일반적인 오류](#) 섹션을 참조하세요.

AccessDeniedException

이 작업을 수행하는 데 필요한 권한은 없습니다.

HTTP 상태 코드: 401

ClientLimitExceededException

허용된 클라이언트 호출 한도를 초과했기 때문에 Kinesis Video Streams가 요청 병목 현상을 일으켰습니다. 나중에 전화를 걸어 보세요.

HTTP 상태 코드: 400

InvalidArgumentException

이 입력 매개변수의 값이 유효하지 않습니다.

HTTP 상태 코드: 400

ResourceNotFoundException

Amazon Kinesis Video Streams에서 지정한 스트림을 찾을 수 없습니다.

HTTP 상태 코드: 404

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DescribeStream

서비스: Amazon Kinesis Video Streams

지정된 스트림에 대한 최신 정보를 반환합니다. StreamName 또는 를 지정해야 합니다 StreamARN.

Request Syntax

```
POST /describeStream HTTP/1.1
Content-type: application/json

{
  "StreamARN": "string",
  "StreamName": "string"
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

StreamARN

스트림의 Amazon 리소스 이름(ARN)입니다.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+

Required: No

StreamName

스트림의 이름입니다.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256입니다.

패턴: [a-zA-Z0-9_.-]+

필수 여부: 아니요

응답 구문

```
HTTP/1.1 200
Content-type: application/json

{
  "StreamInfo": {
    "CreationTime": number,
    "DataRetentionInHours": number,
    "DeviceName": "string",
    "KmsKeyId": "string",
    "MediaType": "string",
    "Status": "string",
    "StreamARN": "string",
    "StreamName": "string",
    "Version": "string"
  }
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

StreamInfo

스트림을 설명하는 객체입니다.

유형: [StreamInfo](#) 객체

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 [일반적인 오류](#) 섹션을 참조하세요.

ClientLimitExceededException

허용된 클라이언트 호출 한도를 초과했기 때문에 Kinesis Video Streams가 요청 병목 현상을 일으켰습니다. 나중에 전화를 걸어 보세요.

HTTP 상태 코드: 400

InvalidArgumentException

이 입력 매개변수의 값이 유효하지 않습니다.

HTTP 상태 코드: 400

NotAuthorizedException

호출자는 이 작업을 수행할 권한이 없습니다.

HTTP 상태 코드: 401

ResourceNotFoundException

Amazon Kinesis Video Streams에서 지정한 스트림을 찾을 수 없습니다.

HTTP 상태 코드: 404

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

GetDataEndpoint

서비스: Amazon Kinesis Video Streams

읽기 또는 쓰기를 위해 지정된 스트림의 끝점을 가져옵니다. 애플리케이션에서 이 엔드포인트를 사용하여 지정된 스트림에서 읽기 (GetMedia 또는 GetMediaForFragmentList 작업 사용) 하거나 쓰기 (PutMedia 작업 사용) 할 수 있습니다.

Note

반환된 엔드포인트에는 API 이름이 추가되지 않습니다. 클라이언트는 반환된 엔드포인트에 API 이름을 추가해야 합니다.

요청에서 StreamName 또는 로 스트림을 지정합니다 StreamARN.

Request Syntax

```
POST /getDataEndpoint HTTP/1.1
Content-type: application/json

{
  "APIName": "string",
  "StreamARN": "string",
  "StreamName": "string"
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

APIName

엔드포인트를 가져올 API 작업의 이름.

타입: 문자열

유효 값: PUT_MEDIA | GET_MEDIA | LIST_FRAGMENTS |
GET_MEDIA_FOR_FRAGMENT_LIST | GET_HLS_STREAMING_SESSION_URL |
GET_DASH_STREAMING_SESSION_URL | GET_CLIP | GET_IMAGES

필수 사항 여부: 예

StreamARN

엔드포인트를 가져오려는 스트림의 Amazon 리소스 이름 (ARN) StreamName요청에서 이 파라미터 또는 a를 지정해야 합니다.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+

Required: No

StreamName

엔드포인트를 가져오려는 스트림의 이름. StreamARN요청에서 이 파라미터 또는 a를 지정해야 합니다.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256입니다.

패턴: [a-zA-Z0-9_.-]+

필수 여부: 아니요

응답 구문

```
HTTP/1.1 200
Content-type: application/json

{
  "DataEndpoint": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

[DataEndpoint](#)

엔드포인트 값. 스트림에서 데이터를 읽거나 스트림에 데이터를 쓰려면 애플리케이션에서 이 엔드포인트를 지정하십시오.

타입: 문자열

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 [일반적인 오류](#) 섹션을 참조하세요.

ClientLimitExceededException

허용된 클라이언트 호출 한도를 초과했기 때문에 Kinesis Video Streams가 요청 병목 현상을 일으켰습니다. 나중에 전화를 걸어 보세요.

HTTP 상태 코드: 400

InvalidArgumentException

이 입력 매개변수의 값이 유효하지 않습니다.

HTTP 상태 코드: 400

NotAuthorizedException

호출자는 이 작업을 수행할 권한이 없습니다.

HTTP 상태 코드: 401

ResourceNotFoundException

Amazon Kinesis Video Streams에서 지정한 스트림을 찾을 수 없습니다.

HTTP 상태 코드: 404

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

GetSignalingChannelEndpoint

서비스: Amazon Kinesis Video Streams

지정된 신호 채널이 메시지를 보내고 받을 수 있는 엔드포인트를 제공합니다. 이 API는 `Protocols` 및 `Role` 속성으로 구성된 `SingleMasterChannelEndpointConfiguration` 입력 매개변수를 사용합니다.

`Protocols` 통신 메커니즘을 결정하는 데 사용됩니다. 예를 들어 WSS 프로토콜로 지정하는 경우 이 API는 보안 웹소켓 엔드포인트를 생성합니다. HTTPS 프로토콜로 지정하는 경우 이 API는 HTTPS 엔드포인트를 생성합니다. WEBRTC 프로토콜로 지정했지만 신호 채널이 인제스트를 위해 구성되지 않은 경우 오류가 발생합니다. `InvalidArgumentException`

`Role` 메시징 권한을 결정합니다. MASTER 역할을 통해 이 API는 클라이언트가 채널의 모든 시청자와 통신하는 데 사용할 수 있는 엔드포인트를 생성합니다. VIEWER 역할을 수행하면 이 API는 클라이언트가 통신에만 사용할 수 있는 엔드포인트를 생성합니다. MASTER.

Request Syntax

```
POST /getSignalingChannelEndpoint HTTP/1.1
Content-type: application/json

{
  "ChannelARN": "string",
  "SingleMasterChannelEndpointConfiguration": {
    "Protocols": [ "string" ],
    "Role": "string"
  }
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

ChannelARN

엔드포인트를 얻으려는 신호 채널의 Amazon 리소스 이름 (ARN).

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

필수 사항 여부: Yes

[SingleMasterChannelEndpointConfiguration](#)

SINGLE_MASTER 채널 유형에 대한 엔드포인트 구성을 포함하는 구조입니다.

유형: [SingleMasterChannelEndpointConfiguration](#) 객체

필수 항목 여부: 아니요

응답 구문

```
HTTP/1.1 200
Content-type: application/json

{
  "ResourceEndpointList": [
    {
      "Protocol": "string",
      "ResourceEndpoint": "string"
    }
  ]
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

[ResourceEndpointList](#)

지정된 신호 채널의 엔드포인트 목록.

타입: [ResourceEndpointListItem](#) 객체 배열

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 [일반적인 오류](#) 섹션을 참조하세요.

AccessDeniedException

이 작업을 수행하는 데 필요한 권한이 없습니다.

HTTP 상태 코드: 401

ClientLimitExceededException

허용된 클라이언트 호출 한도를 초과했기 때문에 Kinesis Video Streams가 요청 병목 현상을 일으켰습니다. 나중에 전화를 걸어 보세요.

HTTP 상태 코드: 400

InvalidArgumentException

이 입력 매개변수의 값이 유효하지 않습니다.

HTTP 상태 코드: 400

ResourceInUseException

입력 StreamARN 또는 ChannelARN CLOUD_STORAGE_MODE 입력이 이미 다른 Kinesis Video Stream 리소스에 매핑되어 있거나 제공된 StreamARN 입력 ChannelARN 또는 입력이 활성 상태가 아닌 경우 다음 중 하나를 시도해 보십시오.

1. 해당 스트림의 채널이 매핑되는 대상을 결정하는 DescribeMediaStorageConfiguration API입니다.
2. 지정된 스트림이 매핑되는 채널을 결정하는 DescribeMappedResourceConfiguration API입니다.
3. 리소스의 상태를 확인하기 위한 DescribeStream 또는 DescribeSignalingChannel API입니다.

HTTP 상태 코드: 400

ResourceNotFoundException

Amazon Kinesis Video Streams에서 지정한 스트림을 찾을 수 없습니다.

HTTP 상태 코드: 404

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

ListEdgeAgentConfigurations

서비스: Amazon Kinesis Video Streams

지정된 Edge Agent와 관련된 에지 구성의 배열을 반환합니다.

요청 시 에지 에이전트를 지정해야 HubDeviceArn 합니다.

Note

이 API는 AWS 아프리카 (케이프타운) 지역인 af-south-1에서는 사용할 수 없습니다.

Request Syntax

```
POST /listEdgeAgentConfigurations HTTP/1.1
```

```
Content-type: application/json
```

```
{
  "HubDeviceArn": "string",
  "MaxResults": number,
  "NextToken": "string"
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

HubDeviceArn

엣지 에이전트의 “사물 인터넷 (IoT) 사물” Arn.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: arn:[a-z\d-]+:iot:[a-z0-9-]+:[0-9]+:thing/[a-zA-Z0-9_.-]+

필수 사항 여부: Yes

MaxResults

응답으로 반환되는 최대 에지 구성 수입니다. 기본값은 5입니다.

타입: 정수

유효한 범위: 최소값 1. 최대값은 10입니다.

필수 여부: 아니요

NextToken

이 매개 변수를 지정하면 ListEdgeAgentConfigurations 작업 결과가 잘리면 호출은 NextToken 응답에서 를 반환합니다. 다른 엣지 구성을 일괄 처리하려면 다음 요청 시 이 토큰을 제공하십시오.

타입: 문자열

길이 제한: 최소 길이는 0. 최대 길이는 512입니다.

패턴: [a-zA-Z0-9+/=]*

필수 여부: 아니요

응답 구문

```

HTTP/1.1 200
Content-type: application/json

{
  "EdgeConfigs": [
    {
      "CreationTime": number,
      "EdgeConfig": {
        "DeletionConfig": {
          "DeleteAfterUpload": boolean,
          "EdgeRetentionInHours": number,
          "LocalSizeConfig": {
            "MaxLocalMediaSizeInMB": number,
            "StrategyOnFullSize": "string"
          }
        },
        "HubDeviceArn": "string",
        "RecorderConfig": {

```

```

    "MediaSourceConfig": {
      "MediaUriSecretArn": "string",
      "MediaUriType": "string"
    },
    "ScheduleConfig": {
      "DurationInSeconds": number,
      "ScheduleExpression": "string"
    }
  },
  "UploaderConfig": {
    "ScheduleConfig": {
      "DurationInSeconds": number,
      "ScheduleExpression": "string"
    }
  },
  "FailedStatusDetails": "string",
  "LastUpdatedTime": number,
  "StreamARN": "string",
  "StreamName": "string",
  "SyncStatus": "string"
}
],
"NextToken": "string"
}

```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

[EdgeConfigs](#)

단일 스트림의 엣지 구성에 대한 설명.

유형: [ListEdgeAgentConfigurationsEdgeConfig](#) 객체 어레이

[NextToken](#)

응답이 잘린 경우 호출 시 지정된 토큰과 함께 이 요소가 반환됩니다. 다음 엣지 구성 배치를 가져오려면 다음 요청 시 이 토큰을 사용하세요.

타입: 문자열

길이 제한: 최소 길이는 0. 최대 길이는 512입니다.

패턴: [a-zA-Z0-9+/=]*

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 [일반적인 오류](#) 섹션을 참조하세요.

ClientLimitExceededException

허용된 클라이언트 호출 한도를 초과했기 때문에 Kinesis Video Streams가 요청 병목 현상을 일으켰습니다. 나중에 전화를 걸어 보세요.

HTTP 상태 코드: 400

InvalidArgumentException

이 입력 매개변수의 값이 유효하지 않습니다.

HTTP 상태 코드: 400

NotAuthorizedException

호출자는 이 작업을 수행할 권한이 없습니다.

HTTP 상태 코드: 401

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

ListSignalingChannels

서비스: Amazon Kinesis Video Streams

ChannelInfo객체 배열을 반환합니다. 각 객체는 신호 채널을 설명합니다. 특정 조건을 충족하는 채널만 검색하려면 a를 지정하면 됩니다. ChannelNameCondition

Request Syntax

```
POST /listSignalingChannels HTTP/1.1
Content-type: application/json

{
  "ChannelNameCondition": {
    "ComparisonOperator": "string",
    "ComparisonValue": "string"
  },
  "MaxResults": number,
  "NextToken": "string"
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

ChannelNameCondition

선택 사항: 특정 조건을 만족하는 채널만 반환합니다.

유형: ChannelNameCondition객체

필수 항목 여부: 아니요

MaxResults

응답에서 반환할 최대 채널 수입니다. 기본값은 500입니다.

타입: 정수

유효한 범위: 최소값 1. 최대값은 10,000입니다.

필수 여부: 아니요

NextToken

이 매개 변수를 지정하는 경우 ListSignalingChannels 작업 결과가 잘리면 호출은 NextToken 응답에서 를 반환합니다. 다른 채널 배치를 가져오려면 다음 요청 시 이 토큰을 제공하십시오.

타입: 문자열

길이 제한: 최소 길이는 0. 최대 길이는 512입니다.

패턴: [a-zA-Z0-9+/=]*

필수 여부: 아니요

응답 구문

```
HTTP/1.1 200
Content-type: application/json

{
  "ChannelInfoList": [
    {
      "ChannelARN": "string",
      "ChannelName": "string",
      "ChannelStatus": "string",
      "ChannelType": "string",
      "CreationTime": number,
      "SingleMasterConfiguration": {
        "MessageTtlSeconds": number
      },
      "Version": "string"
    }
  ],
  "NextToken": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

ChannelInfoList

ChannelInfo 객체 어레이.

유형: [ChannelInfo](#) 객체 어레이

NextToken

응답이 잘린 경우 호출 시 토큰과 함께 이 요소가 반환됩니다. 다음 스트림 배치를 가져오려면 다음 요청에서 이 토큰을 사용하세요.

타입: 문자열

길이 제한: 최소 길이는 0. 최대 길이는 512입니다.

패턴: [a-zA-Z0-9+/=]*

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 [일반적인 오류](#) 섹션을 참조하세요.

AccessDeniedException

이 작업을 수행하는 데 필요한 권한이 없습니다.

HTTP 상태 코드: 401

ClientLimitExceededException

허용된 클라이언트 호출 한도를 초과했기 때문에 Kinesis Video Streams가 요청 병목 현상을 일으켰습니다. 나중에 전화를 걸어 보세요.

HTTP 상태 코드: 400

InvalidArgumentException

이 입력 매개변수의 값이 유효하지 않습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

ListStreams

서비스: Amazon Kinesis Video Streams

StreamInfo객체 배열을 반환합니다. 각 객체는 스트림을 설명합니다. 특정 조건을 충족하는 스트림만 검색하려면 a를 지정하면 StreamNameCondition 됩니다.

Request Syntax

```
POST /listStreams HTTP/1.1
Content-type: application/json

{
  "MaxResults": number,
  "NextToken": "string",
  "StreamNameCondition": {
    "ComparisonOperator": "string",
    "ComparisonValue": "string"
  }
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

MaxResults

응답에서 반환할 최대 스트림 수입니다. 기본값은 10,000입니다.

타입: 정수

유효한 범위: 최소값 1. 최대값은 10,000입니다.

필수 여부: 아니요

NextToken

이 매개 변수를 지정하는 경우 ListStreams 작업 결과가 잘리면 호출은 NextToken 응답에서 반환합니다. 다른 스트림 배치를 가져오려면 다음 요청 시 이 토큰을 제공하십시오.

타입: 문자열

길이 제한: 최소 길이는 0. 최대 길이는 512입니다.

패턴: [a-zA-Z0-9+/=]*

Required: No

StreamNameCondition

선택 사항: 특정 조건을 충족하는 스트림만 반환합니다. 현재는 스트림 이름의 접두사만 조건으로 지정할 수 있습니다.

유형: [StreamNameCondition](#) 객체

필수 항목 여부: 아니요

응답 구문

```
HTTP/1.1 200
Content-type: application/json

{
  "NextToken": "string",
  "StreamInfoList": [
    {
      "CreationTime": number,
      "DataRetentionInHours": number,
      "DeviceName": "string",
      "KmsKeyId": "string",
      "MediaType": "string",
      "Status": "string",
      "StreamARN": "string",
      "StreamName": "string",
      "Version": "string"
    }
  ]
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

NextToken

응답이 잘린 경우 호출 시 토큰과 함께 이 요소가 반환됩니다. 다음 스트림 배치를 가져오려면 다음 요청에서 이 토큰을 사용하세요.

타입: 문자열

길이 제한: 최소 길이는 0. 최대 길이는 512입니다.

패턴: [a-zA-Z0-9+/=]*

StreamInfoList

StreamInfo 객체 배열.

타입: [StreamInfo](#) 객체 배열

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 [일반적인 오류](#) 섹션을 참조하세요.

ClientLimitExceededException

허용된 클라이언트 호출 한도를 초과했기 때문에 Kinesis Video Streams가 요청 병목 현상을 일으켰습니다. 나중에 전화를 걸어 보세요.

HTTP 상태 코드: 400

InvalidArgumentException

이 입력 매개변수의 값이 유효하지 않습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)

- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

ListTagsForResource

서비스: Amazon Kinesis Video Streams

지정된 신호 채널과 관련된 태그 목록을 반환합니다.

Request Syntax

```
POST /ListTagsForResource HTTP/1.1
Content-type: application/json

{
  "NextToken": "string",
  "ResourceARN": "string"
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

[NextToken](#)

이 매개 변수를 지정했는데 ListTagsForResource 호출 결과가 잘린 경우 응답에는 다음 요청 시 다음 태그 배치를 가져오는 데 사용할 수 있는 토큰이 포함됩니다.

타입: 문자열

길이 제한: 최소 길이는 0. 최대 길이는 512입니다.

패턴: [a-zA-Z0-9+/=]*

Required: No

[ResourceARN](#)

태그를 나열하려는 신호 채널의 Amazon 리소스 이름 (ARN).

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

필수 항목 여부: 예

응답 구문

```
HTTP/1.1 200
Content-type: application/json

{
  "NextToken": "string",
  "Tags": {
    "string" : "string"
  }
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

NextToken

이 파라미터를 지정했는데 `ListTagsForResource` 호출 결과가 잘린 경우 응답에는 다음 요청 시 다음 태그 세트를 가져오는 데 사용할 수 있는 토큰이 포함됩니다.

타입: 문자열

길이 제한: 최소 길이는 0. 최대 길이는 512입니다.

패턴: `[a-zA-Z0-9+/=]*`

Tags

지정된 신호 채널과 관련된 태그 키 및 값의 맵입니다.

유형: 문자열-문자열 맵

맵 항목: 최대 항목 수는 50입니다.

키 길이 제약 조건: 최소 길이는 1. 최대 길이 128.

키 패턴: `^[a-zA-Z0-9_./+=\-\@]*$`

값 길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 256.

가치 패턴: `[a-zA-Z0-9_./+=\-\@]*`

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 [일반적인 오류](#) 섹션을 참조하세요.

AccessDeniedException

이 작업을 수행하는 데 필요한 권한은 없습니다.

HTTP 상태 코드: 401

ClientLimitExceededException

허용된 클라이언트 호출 한도를 초과했기 때문에 Kinesis Video Streams가 요청 병목 현상을 일으켰습니다. 나중에 전화를 걸어 보세요.

HTTP 상태 코드: 400

InvalidArgumentException

이 입력 매개변수의 값이 유효하지 않습니다.

HTTP 상태 코드: 400

ResourceNotFoundException

Amazon Kinesis Video Streams에서 지정한 스트림을 찾을 수 없습니다.

HTTP 상태 코드: 404

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)

- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

ListTagsForStream

서비스: Amazon Kinesis Video Streams

지정된 스트림과 관련된 태그 목록을 반환합니다.

요청에서 StreamName 또는 를 지정해야 합니다StreamARN.

Request Syntax

```
POST /listTagsForStream HTTP/1.1
Content-type: application/json
```

```
{
  "NextToken": "string",
  "StreamARN": "string",
  "StreamName": "string"
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

[NextToken](#)

이 매개 변수를 지정했는데 ListTagsForStream 호출 결과가 잘린 경우 응답에는 다음 요청에서 다음 태그 배치를 가져오는 데 사용할 수 있는 토큰이 포함됩니다.

타입: 문자열

길이 제한: 최소 길이는 0. 최대 길이는 512입니다.

패턴: [a-zA-Z0-9+/=]*

Required: No

[StreamARN](#)

태그를 나열하려는 스트림의 Amazon 리소스 이름 (ARN)

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

StreamName

태그를 나열하려는 스트림의 이름.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256입니다.

패턴: `[a-zA-Z0-9_.-]+`

필수 여부: 아니요

응답 구문

```
HTTP/1.1 200
Content-type: application/json

{
  "NextToken": "string",
  "Tags": {
    "string" : "string"
  }
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

NextToken

이 매개 변수를 지정했는데 ListTags 호출 결과가 잘린 경우 응답에는 다음 요청 시 다음 태그 세트를 가져오는 데 사용할 수 있는 토큰이 포함됩니다.

타입: 문자열

길이 제한: 최소 길이는 0. 최대 길이는 512입니다.

패턴: [a-zA-Z0-9+/=]*

Tags

지정된 스트림과 관련된 태그 키 및 값의 맵입니다.

유형: 문자열-문자열 맵

맵 항목: 최대 항목 수는 50입니다.

키 길이 제약 조건: 최소 길이는 1. 최대 길이 128.

키 패턴: ^([\p{L}\p{Z}\p{N}_./=+\-@]*)\$

값 길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 256.

가치 패턴: [\p{L}\p{Z}\p{N}_./=+\-@]*

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 [일반적인 오류](#) 섹션을 참조하세요.

ClientLimitExceededException

허용된 클라이언트 호출 한도를 초과했기 때문에 Kinesis Video Streams가 요청 병목 현상을 일으켰습니다. 나중에 전화를 걸어 보세요.

HTTP 상태 코드: 400

InvalidArgumentException

이 입력 매개변수의 값이 유효하지 않습니다.

HTTP 상태 코드: 400

InvalidResourceFormatException

의 StreamARN 형식이 잘못되었습니다.

HTTP 상태 코드: 400

NotAuthorizedException

발신자는 이 작업을 수행할 권한이 없습니다.

HTTP 상태 코드: 401

ResourceNotFoundException

Amazon Kinesis Video Streams에서 지정한 스트림을 찾을 수 없습니다.

HTTP 상태 코드: 404

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

StartEdgeConfigurationUpdate

서비스: Amazon Kinesis Video Streams

스트림의 기존 에지 구성을 업데이트하는 비동기 API입니다. Kinesis Video Stream은 스트림의 에지 구성을 사내에 설정된 IoT Hub 디바이스에서 실행되는 Edge Agent IoT Greengrass 구성 요소와 동기화합니다. 동기화 시간은 허브 디바이스의 연결 상태에 따라 달라질 수 있습니다. 에지 구성이 승인되고 Edge Agent와 SyncStatus 동기화되면 가 업데이트됩니다.

이 API를 처음 호출하면 스트림에 대한 새 에지 구성이 생성되고 동기화 상태가 로 설정됩니다. SYNCING 이 API를 다시 사용하려면 동기화 상태가 다음과 같은 터미널 상태 (예: IN_SYNC 또는 SYNC_FAILED) 에 도달할 때까지 기다려야 합니다. 동기화 프로세스 중에 이 API를 ResourceInUseException 호출하면 a가 발생합니다. 스트림의 에지 구성과 Edge Agent의 연결은 15분 동안 재시도됩니다. 15분 후 상태가 SYNC_FAILED 상태로 전환됩니다.

한 장치에서 다른 장치로 에지 구성을 이동하려면 [DeleteEdgeConfiguration](#) 를 사용하여 현재 에지 구성을 삭제합니다. 그런 다음 업데이트된 허브 디바이스 StartEdgeConfigurationUpdate ARN으로 호출할 수 있습니다.

Note

이 API는 AWS 아프리카 (케이프타운) 지역인 af-south-1에서는 사용할 수 없습니다.

Request Syntax

```
POST /startEdgeConfigurationUpdate HTTP/1.1
Content-type: application/json
```

```
{
  "EdgeConfig": {
    "DeletionConfig": {
      "DeleteAfterUpload": boolean,
      "EdgeRetentionInHours": number,
      "LocalSizeConfig": {
        "MaxLocalMediaSizeInMB": number,
        "StrategyOnFullSize": "string"
      }
    },
    "HubDeviceArn": "string",
    "RecorderConfig": {
      "MediaSourceConfig": {
```

```

    "MediaUriSecretArn": "string",
    "MediaUriType": "string"
  },
  "ScheduleConfig": {
    "DurationInSeconds": number,
    "ScheduleExpression": "string"
  }
},
"UploaderConfig": {
  "ScheduleConfig": {
    "DurationInSeconds": number,
    "ScheduleExpression": "string"
  }
}
},
"StreamARN": "string",
"StreamName": "string"
}

```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

EdgeConfig

업데이트 프로세스를 호출하는 데 필요한 엣지 구성 세부 정보.

유형: [EdgeConfig](#) 객체

필수 여부: 예

StreamARN

스트림의 Amazon 리소스 이름(ARN)입니다. StreamName 또는 를 지정합니다. StreamARN

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+

Required: No

StreamName

엣지 구성을 업데이트하려는 스트림의 이름. StreamName 또는 를 지정합니다 StreamARN.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256입니다.

패턴: [a-zA-Z0-9_.-]+

필수 여부: 아니요

응답 구문

```
HTTP/1.1 200
Content-type: application/json

{
  "CreationTime": number,
  "EdgeConfig": {
    "DeletionConfig": {
      "DeleteAfterUpload": boolean,
      "EdgeRetentionInHours": number,
      "LocalSizeConfig": {
        "MaxLocalMediaSizeInMB": number,
        "StrategyOnFullSize": "string"
      }
    },
    "HubDeviceArn": "string",
    "RecorderConfig": {
      "MediaSourceConfig": {
        "MediaUriSecretArn": "string",
        "MediaUriType": "string"
      },
      "ScheduleConfig": {
        "DurationInSeconds": number,
        "ScheduleExpression": "string"
      }
    },
    "UploaderConfig": {
      "ScheduleConfig": {
        "DurationInSeconds": number,
```

```

        "ScheduleExpression": "string"
    }
}
},
"FailedStatusDetails": "string",
"LastUpdatedTime": number,
"StreamARN": "string",
"StreamName": "string",
"SyncStatus": "string"
}

```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

CreationTime

스트림의 에지 구성이 처음 생성된 시점의 타임스탬프입니다.

유형: 타임스탬프

EdgeConfig

Edge Agent IoT Greengrass 구성 요소와 동기화하는 데 사용할 스트림의 에지 구성에 대한 설명입니다. Edge Agent 구성 요소는 온프레미스의 IoT Hub 장치 설정에서 실행됩니다.

유형: [EdgeConfig](#) 객체

FailedStatusDetails

생성된 장애 상태에 대한 설명.

타입: 문자열

LastUpdatedTime

스트림의 에지 구성이 마지막으로 업데이트된 시점의 타임스탬프입니다.

유형: 타임스탬프

StreamARN

스트림의 Amazon 리소스 이름(ARN)입니다.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

StreamName

에지 구성이 업데이트된 스트림의 이름.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256입니다.

패턴: `[a-zA-Z0-9_.-]+`

SyncStatus

스트림 에지 구성의 현재 동기화 상태. 이 API를 호출하면 동기화 상태가 SYNCING 상태로 설정됩니다. DescribeEdgeConfigurationAPI를 사용하여 에지 구성의 최신 상태를 가져올 수 있습니다.

타입: 문자열

유효 값: SYNCING | ACKNOWLEDGED | IN_SYNC | SYNC_FAILED | DELETING | DELETE_FAILED | DELETING_ACKNOWLEDGED

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 [일반적인 오류](#) 섹션을 참조하세요.

AccessDeniedException

이 작업을 수행하는 데 필요한 권한은 없습니다.

HTTP 상태 코드: 401

ClientLimitExceededException

허용된 클라이언트 호출 한도를 초과했기 때문에 Kinesis Video Streams가 요청 병목 현상을 일으켰습니다. 나중에 전화를 걸어 보세요.

HTTP 상태 코드: 400

InvalidArgumentException

이 입력 매개변수의 값이 유효하지 않습니다.

HTTP 상태 코드: 400

NoDataRetentionException

스트림 데이터 보존 시간 단위는 0입니다.

HTTP 상태 코드: 400

ResourceInUseException

입력 StreamARN 또는 ChannelARN CLOUD_STORAGE_MODE 입력이 이미 다른 Kinesis Video Stream 리소스에 매핑되어 있거나 제공된 StreamARN 입력 ChannelARN 또는 입력이 활성 상태가 아닌 경우 다음 중 하나를 시도해 보십시오.

1. 해당 스트림의 채널이 매핑되는 대상을 결정하는 DescribeMediaStorageConfiguration API입니다.
2. 지정된 스트림이 매핑되는 채널을 결정하는 DescribeMappedResourceConfiguration API입니다.
3. 리소스 상태를 확인하기 위한 DescribeStream 또는 DescribeSignalingChannel API입니다.

HTTP 상태 코드: 400

ResourceNotFoundException

Amazon Kinesis Video Streams에서 지정한 스트림을 찾을 수 없습니다.

HTTP 상태 코드: 404

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)

- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

TagResource

서비스: Amazon Kinesis Video Streams

신호 채널에 하나 이상의 태그를 추가합니다. 태그는 사용자가 정의하여 리소스에 할당할 수 있는 키-값 쌍 (값은 선택 사항) 입니다. AWS 이미 있는 태그를 지정하는 경우, 태그 값은 요청에서 지정한 값으로 대체됩니다. 자세한 내용은 [AWS Billing and Cost Management](#) 및 [비용 관리 사용 설명서의 비용 할당 태그 사용](#)을 참조하십시오.

Request Syntax

```
POST /TagResource HTTP/1.1
Content-type: application/json

{
  "ResourceARN": "string",
  "Tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ]
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

[ResourceARN](#)

태그를 추가하려는 신호 채널의 Amazon 리소스 이름 (ARN).

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

필수 사항 여부: Yes

Tags

지정된 신호 채널과 연결할 태그 목록입니다. 각 태그는 키-값 페어입니다.

유형: [Tag](#) 객체 어레이

배열 멤버: 최소 항목 수는 1개입니다. 최대 항목 수는 50개입니다.

필수 여부: 예

응답 구문

```
HTTP/1.1 200
```

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 반환합니다.

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 [일반적인 오류](#) 섹션을 참조하세요.

AccessDeniedException

이 작업을 수행하는 데 필요한 권한은 없습니다.

HTTP 상태 코드: 401

ClientLimitExceededException

허용된 클라이언트 호출 한도를 초과했기 때문에 Kinesis Video Streams가 요청 병목 현상을 일으켰습니다. 나중에 전화를 걸어 보세요.

HTTP 상태 코드: 400

InvalidArgumentException

이 입력 매개변수의 값이 유효하지 않습니다.

HTTP 상태 코드: 400

ResourceNotFoundException

Amazon Kinesis Video Streams에서 지정한 스트림을 찾을 수 없습니다.

HTTP 상태 코드: 404

TagsPerResourceExceededLimitException

리소스에 연결할 수 있는 태그의 제한을 초과했습니다. Kinesis 비디오 스트림은 최대 50개의 태그를 지원할 수 있습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS 파이썬용 SDK](#)
- [AWS 루비 V3용 SDK](#)

TagStream

서비스: Amazon Kinesis Video Streams

스트림에 하나 이상의 태그를 추가합니다. 태그는 정의하여 리소스에 할당할 AWS 수 있는 키-값 쌍 (값은 선택 사항) 입니다. 이미 있는 태그를 지정하는 경우, 태그 값은 요청에서 지정한 값으로 대체됩니다. 자세한 내용은 AWS Billing and Cost Management 및 비용 관리 사용 설명서의 [비용 할당 태그 사용](#)을 참조하십시오.

StreamName 또는 를 제공해야 합니다 StreamARN.

이 작업에는 KinesisVideo:TagStream 작업에 대한 권한이 필요합니다.

Kinesis 비디오 스트림은 최대 50개의 태그를 지원할 수 있습니다.

Request Syntax

```
POST /tagStream HTTP/1.1
Content-type: application/json

{
  "StreamARN": "string",
  "StreamName": "string",
  "Tags": {
    "string" : "string"
  }
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

[StreamARN](#)

태그 또는 태그를 추가하려는 리소스의 Amazon 리소스 이름 (ARN).

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

StreamName

태그 또는 태그를 추가하려는 스트림의 이름.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256입니다.

패턴: `[a-zA-Z0-9_.-]+`

Required: No

Tags

지정된 스트림과 연결할 태그 목록입니다. 각 태그는 키-값 쌍입니다 (값은 선택 사항).

유형: 문자열-문자열 맵

맵 항목: 최대 항목 수는 50입니다.

키 길이 제약 조건: 최소 길이는 1. 최대 길이 128.

키 패턴: `^([\p{L}\p{Z}\p{N}_.: / = + \ - @] *) $`

값 길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 256.

가치 패턴: `[\p{L}\p{Z}\p{N}_.: / = + \ - @] *`

필수 여부: 예

응답 구문

```
HTTP/1.1 200
```

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 반환합니다.

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 [일반적인 오류](#) 섹션을 참조하세요.

ClientLimitExceededException

허용된 클라이언트 호출 한도를 초과했기 때문에 Kinesis Video Streams가 요청 병목 현상을 일으켰습니다. 나중에 전화를 걸어 보세요.

HTTP 상태 코드: 400

InvalidArgumentException

이 입력 매개변수의 값이 유효하지 않습니다.

HTTP 상태 코드: 400

InvalidResourceFormatException

의 StreamARN 형식이 잘못되었습니다.

HTTP 상태 코드: 400

NotAuthorizedException

발신자는 이 작업을 수행할 권한이 없습니다.

HTTP 상태 코드: 401

ResourceNotFoundException

Amazon Kinesis Video Streams에서 지정한 스트림을 찾을 수 없습니다.

HTTP 상태 코드: 404

TagsPerResourceExceededLimitException

리소스에 연결할 수 있는 태그의 한도를 초과했습니다. Kinesis 비디오 스트림은 최대 50개의 태그를 지원할 수 있습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

UntagResource

서비스: Amazon Kinesis Video Streams

신호 채널에서 하나 이상의 태그를 제거합니다. 요청 시 태그 키만 지정하고 값은 지정하지 마십시오. 존재하지 않는 태그 키를 지정하면 무시됩니다.

Request Syntax

```
POST /UntagResource HTTP/1.1
Content-type: application/json

{
  "ResourceARN": "string",
  "TagKeyList": [ "string" ]
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

ResourceARN

태그를 제거하려는 신호 채널의 Amazon 리소스 이름 (ARN).

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+

필수 사항 여부: Yes

TagKeyList

제거하려는 태그의 키 목록.

유형: 문자열 어레이

배열 멤버: 최소 항목 수는 1개입니다. 최대수 50개.

길이 제약: 최소 길이 1. 최대 길이 128.

패턴: $^([\p{L}\p{Z}\p{N}_\cdot:/=+\-@]^*)\$$

필수 항목 여부: 예

응답 구문

```
HTTP/1.1 200
```

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 반환합니다.

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 [일반적인 오류](#) 섹션을 참조하세요.

AccessDeniedException

이 작업을 수행하는 데 필요한 권한은 없습니다.

HTTP 상태 코드: 401

ClientLimitExceededException

허용된 클라이언트 호출 한도를 초과했기 때문에 Kinesis Video Streams가 요청 병목 현상을 일으켰습니다. 나중에 전화를 걸어 보세요.

HTTP 상태 코드: 400

InvalidArgumentException

이 입력 매개변수의 값이 유효하지 않습니다.

HTTP 상태 코드: 400

ResourceNotFoundException

Amazon Kinesis Video Streams에서 지정한 스트림을 찾을 수 없습니다.

HTTP 상태 코드: 404

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

UntagStream

서비스: Amazon Kinesis Video Streams

스트림에서 하나 이상의 태그를 제거합니다. 요청 시 태그 키만 지정하고 값은 지정하지 마십시오. 존재하지 않는 태그 키를 지정하면 무시됩니다.

요청 시 StreamName OR를 제공해야 합니다StreamARN.

Request Syntax

```
POST /untagStream HTTP/1.1
Content-type: application/json

{
  "StreamARN": "string",
  "StreamName": "string",
  "TagKeyList": [ "string" ]
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

StreamARN

태그를 제거하려는 스트림의 Amazon 리소스 이름 (ARN).

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+

Required: No

StreamName

태그를 제거하려는 스트림의 이름.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256입니다.

패턴: [a-zA-Z0-9_.-]+

Required: No

TagKeyList

제거하려는 태그의 키 목록.

유형: 문자열 어레이

배열 멤버: 최소 항목 수는 1개입니다. 최대수 50개.

길이 제약: 최소 길이 1. 최대 길이 128.

패턴: ^([\p{L}\p{Z}\p{N}_.:/+\\-@]*)\$

필수 항목 여부: 예

응답 구문

```
HTTP/1.1 200
```

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 반환합니다.

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 [일반적인 오류](#) 섹션을 참조하세요.

ClientLimitExceededException

허용된 클라이언트 호출 한도를 초과했기 때문에 Kinesis Video Streams가 요청 병목 현상을 일으켰습니다. 나중에 전화를 걸어 보세요.

HTTP 상태 코드: 400

InvalidArgumentException

이 입력 매개변수의 값이 유효하지 않습니다.

HTTP 상태 코드: 400

InvalidResourceFormatException

의 StreamARN 형식이 잘못되었습니다.

HTTP 상태 코드: 400

NotAuthorizedException

발신자는 이 작업을 수행할 권한이 없습니다.

HTTP 상태 코드: 401

ResourceNotFoundException

Amazon Kinesis Video Streams에서 지정한 스트림을 찾을 수 없습니다.

HTTP 상태 코드: 404

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

UpdateDataRetention

서비스: Amazon Kinesis Video Streams

스트림의 데이터 보존 기간을 지정한 값만큼 늘리거나 줄입니다. 데이터 보존 기간을 늘릴지 또는 줄일지 여부를 지정하려면 요청 본문에 Operation 파라미터를 지정하십시오. 요청에서 StreamName 또는 StreamARN을 지정해야 합니다.

이 작업에는 KinesisVideo:UpdateDataRetention 작업에 대한 권한이 필요합니다.

데이터 보존 기간을 변경하면 다음과 같이 스트림의 데이터에 영향을 줍니다.

- 데이터 보존 기간이 늘어나는 경우 기존 데이터는 새 보존 기간 동안 보존됩니다. 예를 들어 데이터 보존 기간을 1시간에서 7시간으로 늘리면 기존 데이터는 모두 7시간 동안 보존됩니다.
- 데이터 보존 기간이 줄어들면 기존 데이터는 새 보존 기간 동안 보존됩니다. 예를 들어 데이터 보존 기간이 7시간에서 1시간으로 단축되면 기존의 모든 데이터는 1시간 동안 보존되고 1시간이 지난 데이터는 즉시 삭제됩니다.

Request Syntax

```
POST /updateDataRetention HTTP/1.1
Content-type: application/json

{
  "CurrentVersion": "string",
  "DataRetentionChangeInHours": number,
  "Operation": "string",
  "StreamARN": "string",
  "StreamName": "string"
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

CurrentVersion

보존 기간을 변경하려는 스트림의 버전. 버전을 가져오려면 DescribeStream 또는 ListStreams API를 호출하십시오.

유형: 문자열

길이 제한: 최소 길이는 1. 최대 길이는 64.

패턴: [a-zA-Z0-9]+

필수 사항 여부: Yes

DataRetentionChangeInHours

현재 리텐션을 조정하는 데 걸리는 시간. 지정한 값은 에 따라 현재 값에 더해지거나 빼집니다.
operation

데이터 보존 최소값은 0이고 최대값은 87600 (10년)입니다.

타입: 정수

유효한 범위: 최소값 1.

필수 여부: 예

Operation

보존 기간을 늘릴지 아니면 줄일지를 나타냅니다.

타입: 문자열

유효 값: INCREASE_DATA_RETENTION | DECREASE_DATA_RETENTION

필수 사항 여부: 예

StreamARN

보존 기간을 변경하려는 스트림의 Amazon 리소스 이름 (ARN).

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+

Required: No

StreamName

보존 기간을 변경하려는 스트림의 이름.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256입니다.

패턴: [a-zA-Z0-9_.-]+

필수 여부: 아니요

응답 구문

```
HTTP/1.1 200
```

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 반환합니다.

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 [일반적인 오류](#) 섹션을 참조하세요.

ClientLimitExceededException

허용된 클라이언트 호출 한도를 초과했기 때문에 Kinesis Video Streams가 요청 병목 현상을 일으켰습니다. 나중에 전화를 걸어 보세요.

HTTP 상태 코드: 400

InvalidArgumentException

이 입력 매개변수의 값이 유효하지 않습니다.

HTTP 상태 코드: 400

NotAuthorizedException

호출자는 이 작업을 수행할 권한이 없습니다.

HTTP 상태 코드: 401

ResourceInUseException

입력 StreamARN 또는 ChannelARN CLOUD_STORAGE_MODE 입력이 이미 다른 Kinesis Video Stream 리소스에 매핑되어 있거나 제공된 StreamARN 입력 ChannelARN 또는 입력이 활성 상태가 아닌 경우 다음 중 하나를 시도해 보십시오.

1. 해당 스트림의 채널이 매핑되는 대상을 결정하는 DescribeMediaStorageConfiguration API입니다.
2. 지정된 스트림이 매핑되는 채널을 결정하는 DescribeMappedResourceConfiguration API입니다.
3. 리소스 상태를 확인하기 위한 DescribeStream 또는 DescribeSignalingChannel API입니다.

HTTP 상태 코드: 400

ResourceNotFoundException

Amazon Kinesis Video Streams에서 지정한 스트림을 찾을 수 없습니다.

HTTP 상태 코드: 404

VersionMismatchException

지정한 스트림 버전은 최신 버전이 아닙니다. 최신 버전을 가져오려면 [DescribeStream](#) API를 사용하세요.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)

- [AWS 루비 V3용 SDK](#)

UpdateImageGenerationConfiguration

서비스: Amazon Kinesis Video Streams

StreamInfo 및 ImageProcessingConfiguration 필드를 업데이트합니다.

Request Syntax

```
POST /updateImageGenerationConfiguration HTTP/1.1
Content-type: application/json
```

```
{
  "ImageGenerationConfiguration": {
    "DestinationConfig": {
      "DestinationRegion": "string",
      "Uri": "string"
    },
    "Format": "string",
    "FormatConfig": {
      "string": "string"
    },
    "HeightPixels": number,
    "ImageSelectorType": "string",
    "SamplingInterval": number,
    "Status": "string",
    "WidthPixels": number
  },
  "StreamARN": "string",
  "StreamName": "string"
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

ImageGenerationConfiguration

KVS 이미지 전송에 필요한 정보가 포함된 구조입니다. 구조가 null인 경우 구성이 스트림에서 삭제됩니다.

유형: [ImageGenerationConfiguration](#) 객체

필수 항목 여부: 아니요

[StreamARN](#)

이미지 생성 구성을 업데이트하려는 Kinesis 비디오 스트림의 Amazon 리소스 이름 (ARN). 또는 를 지정해야 합니다. StreamName StreamARN

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+

Required: No

[StreamName](#)

이미지 생성 구성을 업데이트할 스트림의 이름. StreamName 또는 를 지정해야 합니다 StreamARN.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256입니다.

패턴: [a-zA-Z0-9_.-]+

필수 여부: 아니요

응답 구문

```
HTTP/1.1 200
```

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 반환합니다.

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 [일반적인 오류](#) 섹션을 참조하세요.

AccessDeniedException

이 작업을 수행하는 데 필요한 권한은 없습니다.

HTTP 상태 코드: 401

ClientLimitExceededException

허용된 클라이언트 호출 한도를 초과했기 때문에 Kinesis Video Streams가 요청 병목 현상을 일으켰습니다. 나중에 전화를 걸어 보세요.

HTTP 상태 코드: 400

InvalidArgumentException

이 입력 매개변수의 값이 유효하지 않습니다.

HTTP 상태 코드: 400

NoDataRetentionException

스트림 데이터 보존 기간 (시간) 은 0입니다.

HTTP 상태 코드: 400

ResourceInUseException

입력 StreamARN 또는 ChannelARN CLOUD_STORAGE_MODE 입력이 이미 다른 Kinesis Video Stream 리소스에 매핑되어 있거나 제공된 StreamARN 입력 ChannelARN 또는 입력이 활성 상태가 아닌 경우 다음 중 하나를 시도해 보십시오.

1. 해당 스트림의 채널이 매핑되는 대상을 결정하는 DescribeMediaStorageConfiguration API입니다.
2. 지정된 스트림이 매핑되는 채널을 결정하는 DescribeMappedResourceConfiguration API입니다.
3. 리소스 상태를 확인하기 위한 DescribeStream 또는 DescribeSignalingChannel API입니다.

HTTP 상태 코드: 400

ResourceNotFoundException

Amazon Kinesis Video Streams에서 지정한 스트림을 찾을 수 없습니다.

HTTP 상태 코드: 404

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

UpdateMediaStorageConfiguration

서비스: Amazon Kinesis Video Streams

미디어를 SignalingChannel 저장하기 위해 a를 스트림에 연결합니다. 다음과 같은 두 가지 신호 모드를 지정할 수 있습니다.

- StorageStatus가 활성화되면 데이터가 StreamARN 제공된 위치에 저장됩니다. WebRTC 통합이 작동하려면 스트림에 데이터 보존이 활성화되어 있어야 합니다.
- 를 비활성화하면 StorageStatus 데이터가 저장되지 않으므로 StreamARN 파라미터가 필요하지 않습니다.

Important

를 활성화하면 StorageStatus 직접 peer-to-peer (마스터-뷰어) 연결이 더 이상 발생하지 않습니다. 피어는 스토리지 세션에 직접 연결합니다. JoinStorageSessionAPI를 호출하여 SDP 오퍼 전송을 트리거하고 피어와 스토리지 세션 간에 연결을 설정해야 합니다.

Request Syntax

```
POST /updateMediaStorageConfiguration HTTP/1.1
Content-type: application/json
```

```
{
  "ChannelARN": "string",
  "MediaStorageConfiguration": {
    "Status": "string",
    "StreamARN": "string"
  }
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

ChannelARN

채널의 Amazon 리소스 이름 (ARN).

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

필수 사항 여부: Yes

MediaStorageConfiguration

미디어 스토리지 구성 속성을 캡슐화하거나 포함하는 구조입니다.

유형: [MediaStorageConfiguration](#) 객체

필수 여부: 예

응답 구문

```
HTTP/1.1 200
```

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 반환합니다.

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 [일반적인 오류](#) 섹션을 참조하세요.

AccessDeniedException

이 작업을 수행하는 데 필요한 권한은 없습니다.

HTTP 상태 코드: 401

ClientLimitExceededException

허용된 클라이언트 호출 한도를 초과했기 때문에 Kinesis Video Streams가 요청 병목 현상을 일으켰습니다. 나중에 전화를 걸어 보세요.

HTTP 상태 코드: 400

InvalidArgumentException

이 입력 매개변수의 값이 유효하지 않습니다.

HTTP 상태 코드: 400

NoDataRetentionException

스트림 데이터 보존 시간 단위는 0입니다.

HTTP 상태 코드: 400

ResourceInUseException

입력 StreamARN 또는 ChannelARN CLOUD_STORAGE_MODE 입력이 이미 다른 Kinesis Video Stream 리소스에 매핑되어 있거나 제공된 StreamARN 입력 ChannelARN 또는 입력이 활성 상태가 아닌 경우 다음 중 하나를 시도해 보십시오.

1. 해당 스트림의 채널이 매핑되는 대상을 결정하는 DescribeMediaStorageConfiguration API입니다.
2. 지정된 스트림이 매핑되는 채널을 결정하는 DescribeMappedResourceConfiguration API입니다.
3. 리소스 상태를 확인하기 위한 DescribeStream 또는 DescribeSignalingChannel API입니다.

HTTP 상태 코드: 400

ResourceNotFoundException

Amazon Kinesis Video Streams에서 지정한 스트림을 찾을 수 없습니다.

HTTP 상태 코드: 404

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)

- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

UpdateNotificationConfiguration

서비스: Amazon Kinesis Video Streams

스트림에 대한 알림 정보를 업데이트합니다.

Request Syntax

```
POST /updateNotificationConfiguration HTTP/1.1
Content-type: application/json
```

```
{
  "NotificationConfiguration": {
    "DestinationConfig": {
      "Uri": "string"
    },
    "Status": "string"
  },
  "StreamARN": "string",
  "StreamName": "string"
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

[NotificationConfiguration](#)

알림에 필요한 정보를 포함하는 구조입니다. 구조가 null인 경우 구성이 스트림에서 삭제됩니다.

유형: [NotificationConfiguration](#) 객체

필수 항목 여부: 아니요

[StreamARN](#)

알림 구성을 업데이트하려는 Kinesis 비디오 스트림의 Amazon 리소스 이름 (ARN). 또는 를 지정해야 합니다. StreamName StreamARN

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

StreamName

알림 구성을 업데이트할 스트림의 이름. StreamName 또는 를 지정해야 합니다 StreamARN.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256입니다.

패턴: `[a-zA-Z0-9_.-]+`

필수 여부: 아니요

응답 구문

```
HTTP/1.1 200
```

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 반환합니다.

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 [일반적인 오류](#) 섹션을 참조하세요.

AccessDeniedException

이 작업을 수행하는 데 필요한 권한은 없습니다.

HTTP 상태 코드: 401

ClientLimitExceededException

허용된 클라이언트 호출 한도를 초과했기 때문에 Kinesis Video Streams가 요청 병목 현상을 일으켰습니다. 나중에 전화를 걸어 보세요.

HTTP 상태 코드: 400

InvalidArgumentException

이 입력 매개변수의 값이 유효하지 않습니다.

HTTP 상태 코드: 400

NoDataRetentionException

스트림 데이터 보존 기간 (시간) 은 0입니다.

HTTP 상태 코드: 400

ResourceInUseException

입력 StreamARN 또는 ChannelARN CLOUD_STORAGE_MODE 입력이 이미 다른 Kinesis Video Stream 리소스에 매핑되어 있거나 제공된 StreamARN 입력 ChannelARN 또는 입력이 활성 상태가 아닌 경우 다음 중 하나를 시도해 보십시오.

1. 해당 스트림의 채널이 매핑되는 대상을 결정하는 DescribeMediaStorageConfiguration API입니다.
2. 지정된 스트림이 매핑되는 채널을 결정하는 DescribeMappedResourceConfiguration API입니다.
3. 리소스 상태를 확인하기 위한 DescribeStream 또는 DescribeSignalingChannel API입니다.

HTTP 상태 코드: 400

ResourceNotFoundException

Amazon Kinesis Video Streams에서 지정한 스트림을 찾을 수 없습니다.

HTTP 상태 코드: 404

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)

- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

UpdateSignalingChannel

서비스: Amazon Kinesis Video Streams

기존 신호 채널을 업데이트합니다. 이는 비동기 작업이며 완료하는 데 시간이 걸립니다.

MessageTtlSeconds 값이 업데이트 (증가 또는 감소) 되면 업데이트 후 이 채널을 통해 전송되는 새 메시지에만 적용됩니다. 기존 메시지는 여전히 이전 MessageTtlSeconds 값에 따라 만료됩니다.

Request Syntax

```
POST /updateSignalingChannel HTTP/1.1
Content-type: application/json

{
  "ChannelARN": "string",
  "CurrentVersion": "string",
  "SingleMasterConfiguration": {
    "MessageTtlSeconds": number
  }
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

ChannelARN

업데이트하려는 신호 채널의 Amazon 리소스 이름 (ARN).

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+

필수 사항 여부: Yes

CurrentVersion

업데이트하려는 신호 채널의 현재 버전.

유형: 문자열

길이 제한: 최소 길이는 1. 최대 길이는 64.

패턴: [a-zA-Z0-9]+

필수 사항 여부: Yes

SingleMasterConfiguration

업데이트하려는 신호 채널 SINGLE_MASTER 유형에 대한 구성이 포함된 구조입니다.

유형: [SingleMasterConfiguration](#) 객체

필수 항목 여부: 아니요

응답 구문

```
HTTP/1.1 200
```

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 반환합니다.

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 [일반적인 오류](#) 섹션을 참조하세요.

AccessDeniedException

이 작업을 수행하는 데 필요한 권한은 없습니다.

HTTP 상태 코드: 401

ClientLimitExceededException

허용된 클라이언트 호출 한도를 초과했기 때문에 Kinesis Video Streams가 요청 병목 현상을 일으켰습니다. 나중에 전화를 걸어 보세요.

HTTP 상태 코드: 400

InvalidArgumentException

이 입력 매개변수의 값이 유효하지 않습니다.

HTTP 상태 코드: 400

ResourceInUseException

입력 StreamARN 또는 ChannelARN CLOUD_STORAGE_MODE 입력이 이미 다른 Kinesis Video Stream 리소스에 매핑되어 있거나 제공된 StreamARN 입력 ChannelARN 또는 입력이 활성 상태가 아닌 경우 다음 중 하나를 시도해 보십시오.

1. 해당 스트림의 채널이 매핑되는 대상을 결정하는 DescribeMediaStorageConfiguration API입니다.
2. 지정된 스트림이 매핑되는 채널을 결정하는 DescribeMappedResourceConfiguration API입니다.
3. 리소스 상태를 확인하기 위한 DescribeStream 또는 DescribeSignalingChannel API입니다.

HTTP 상태 코드: 400

ResourceNotFoundException

Amazon Kinesis Video Streams에서 지정한 스트림을 찾을 수 없습니다.

HTTP 상태 코드: 404

VersionMismatchException

지정한 스트림 버전은 최신 버전이 아닙니다. 최신 버전을 가져오려면 [DescribeStream](#) API를 사용하세요.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)

- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS 파이썬용 SDK](#)
- [AWS 루비 V3용 SDK](#)

UpdateStream

서비스: Amazon Kinesis Video Streams

디바이스 이름 및 미디어 유형과 같은 스트림 메타데이터를 업데이트합니다.

스트림의 스트림 이름 또는 Amazon 리소스 이름 (ARN) 을 제공해야 합니다.

업데이트하기 전에 스트림의 최신 버전이 있는지 확인하기 위해 스트림 버전을 지정할 수 있습니다. Kinesis Video Streams는 각 스트림에 버전을 할당합니다. 스트림을 업데이트하면 Kinesis Video Streams가 새 버전 번호를 할당합니다. 최신 스트림 버전을 가져오려면 API를 사용하십시오.

DescribeStream

UpdateStream비동기 작업이므로 완료하는 데 시간이 걸립니다.

Request Syntax

```
POST /updateStream HTTP/1.1
Content-type: application/json

{
  "CurrentVersion": "string",
  "DeviceName": "string",
  "MediaType": "string",
  "StreamARN": "string",
  "StreamName": "string"
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

CurrentVersion

메타데이터를 업데이트하려는 스트림의 버전입니다.

유형: 문자열

길이 제한: 최소 길이는 1. 최대 길이는 64.

패턴: [a-zA-Z0-9]+

필수 사항 여부: Yes

DeviceName

스트림에 쓰고 있는 디바이스의 이름.

Note

현재 구현에서는 Kinesis Video Streams가 이 이름을 사용하지 않습니다.

유형: 문자열

길이 제약: 최소 길이는 1. 최대 길이 128.

패턴: [a-zA-Z0-9_.-]+

Required: No

MediaType

스트림의 미디어 유형. 스트림에 포함된 콘텐츠 유형을 스트림 MediaType 소비자에게 지정하는 데 사용됩니다. 미디어 유형에 대한 자세한 내용은 [미디어 유형을](#) 참조하십시오. 를 지정하기로 선택한 경우 [이름 지정 요구 사항을](#) 참조하십시오. MediaType

콘솔에서 비디오를 재생하려면 올바른 비디오 유형을 지정해야 합니다. 예를 들어 스트림의 비디오가 H.264인 경우 로 지정하십시오 video/h264. MediaType

유형: 문자열

길이 제약: 최소 길이는 1. 최대 길이 128.

패턴: [\w\-\.\.]+/[\w\-\.\.]+(,[\w\-\.\.]+/[\w\-\.\.]+)*

Required: No

StreamARN

메타데이터를 업데이트하려는 스트림의 ARN입니다.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

StreamName

메타데이터를 업데이트하려는 스트림의 이름.

스트림 이름은 스트림의 식별자이며 각 계정 및 지역별로 고유해야 합니다.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256입니다.

패턴: `[a-zA-Z0-9_.-]+`

필수 여부: 아니요

응답 구문

```
HTTP/1.1 200
```

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 반환합니다.

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 [일반적인 오류](#) 섹션을 참조하세요.

ClientLimitExceededException

허용된 클라이언트 호출 한도를 초과했기 때문에 Kinesis Video Streams가 요청 병목 현상을 일으켰습니다. 나중에 전화를 걸어 보세요.

HTTP 상태 코드: 400

InvalidArgumentException

이 입력 매개변수의 값이 유효하지 않습니다.

HTTP 상태 코드: 400

NotAuthorizedException

호출자는 이 작업을 수행할 권한이 없습니다.

HTTP 상태 코드: 401

ResourceInUseException

입력 StreamARN 또는 ChannelARN CLOUD_STORAGE_MODE 입력이 이미 다른 Kinesis Video Stream 리소스에 매핑되어 있거나 제공된 StreamARN 입력 ChannelARN 또는 입력이 활성 상태가 아닌 경우 다음 중 하나를 시도해 보십시오.

1. 해당 스트림의 채널이 매핑되는 대상을 결정하는 DescribeMediaStorageConfiguration API입니다.
2. 지정된 스트림이 매핑되는 채널을 결정하는 DescribeMappedResourceConfiguration API입니다.
3. 리소스 상태를 확인하기 위한 DescribeStream 또는 DescribeSignalingChannel API입니다.

HTTP 상태 코드: 400

ResourceNotFoundException

Amazon Kinesis Video Streams에서 지정한 스트림을 찾을 수 없습니다.

HTTP 상태 코드: 404

VersionMismatchException

지정한 스트림 버전은 최신 버전이 아닙니다. 최신 버전을 가져오려면 [DescribeStream](#) API를 사용하세요.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)

- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

Amazon Kinesis Video Streams

Amazon Kinesis Video Streams Media에서는 다음 작업을 지원합니다.

- [GetMedia](#)
- [PutMedia](#)

GetMedia

서비스: Amazon Kinesis Video Streams Media

이 API를 사용하여 Kinesis 비디오 스트림에서 미디어 콘텐츠를 검색할 수 있습니다. 요청에서 스트림 이름 또는 스트림 Amazon 리소스 이름 (ARN) 과 시작 청크를 식별합니다. 그러면 Kinesis Video Streams는 청크 스트림을 프래그먼트 번호에 따라 순서대로 반환합니다.

Note

먼저 `GetDataEndpoint` API를 호출하여 엔드포인트를 가져와야 합니다. 그런 다음 `GetMedia` `--endpoint-url` 파라미터를 사용하여 이 엔드포인트로 요청을 전송합니다.

미디어 데이터 (프래그먼트) 를 스트림에 저장하면 Kinesis Video Streams는 들어오는 각 프래그먼트와 관련 메타데이터를 “청크”라고 하는 위치에 저장합니다. 자세한 내용은 을 참조하십시오. [PutMedia](#) `GetMedia`API는 요청에서 지정한 청크부터 시작하여 이러한 청크의 스트림을 반환합니다.

API 사용 시 다음과 같은 제한이 적용됩니다. `GetMedia`

- 클라이언트는 스트림당 초당 `GetMedia` 최대 5회까지 호출할 수 있습니다.
- Kinesis Video Streams는 세션 중에 초당 최대 25메가바이트 (또는 초당 200메가비트) 의 속도로 미디어 데이터를 전송합니다. `GetMedia`

Note

`GetMediaHTTP` 응답 상태 코드는 즉시 반환되지만 재생에 사용할 수 있는 인제스트된 프래그먼트가 없는 경우 3초 후에 HTTP 응답 페이로드 읽기 시간이 초과됩니다.

Note

Kinesis Video Streams 미디어 API를 호출한 후 오류가 발생하는 경우 HTTP 상태 코드 및 응답 본문 외에도 다음과 같은 정보가 포함됩니다.

- `x-amz-ErrorTypeHTTP` 헤더 - HTTP 상태 코드가 제공하는 내용 외에도 보다 구체적인 오류 유형을 포함합니다.
- `x-amz-RequestIdHTTP` 헤더 — 문제를 신고하려는 경우 지원 팀에서 요청 ID를 제공하면 문제를 더 잘 진단할 수 있습니다. AWS

HTTP 상태 코드와 `ErrorType` 헤더를 모두 사용하여 오류를 재시도할 수 있는지 여부와 조건에 대해 프로그래밍 방식으로 결정할 수 있으며, 클라이언트 프로그래머가 성공적으로 다시 시도하기 위해 취해야 할 조치에 대한 정보도 제공할 수 있습니다.

[자세한 내용은 이 항목 하단의 오류 섹션 및 일반적인 오류를 참조하십시오.](#)

Request Syntax

```
POST /getMedia HTTP/1.1
Content-type: application/json

{
  "StartSelector": {
    "AfterFragmentNumber": "string",
    "ContinuationToken": "string",
    "StartSelectorType": "string",
    "StartTimestamp": number
  },
  "StreamARN": "string",
  "StreamName": "string"
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

[StartSelector](#)

지정된 스트림에서 가져올 시작 청크를 식별합니다.

유형: [StartSelector](#) 객체

필수 여부: 예

[StreamARN](#)

미디어 콘텐츠를 가져오려는 스트림의 ARN입니다. 를 지정하지 않는 경우 를 지정해야 합니다.

streamARN streamName

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

StreamName

미디어 콘텐츠를 가져오려는 Kinesis 비디오 스트림 이름입니다. 를 지정하지 않는 경우 를 지정해야 합니다. `streamName` `streamARN`

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256입니다.

패턴: `[a-zA-Z0-9_.-]+`

필수 여부: 아니요

응답 구문

```
HTTP/1.1 200
Content-Type: ContentType
```

Payload

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

응답에 다음 HTTP 헤더가 반환됩니다.

ContentType

요청된 미디어의 콘텐츠 유형.

길이 제약: 최소 길이 1. 최대 길이 128.

패턴: `^[a-zA-Z0-9_\.\\-]+$`

응답은 다음 내용을 HTTP 본문으로 반환합니다.

Payload

Kinesis Video Streams가 반환하는 페이로드에는 지정된 스트림의 청크 시퀀스입니다. 청크에 대한 자세한 내용은 [이 링크](#)를 참조하십시오. [PutMedia](#) Kinesis Video Streams가 호출에서 GetMedia 반환하는 청크에는 다음과 같은 추가 마트로스카 (MKV) 태그도 포함됩니다.

- AWS_KINESISVIDEO_CONTINUATION_TOKEN (UTF-8 문자열) - GetMedia 통화가 종료되는 경우 다음 요청에서 이 연속 토큰을 사용하여 마지막 요청이 종료된 다음 청크를 가져올 수 있습니다.
- AWS_KINESISVIDEO_MILLIS_BEHIND_NOW (UTF-8 문자열) - 클라이언트 애플리케이션은 이 태그 값을 사용하여 응답에 반환된 청크가 스트림의 최신 청크보다 얼마나 뒤쳐져 있는지 확인할 수 있습니다.
- AWS_KINESISVIDEO_FRAGMENT_NUMBER - 청크에 반환된 프래그먼트 번호입니다.
- AWS_KINESISVIDEO_서버_타임스탬프 - 프래그먼트의 서버 타임스탬프.
- AWS_KINESISVIDEO_프로듀서_타임스탬프 - 프래그먼트의 프로듀서 타임스탬프.

오류가 발생할 경우 다음 태그가 표시됩니다.

- AWS_KINESISVIDEO_ERROR_CODE - 중지를 일으킨 오류에 대한 문자열 설명입니다.
GetMedia
- AWS_KINESISVIDEO_ERROR_ID: 오류의 정수 코드입니다.

오류 코드는 다음과 같습니다.

- 3002 - 스트림에 쓰는 중 오류 발생
- 4000 - 요청된 프래그먼트를 찾을 수 없음
- 4500 - 스트림의 KMS 키에 대한 액세스가 거부되었습니다.
- 4501 - 스트림의 KMS 키가 비활성화되었습니다.
- 4502 - 스트림의 KMS 키에 대한 유효성 검사 오류
- 4503 - 스트림에 지정된 KMS 키를 사용할 수 없습니다.
- 4504 - 스트림에 지정된 KMS 키가 잘못 사용되었습니다.
- 4505 - 스트림에 지정된 KMS 키의 상태가 잘못되었습니다.
- 4506 - 스트림에 지정된 KMS 키를 찾을 수 없습니다.
- 5000 - 내부 오류

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 [일반적인 오류](#) 섹션을 참조하세요.

ClientLimitExceededException

허용된 클라이언트 호출 한도를 초과했기 때문에 Kinesis Video Streams가 요청 병목 현상을 일으켰습니다. 나중에 전화를 걸어 보세요.

HTTP 상태 코드: 400

ConnectionLimitExceededException

허용된 클라이언트 연결 제한을 초과했기 때문에 Kinesis Video Streams가 요청 병목 현상을 일으켰습니다.

HTTP 상태 코드: 400

InvalidArgumentException

이 입력 파라미터의 값이 유효하지 않습니다.

HTTP 상태 코드: 400

InvalidEndpointException

호출자가 잘못된 엔드포인트를 사용하여 스트림에 데이터를 썼습니다. 이러한 예외를 수신한 사용자는 APIName set GetDataEndpoint to로 PUT_MEDIA 호출하고 응답의 엔드포인트를 사용하여 다음 PutMedia 호출을 호출해야 합니다.

HTTP 상태 코드: 400

NotAuthorizedException

호출자가 지정된 스트림에서 작업을 수행할 권한이 없거나 토큰이 만료되었습니다.

HTTP 상태 코드: 401

ResourceNotFoundException

상태 코드: 404, 지정된 이름의 스트림이 존재하지 않습니다.

HTTP 상태 코드: 404

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS 파이썬용 SDK](#)
- [AWS 루비 V3용 SDK](#)

PutMedia

서비스: Amazon Kinesis Video Streams Media

이 API를 사용하여 Kinesis 비디오 스트림으로 미디어 데이터를 전송할 수 있습니다.

Note

먼저 `GetDataEndpoint` API를 호출하여 엔드포인트를 가져와야 합니다. 그런 다음 `PutMedia` [--endpoint-url](#) 파라미터를 사용하여 이 엔드포인트로 요청을 전송합니다.

요청에서 HTTP 헤더를 사용하여 매개 변수 정보 (예: 스트림 이름, 타임스탬프, 타임스탬프 값이 절대 값인지 또는 제작자가 녹화를 시작한 시점을 기준으로 하는지 여부) 를 제공합니다. 요청 본문을 사용하여 미디어 데이터를 전송합니다. Kinesis Video Streams는 이 API를 사용하여 미디어 데이터를 전송할 때 마트로스카 (MKV) 컨테이너 형식만 지원합니다.

이 API를 사용하여 데이터를 전송하는 옵션은 다음과 같습니다.

- 미디어 데이터 실시간 전송: 예를 들어 보안 카메라가 프레임을 생성하면서 실시간으로 프레임을 전송할 수 있습니다. 이 접근 방식은 비디오 녹화와 유선으로 전송되는 데이터 사이의 지연 시간을 최소화합니다. 이를 연속 프로듀서라고 합니다. 이 경우 소비자 애플리케이션은 실시간으로 또는 필요할 때 스트림을 읽을 수 있습니다.
- 미디어 데이터를 오프라인으로 전송 (일괄): 예를 들어 바디 카메라가 몇 시간 동안 비디오를 녹화하고 이를 기기에 저장할 수 있습니다. 나중에 카메라를 도킹 포트에 연결하면 카메라가 `PutMedia` 세션을 시작하여 Kinesis 비디오 스트림으로 데이터를 전송할 수 있습니다. 이 시나리오에서는 지연 시간이 문제가 되지 않습니다.

이 API를 사용할 때는 다음 고려 사항을 참고하십시오.

- `streamName` 또는 `streamARN` 중 하나만 지정해야 하지만 두 가지 모두 지정해서는 안 됩니다.
- 콘솔에서 또는 HLS를 통해 미디어를 재생하려면 각 프래그먼트의 트랙 1에는 h.264로 인코딩된 비디오가 포함되어야 하고, 프래그먼트 메타데이터의 `CodeCid`는 "V_MPEG/ISO/AVC"여야 하며, 프래그먼트 메타데이터에는 AVCC 형식의 h.264 코덱 전용 데이터가 포함되어야 합니다. 선택적으로 각 프래그먼트의 트랙 2에는 AAC로 인코딩된 오디오가 포함되어야 하고, 프래그먼트 메타데이터의 `CodeCid`는 "A_AAC"여야 하며, 프래그먼트 메타데이터에는 AAC 코덱 프라이빗 데이터가 포함되어야 합니다.
- 장기간 실행되는 단일 `PutMedia` 세션을 사용하고 페이로드에 많은 미디어 데이터 프래그먼트를 전송하는 것이 더 쉬울 수도 있습니다. Kinesis Video Streams는 수신된 각 프래그먼트에 대해 하나 이

상의 승인을 보냅니다. 잠재적인 네트워크 고려 사항으로 인해 이러한 승인이 생성될 때 모든 승인을 받지 못할 수 있습니다.

- 서비스로부터 모든 승인을 실시간으로 받을 수 있도록 하기 위해 각 세션마다 프래그먼트 수가 적은 연속 PutMedia 세션을 여러 개 선택할 수 있습니다.

Note

여러 개의 동시 PutMedia 세션에서 동일한 스트림으로 데이터를 전송하는 경우 미디어 프래그먼트는 스트림에서 인터리빙됩니다. 애플리케이션 시나리오에서 이 문제가 없는지 확인해야 합니다.

PutMediaAPI를 사용할 때는 다음과 같은 제한이 적용됩니다.

- 클라이언트는 스트림당 초당 PutMedia 최대 5회까지 호출할 수 있습니다.
- 클라이언트는 스트림당 초당 최대 5개의 프래그먼트를 전송할 수 있습니다.
- Kinesis Video Streams는 세션 중에 최대 12.5MB/초 또는 100Mbps의 속도로 미디어 데이터를 읽습니다. PutMedia

다음 제약 조건에 유의하십시오. 이러한 경우 Kinesis Video Streams는 응답으로 오류 승인을 보냅니다.

- 최대 허용 한도보다 긴 타임 코드가 있고 50MB를 초과하는 데이터를 포함하는 프래그먼트는 허용되지 않습니다.
- 트랙이 세 개 이상 포함된 프래그먼트는 허용되지 않습니다. 모든 프래그먼트의 각 프레임은 프래그먼트 헤더에 정의된 트랙 중 하나와 동일한 트랙 번호를 가져야 합니다. 또한 모든 프래그먼트는 프래그먼트 헤더에 정의된 각 트랙에 대해 하나 이상의 프레임을 포함해야 합니다.
- 각 프래그먼트는 프래그먼트 메타데이터에 정의된 각 트랙에 대해 하나 이상의 프레임을 포함해야 합니다.
- 프래그먼트에서 가장 이른 프레임 타임스탬프는 이전 프래그먼트의 최신 프레임 타임스탬프 이후여야 합니다.
- MKV 세그먼트를 두 개 이상 포함하거나 허용되지 않는 MKV 요소 (예:) 를 포함하는 MKV 스트림도 오류 확인으로 이어집니다. track*

Kinesis Video Streams는 수신되는 각 프래그먼트와 관련 메타데이터를 소위 “청크”에 저장합니다. 프래그먼트 메타데이터에는 다음이 포함됩니다.

- 요청 시작 시 제공된 MKV 헤더 PutMedia
- 프래그먼트에 대한 다음과 같은 Kinesis Video Streams 관련 메타데이터:
 - `server_timestamp`- Kinesis Video Streams가 프래그먼트 수신을 시작한 시점의 타임스탬프입니다.
 - `producer_timestamp`- 타임스탬프: 프로듀서가 프래그먼트 녹화를 시작한 시점의 타임스탬프. Kinesis Video Streams는 요청에서 받은 세 가지 정보를 사용하여 이 값을 계산합니다.
 - 요청 본문에서 프래그먼트와 함께 수신된 프래그먼트 타임코드 값입니다.
 - 두 개의 요청 헤더: `producerStartTimeStamp` (제작자가 녹화를 시작한 시점) 및 `fragmentTimeCodeType` (페이로드의 프래그먼트 타임코드가 절대적인지 상대적인지 여부)

그러면 Kinesis Video Streams는 다음과 `producer_timestamp` 같이 프래그먼트에 대한 값을 계산합니다.

`fragmentTimeCodeType`상대적인 값인 경우

`producer_timestamp`= `producerStartTimeStamp` + 프래그먼트 타임코드

`fragmentTimeCodeType`절대값인 경우

`producer_timestamp`= 프래그먼트 타임코드 (밀리초로 변환)

- Kinesis Video Streams에서 할당된 고유한 프래그먼트 번호입니다.

Note

GetMedia요청을 하면 Kinesis Video Streams는 이러한 청크의 스트림을 반환합니다. 클라이언트는 필요에 따라 메타데이터를 처리할 수 있습니다.

Note

이 작업은 Java용 AWS SDK에서만 사용할 수 있습니다. 다른 언어용 AWS SDK에서는 지원하지 않습니다.

Note

Kinesis Video Streams는 API를 통한 수집 및 보관 중에 코덱 프라이빗 데이터를 파싱하고 검증하지 않습니다. PutMedia KVS는 HLS API를 통해 스트림을 사용할 때 MPEG-TS 및 MP4 프레임먼트 패키징을 위해 코덱 전용 데이터에서 필요한 정보를 추출하고 검증합니다.

Note

Kinesis Video Streams 미디어 API를 호출한 후 오류가 발생하는 경우 HTTP 상태 코드 및 응답 본문 외에도 다음과 같은 정보가 포함됩니다.

- `x-amz-ErrorTypeHTTP` 헤더 - HTTP 상태 코드가 제공하는 내용 외에도 보다 구체적인 오류 유형을 포함합니다.
- `x-amz-RequestIdHTTP` 헤더 — 문제를 신고하려는 경우 지원 팀에서 요청 ID를 제공하면 문제를 더 잘 진단할 수 있습니다. AWS

HTTP 상태 코드와 `ErrorType` 헤더를 모두 사용하여 오류를 재시도할 수 있는지 여부와 조건에 대해 프로그래밍 방식으로 결정할 수 있으며, 클라이언트 프로그래머가 성공적으로 다시 시도하기 위해 취해야 할 조치에 대한 정보도 제공할 수 있습니다.

[자세한 내용은 이 항목 하단의 오류 섹션 및 일반적인 오류를 참조하십시오.](#)

Request Syntax

```
POST /putMedia HTTP/1.1
x-amzn-stream-name: StreamName
x-amzn-stream-arn: StreamARN
x-amzn-fragment-timecode-type: FragmentTimecodeType
x-amzn-producer-start-timestamp: ProducerStartTimestamp
```

Payload

URI 요청 파라미터

요청은 다음 URI 파라미터를 사용합니다.

FragmentTimecodeType

이것은 `x-amzn-fragment-timecode-type` HTTP 헤더의 값으로 전달됩니다.

프래그먼트 (페이로드, HTTP 요청 본문) 의 타임코드가 절대적인지 상대적인지를 나타냅니다. `producerStartTimestamp` Kinesis Video Streams는 API 개요에 설명된 대로 이 정보를 사용하여 요청에서 수신한 프래그먼트에 대해 계산합니다 `producer_timestamp`.

유효 값: ABSOLUTE | RELATIVE

필수 사항 여부: 예

ProducerStartTimestamp

이것은 `x-amzn-producer-start-timestamp` HTTP 헤더의 값으로 전달됩니다.

이는 프로듀서가 미디어 녹화를 시작한 시점의 프로듀서 타임스탬프입니다 (요청에 있는 특정 프래그먼트의 타임스탬프는 아님).

StreamARN

이것은 `x-amzn-stream-arn` HTTP 헤더의 값으로 전달됩니다.

미디어 콘텐츠를 작성하려는 Kinesis 비디오 스트림의 Amazon 리소스 이름 (ARN) 입니다. 를 지정하지 않는 `streamARN` 경우 를 지정해야 합니다. `streamName`

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

StreamName

이것은 `x-amzn-stream-name` HTTP 헤더의 값으로 전달됩니다.

미디어 콘텐츠를 작성하려는 Kinesis 비디오 스트림의 이름입니다. 를 지정하지 않는 경우 를 지정해야 합니다. `streamName` `streamARN`

길이 제약: 최소 길이 1. 최대 길이는 256입니다.

패턴: `[a-zA-Z0-9_.-]+`

요청 본문

요청은 다음의 이진 데이터를 허용합니다.

Payload

Kinesis 비디오 스트림에 쓸 미디어 콘텐츠입니다. 현재 구현에서 Kinesis Video Streams는 단일 MKV 세그먼트가 있는 마트로스카 (MKV) 컨테이너 형식만 지원합니다. 세그먼트는 하나 이상의 클러스터를 포함할 수 있습니다.

Note

각 MKV 클러스터는 Kinesis 비디오 스트림 프래그먼트에 매핑됩니다. 어떤 클러스터 기간을 선택하든 프래그먼트 지속 시간이 됩니다.

Response Syntax

```
HTTP/1.1 200
```

```
Payload
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

응답은 다음 내용을 HTTP 본문으로 반환합니다.


Payload

Kinesis Video Streams가 요청을 성공적으로 PutMedia 수신한 후 서비스는 요청 헤더를 검증합니다. 그러면 서비스가 페이로드 읽기를 시작하고 먼저 HTTP 200 응답을 보냅니다.

그런 다음 서비스는 줄바꿈으로 구분된 일련의 JSON 객체 (Acknowledgement 객체) 가 포함된 스트림을 반환합니다. 확인은 미디어 데이터가 전송되는 동일한 연결에서 수신됩니다. 요청에는 여러 개의 승인이 있을 수 있습니다. PutMedia Acknowledgement 각각은 다음과 같은 키값 쌍으로 구성됩니다.

- AckEventType- 승인이 나타내는 이벤트 유형.
 - 버퍼링: Kinesis Video Streams가 프래그먼트를 수신하기 시작했습니다. Kinesis Video Streams는 프래그먼트 데이터의 첫 바이트가 수신되면 첫 번째 버퍼링 승인을 보냅니다.
 - 수신: Kinesis Video Streams가 전체 프래그먼트를 수신했습니다. 데이터를 유지하도록 스트림을 구성하지 않은 경우 제작자는 이 확인을 받은 후 프래그먼트의 버퍼링을 중단할 수 있습니다.

- **지속됨:** Kinesis Video Streams가 프래그먼트를 유지했습니다 (예: Amazon S3에). 데이터를 유지하도록 스트림을 구성한 경우 이 승인을 받게 됩니다. 이 승인을 받은 후 제작자는 프래그먼트의 버퍼링을 중단할 수 있습니다.
- **오류:** 프래그먼트를 처리하는 동안 Kinesis Video Streams에서 오류가 발생했습니다. 오류 코드를 검토하고 다음 조치를 결정할 수 있습니다.
- **유휴:** PutMedia 세션이 진행 중입니다. 하지만 Kinesis Video Streams는 현재 데이터를 수신하지 않습니다. Kinesis Video Streams는 마지막으로 데이터를 수신한 후 최대 30초 동안 주기적으로 이 확인 메시지를 전송합니다. 30초 내에 데이터가 수신되지 않으면 Kinesis Video Streams는 요청을 종료합니다.

 Note

이 확인은 생산자가 데이터를 전송하지 않는 경우에도 PutMedia 연결이 활성 상태인지 판단하는 데 도움이 될 수 있습니다.

- **FragmentTimecode-** 승인이 전송되는 프래그먼트 타임코드.

가 Idle인 경우 요소가 누락될 수 있습니다. **AckEventType**

- **FragmentNumber-** Kinesis Video Streams에서 생성한 프래그먼트 번호로서 확인이 전송됩니다.
- **ErrorId 및 ErrorCode -** 인 경우 AckEventType 이 필드에 해당 오류 Error 코드가 제공됩니다. 다음은 오류 ID와 해당 오류 코드 및 오류 메시지 목록입니다.
 - 4000 - STREAM_READ_ERROR - 데이터 스트림을 읽는 동안 오류가 발생했습니다.
 - 4001 - MAX_FRAGMENT_SIZE_REACHED - 프래그먼트 크기가 허용된 최대 제한값인 50MB를 초과했습니다.
 - 4002 - MAX_FRAGMENT_DURATION_REACHED - 프래그먼트 지속 시간이 최대 허용 한도보다 깁니다.
 - 4003 - MAX_CONNECTION_DURATION_REACHED - 연결 지속 시간이 최대 허용 임계값보다 큼니다.
 - 4004 - FRAGMENT_TIMECODE_LESSER_THAN_PREVERSION - 프래그먼트 타임코드가 타임코드 이전 타임 코드보다 작습니다 (호출 시 프래그먼트를 잘못된 순서로 전송할 수 없음). PutMedia
 - 4005 - MORE_THAN_ALLOWED_TRACKS_FOUND - MKV에는 둘 이상의 트랙이 있습니다. (더 이상 사용되지 않음)
 - 4006 - INVALID_MKV_DATA - 입력 스트림을 유효한 MKV 형식으로 파싱하지 못했습니다.

- 4007 - 유효하지 않은_프로듀서_타임스탬프 - 잘못된 프로듀서 타임스탬프입니다.
- 4008 - STREAM_NOT_ACTIVE - 스트림이 더 이상 존재하지 않음 (삭제됨).
- 4009 - FRAGMENT_METADATA_LIMIT_REMIT_REACHED - 프래그먼트 메타데이터 한도에 도달했습니다. [개발자 가이드의 제한 섹션](#)을 참조하십시오.
- 4010 - 트랙_번호_불일치 - MKV 프레임의 트랙 번호가 MKV 헤더의 트랙과 일치하지 않습니다.
- 4011 - FRAMES_MISSING_FOR_TRACK - 프래그먼트에 MKV 헤더에 있는 트랙 중 하나 이상에 대한 프레임이 포함되어 있지 않았습니다.
- 4012 - INVALID_FRAGMETAL_METADATA - 프래그먼트 메타데이터 이름은 문자열로 시작할 수 없습니다. AWS_
- 4500 - KMS_KEY_ACCESS_DENIED - 스트림의 지정된 KMS 키에 대한 액세스가 거부되었습니다.
- 4501 - KMS_KEY_DISABLED - 스트림에 지정된 KMS 키가 비활성화되었습니다.
- 4502 - KMS_KEY_VALIDATION_ERROR - 스트림의 지정된 KMS 키 검증에 실패했습니다.
- 4503 - KMS_KEY_UNAVAILABLE - 스트림에 지정된 KMS 키를 사용할 수 없습니다.
- 4504 - KMS_KEY_INVALID_USAGE - 스트림에 지정된 KMS 키가 잘못 사용되었습니다.
- 4505 - KMS_KEY_INVALID_STATE - 스트림의 지정된 KMS 키가 잘못된 상태입니다.
- 4506 - KMS_KEY_NOT_FOUND - 스트림의 지정된 KMS 키를 찾을 수 없습니다.
- 5000 - 내부_오류 - 내부 서비스 오류입니다.
- 5001 - ARCHIVAL_ERROR - Kinesis Video Streams가 프래그먼트를 데이터 스토어에 지속시키지 못했습니다.

Note

제작자는 장기 실행 PutMedia 요청을 위해 페이로드를 전송하는 동안 응답을 읽고 승인을 받아야 합니다. 중간 프록시 서버의 버퍼링으로 인해 프로듀서가 동시에 많은 양의 승인을 받을 수 있습니다. 적시에 승인을 받고자 하는 프로듀서는 각 요청에서 보내는 프래그먼트의 수를 줄일 수 있습니다. PutMedia

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 [일반적인 오류](#) 섹션을 참조하세요.

ClientLimitExceededException

허용된 클라이언트 호출 한도를 초과했기 때문에 Kinesis Video Streams가 요청 병목 현상을 일으켰습니다. 나중에 전화를 걸어 보세요.

HTTP 상태 코드: 400

ConnectionLimitExceededException

허용된 클라이언트 연결 제한을 초과했기 때문에 Kinesis Video Streams가 요청 병목 현상을 일으켰습니다.

HTTP 상태 코드: 400

InvalidArgumentException

이 입력 파라미터의 값이 유효하지 않습니다.

HTTP 상태 코드: 400

InvalidEndpointException

호출자가 잘못된 엔드포인트를 사용하여 스트림에 데이터를 썼습니다. 이러한 예외를 수신한 사용자는 APIName set GetDataEndpoint to로 PUT_MEDIA 호출하고 응답의 엔드포인트를 사용하여 다음 PutMedia 호출을 호출해야 합니다.

HTTP 상태 코드: 400

NotAuthorizedException

호출자가 지정된 스트림에서 작업을 수행할 권한이 없거나 토큰이 만료되었습니다.

HTTP 상태 코드: 401

ResourceNotFoundException

상태 코드: 404, 지정된 이름의 스트림이 존재하지 않습니다.

HTTP 상태 코드: 404

예제

승인 형식

승인 형식은 다음과 같습니다.

```
{
  Acknowledgement : {
    "EventType": enum
    "FragmentTimecode": Long,
    "FragmentNumber": Long,
    "ErrorId" : String
  }
}
```

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

Amazon Kinesis Video Streams 아카이브

Amazon Kinesis Video Streams 아카이브된 미디어에서 지원하는 작업은 다음과 같습니다.

- [GetClip](#)
- [GetDASHStreamingSessionURL](#)
- [GetHLSStreamingSessionURL](#)
- [GetImages](#)
- [GetMediaForFragmentList](#)
- [ListFragments](#)

GetClip

서비스: Amazon Kinesis Video Streams Archived Media

지정된 시간 범위 동안 지정된 비디오 스트림에서 보관된 온디맨드 미디어가 포함된 MP4 파일 (클립) 을 다운로드합니다.

StreamName 및 StreamARN 매개 변수는 모두 선택 사항이지만 이 API 작업을 호출할 때는 StreamName 또는 StreamARN을 지정해야 합니다.

Note

먼저 GetDataEndpoint API를 호출하여 엔드포인트를 가져와야 합니다. 그런 다음 GetClip [--endpoint-url](#) 파라미터를 사용하여 이 엔드포인트로 요청을 전송합니다.

Amazon Kinesis 비디오 스트림에는 MP4를 통해 데이터를 제공하기 위한 다음과 같은 요구 사항이 있습니다.

- [비디오 재생 트랙 요구 사항](#).
- 데이터 보존이 0보다 커야 합니다.
- 각 조각의 비디오 추적에는 H.264용 AVC(Advanced Video Coding) 형식 및 H.265용 HEVC 형식의 코덱 프라이빗 데이터가 포함되어야 합니다. 자세한 내용은 [MPEG-4 사양 ISO/IEC 14496-15](#)를 참조하십시오. 스트림 데이터를 지정된 형식에 맞게 조정하는 방법에 대한 자세한 내용은 [NAL 적응 플래그](#)를 참조하십시오.
- 각 조각의 오디오 조각에는 AAC 형식([AAC 사양 ISO/IEC 13818-7](#)) 또는 [MS Wave 형식](#)의 코덱 프라이빗 데이터가 포함되어야 합니다.

GetClip.OutgoingBytesAmazon CloudWatch 지표를 모니터링하여 발신 데이터의 양을 모니터링할 수 있습니다. Kinesis Video Streams를 모니터링하는 CloudWatch 데 사용하는 방법에 대한 자세한 내용은 Kinesis 비디오 스트림 [모니터링을 참조하십시오](#). 요금 정보는 [Amazon Kinesis Video Streams 요금 AWS 및](#) 요금을 참조하십시오. 발신 AWS 데이터에 대한 요금이 적용됩니다.

Important

각 프레임먼트에 포함된 코덱 전용 데이터 (CPD)에는 프레임먼트를 올바르게 디코딩하는 데 필요한 프레임 속도, 해상도 및 인코딩 프로파일과 같은 코덱별 초기화 정보가 포함되어 있습니다.

니다. 결과 클립의 대상 조각 간에는 CPD 변경이 지원되지 않습니다. CPD는 쿼리된 미디어에서 일관성을 유지해야 합니다. 그렇지 않으면 오류가 반환됩니다.

⚠ Important

트랙 변경은 지원되지 않습니다. 트랙은 쿼리된 미디어 전체에서 일관성을 유지해야 합니다. 스트림의 프래그먼트가 비디오만 포함하는 것에서 오디오와 비디오를 모두 포함하는 것으로 변경되거나 AAC 오디오 트랙이 A-Law 오디오 트랙으로 변경되면 오류가 반환됩니다.

Request Syntax

```
POST /getClip HTTP/1.1
Content-type: application/json

{
  "ClipFragmentSelector": {
    "FragmentSelectorType": "string",
    "TimestampRange": {
      "EndTimeStamp": number,
      "StartTimeStamp": number
    }
  },
  "StreamARN": "string",
  "StreamName": "string"
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

ClipFragmentSelector

요청된 클립의 시간 범위 및 타임스탬프의 소스.

유형: ClipFragmentSelector 객체

필수 여부: 예

StreamARN

미디어 클립을 검색할 스트림의 Amazon 리소스 이름 (ARN).

StreamName 또는 StreamARN을 지정해야 합니다.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+

Required: No

StreamName

미디어 클립을 검색할 스트림의 이름.

StreamName 또는 StreamARN을 지정해야 합니다.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256입니다.

패턴: [a-zA-Z0-9_.-]+

필수 여부: 아니요

응답 구문

```
HTTP/1.1 200  
Content-Type: ContentType
```

```
Payload
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

응답에 다음 HTTP 헤더가 반환됩니다.

ContentType

요청된 클립에 있는 미디어의 콘텐츠 유형.

길이 제약: 최소 길이 1. 최대 길이 128.

패턴: `^[a-zA-Z0-9_\.\\-]+$`

응답은 다음 내용을 HTTP 본문으로 반환합니다.

Payload

지정된 비디오 스트림의 미디어 클립이 포함된 일반 MP4 파일입니다. 출력에는 지정된 시작 타임스탬프의 처음 100MB 또는 처음 200개의 조각이 포함됩니다. 자세한 내용은 [Kinesis Video Streams 제한](#)을 참조하십시오.

Errors

모든 작업에 공통되는 오류에 대한 내용은 [일반적인 오류](#) 섹션을 참조하십시오.

ClientLimitExceededException

한도를 초과했기 때문에 Kinesis Video Streams가 요청 병목 현상을 일으켰습니다. 나중에 전화를 걸어 보세요. 제한에 대한 자세한 내용은 [Kinesis Video Streams](#) 제한을 참조하십시오.

HTTP 상태 코드: 400

InvalidArgumentException

지정된 파라미터는 제한을 초과하거나, 지원되지 않거나, 사용할 수 없습니다.

HTTP 상태 코드: 400

InvalidCodecPrivateDataException

비디오 스트림의 트랙 중 하나 이상에 있는 코덱 개인 데이터는 이 작업에 유효하지 않습니다.

HTTP 상태 코드: 400

InvalidMediaFrameException

요청된 클립에서 하나 이상의 프레임을 지정된 코덱에 따라 파싱할 수 없습니다.

HTTP 상태 코드: 400

MissingCodecPrivateDataException

비디오 스트림의 트랙 중 하나 이상에서 코덱 개인 데이터를 찾을 수 없습니다.

HTTP 상태 코드: 400

NoDataRetentionException

GetImages 데이터를 보관하지 않는 스트림 (즉, a가 0인 스트림) 에 DataRetentionInHours 대 해 요청되었습니다.

HTTP 상태 코드: 400

NotAuthorizedException

상태 코드: 403, 호출자가 지정된 스트림에서 작업을 수행할 권한이 없거나 토큰이 만료되었습니다.

HTTP 상태 코드: 401

ResourceNotFoundException

GetImages Kinesis Video Streams에서 지정한 스트림을 찾을 수 없는 경우 이 오류가 발생합니다.

GetHLSStreamingSessionURL 요청된 시간 범위 내에 LIVE_REPLAY 프래그먼트가 없는 스트림에 대해 a ON_DEMAND 또는 f가 있는 세션이 요청되거나 지난 30초 내에 LIVE 프래그먼트가 없는 스트림에 대해 a가 있는 세션이 요청되면 이 GetDASHStreamingSessionURL 오류가 발생합니다. PlaybackMode PlaybackMode

HTTP 상태 코드: 404

UnsupportedStreamMediaTypeException

재생 세션의 첫 번째 프래그먼트에 있는 트랙의 코덱 ID로는 미디어 유형 (예: h.264 또는 h.265 비디오 또는 AAC 또는 G.711 오디오) 을 결정할 수 없습니다. 트랙 1의 코덱 ID는 다음과 같아야 하고, 선택적으로 트랙 2의 코덱 ID는 다음과 같아야 합니다. V_MPEG/ISO/AVC A_AAC

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)

- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

GetDASHStreamingSessionURL

서비스: Amazon Kinesis Video Streams Archived Media

스트림의 HTTP를 통한 MPEG 동적 적응형 스트리밍 (DASH) URL을 검색합니다. 그런 다음 미디어 플레이어에서 URL을 열어 스트림 콘텐츠를 볼 수 있습니다.

StreamName 및 StreamARN 매개 변수는 모두 선택 사항이지만 이 API 작업을 StreamARN 호출할 때는 StreamName 또는 를 지정해야 합니다.

Amazon Kinesis 비디오 스트림에는 MPEG-DASH를 통해 데이터를 제공하기 위한 다음과 같은 요구 사항이 있습니다.

- [비디오 재생 트랙 요구 사항](#).
- 데이터 보존이 0보다 커야 합니다.
- 각 조각의 비디오 추적에는 H.264용 AVC(Advanced Video Coding) 형식 및 H.265용 HEVC 형식의 코덱 프라이빗 데이터가 포함되어야 합니다. 자세한 내용은 [MPEG-4 사양 ISO/IEC 14496-15](#)를 참조하십시오. 스트림 데이터를 지정된 형식에 맞게 조정하는 방법에 대한 자세한 내용은 [NAL 적응 플러그](#)를 참조하십시오.
- 각 조각의 오디오 조각에는 AAC 형식([AAC 사양 ISO/IEC 13818-7](#)) 또는 [MS Wave 형식](#)의 코덱 프라이빗 데이터가 포함되어야 합니다.

다음 절차는 Kinesis Video Streams와 함께 MPEG-DASH를 사용하는 방법을 보여줍니다.

1. GetDataEndpointAPI를 호출하여 엔드포인트를 확보하십시오. 그런 다음 GetDASHStreamingSessionURL [--endpoint-url](#) 파라미터를 사용하여 이 엔드포인트로 요청을 전송합니다.
2. 를 사용하여 MPEG-DASH URL을 검색합니다. GetDASHStreamingSessionURL Kinesis Video Streams는 MPEG-DASH 프로토콜을 사용하여 스트림의 콘텐츠에 액세스하는 데 사용할 MPEG-DASH 스트리밍 세션을 생성합니다. GetDASHStreamingSessionURL 세션의 MPEG-DASH 매니페스트 (MPEG-DASH로 스트리밍하는 데 필요한 루트 리소스) 의 인증된 URL (암호화된 세션 토큰 포함) 을 반환합니다.

Note

승인되지 않은 주체가 액세스할 수 있는 곳에 이 토큰을 공유하거나 저장하지 마십시오. 토큰은 스트림 콘텐츠에 대한 액세스를 제공합니다. AWS 자격 증명에 사용하는 것과 동일한 방법으로 토큰을 보호하십시오.

매니페스트를 통해 사용할 수 있는 미디어는 요청된 스트림, 시간 범위 및 형식으로만 구성됩니다. 다른 미디어 데이터 (예: 요청된 창 밖의 프레임 또는 대체 비트레이트) 는 사용할 수 없습니다.

3. MPEG-DASH 프로토콜을 지원하는 미디어 플레이어에 MPEG-DASH 매니페스트의 URL (암호화된 세션 토큰 포함) 을 제공하십시오. Kinesis Video Streams를 사용하면 매니페스트 URL을 통해 초기화 프래그먼트와 미디어 프래그먼트를 사용할 수 있습니다. 초기화 프래그먼트에는 스트림의 코덱 전용 데이터와 비디오 또는 오디오 디코더 및 렌더러를 설정하는 데 필요한 기타 데이터가 포함됩니다. 미디어 프래그먼트에는 인코딩된 비디오 프레임 또는 인코딩된 오디오 샘플이 포함됩니다.
4. 미디어 플레이어는 인증된 URL을 수신하고 스트림 메타데이터와 미디어 데이터를 정상적으로 요청합니다. 미디어 플레이어가 데이터를 요청하면 다음 액션을 호출합니다.

- **GetDashManifest:** 재생하려는 미디어의 메타데이터가 포함된 MPEG DASH 매니페스트를 검색합니다.
- **GetMP4 InitFragment:** MP4 초기화 프래그먼트를 검색합니다. 미디어 플레이어는 일반적으로 미디어 프래그먼트를 로드하기 전에 초기화 프래그먼트를 로드합니다. 이 프래그먼트에는 "" 및 `fytp "moov"` MP4 아톰과 미디어 플레이어 디코더를 초기화하는 데 필요한 하위 아톰이 포함되어 있습니다.

초기화 프래그먼트는 Kinesis 비디오 스트림의 프래그먼트와 일치하지 않습니다. 여기에는 미디어 플레이어가 미디어 프레임을 디코딩하는 데 필요한 스트림 및 해당 트랙의 코덱 전용 데이터만 포함됩니다.

- **GetMP4MediaFragment:** MP4 미디어 프래그먼트를 검색합니다. 이러한 프래그먼트에는 인코딩된 프래그먼트의 미디어 프레임과 타임스탬프가 들어 있는 `mdat ""` 및 "" MP4 원자와 해당 하위 원자가 포함됩니다. `moof`

Important

각 프래그먼트에 포함된 코덱 전용 데이터 (CPD) 에는 프래그먼트를 올바르게 디코딩하는 데 필요한 프레임 속도, 해상도, 인코딩 프로파일 등 코덱별 초기화 정보가 포함되어 있습니다. 스트리밍 세션 중에는 CPD 변경이 지원되지 않습니다. CPD는 쿼리된 미디어를 통해 일관성을 유지해야 합니다.

Important

트랙 변경은 지원되지 않습니다. 트랙은 쿼리된 미디어 전체에서 일관성을 유지해야 합니다. 스트림의 프래그먼트가 비디오만 포함되다가 오디오와 비디오를 모두 포함하는 것으

로 변경되거나 AAC 오디오 트랙이 A-Law 오디오 트랙으로 변경되면 스트리밍이 실패합니다.

이 작업으로 검색된 데이터에는 요금이 청구됩니다. 자세한 내용은 [요금](#)을 참조하세요.

Note

MPEG-DASH 세션에 적용되는 제한 사항은 [Kinesis Video Streams](#) 제한을 참조하십시오.

GetMP4MediaFragment.OutgoingBytesAmazon CloudWatch 지표를 모니터링하여 미디어 플레이어 소비하는 데이터의 양을 모니터링할 수 있습니다. Kinesis Video Streams를 모니터링하는 CloudWatch 데 사용하는 방법에 대한 자세한 내용은 Kinesis 비디오 스트림 [모니터링을 참조하십시오](#). 요금 정보는 [Amazon Kinesis Video Streams](#) 요금 [AWS 및](#) 요금을 참조하십시오. HLS 세션과 발신 AWS 데이터에 대한 요금이 모두 적용됩니다.

HLS에 대한 자세한 내용은 [Apple](#) 개발자 사이트의 [HTTP 라이브 스트리밍](#)을 참조하십시오.

Important

Kinesis Video Streams 아카이브 미디어 API를 호출한 후 오류가 발생하는 경우 HTTP 상태 코드 및 응답 본문 외에도 다음과 같은 정보가 포함됩니다.

- x-amz-ErrorTypeHTTP 헤더 — HTTP 상태 코드가 제공하는 내용 외에도 보다 구체적인 오류 유형을 포함합니다.
- x-amz-RequestIdHTTP 헤더 — 지원 팀에 문제를 보고하려는 경우 요청 ID를 제공하면 문제를 더 잘 진단할 수 있습니다. AWS

HTTP 상태 코드와 ErrorType 헤더를 모두 사용하여 오류를 재시도할 수 있는지 여부와 조건에 대해 프로그래밍 방식으로 결정할 수 있으며, 클라이언트 프로그래머가 성공적으로 다시 시도하기 위해 취해야 할 조치에 대한 정보도 제공할 수 있습니다.

[자세한 내용은 이 항목 하단의 오류 섹션 및 일반적인 오류를 참조하십시오.](#)

Request Syntax

```
POST /getDASHStreamingSessionURL HTTP/1.1
Content-type: application/json

{
  "DASHFragmentSelector": {
    "FragmentSelectorType": "string",
    "TimestampRange": {
      "EndTimestamp": number,
      "StartTimestamp": number
    }
  },
  "DisplayFragmentNumber": "string",
  "DisplayFragmentTimestamp": "string",
  "Expires": number,
  "MaxManifestFragmentResults": number,
  "PlaybackMode": "string",
  "StreamARN": "string",
  "StreamName": "string"
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

[DASHFragmentSelector](#)

요청된 프래그먼트의 시간 범위 및 타임스탬프의 소스.

이 파라미터는 또는 인 경우 PlaybackMode 필수입니다. ON_DEMAND LIVE_REPLAY 있는 경우 이 매개변수는 PlaybackMode 선택사항입니다 LIVE. PlaybackModeLIVE인 경우는 설정할 FragmentSelectorType 수 있지만 TimestampRange 설정해서는 안 됩니다. ON_DEMAND 또는 PlaybackMode LIVE_REPLAY 인 경우 둘 다 FragmentSelectorType TimestampRange 설정해야 합니다.

유형: [DASHFragmentSelector](#) 객체

필수 여부: 아니요

[DisplayFragmentNumber](#)

프래그먼트는 세션의 시퀀스 번호를 기반으로 매니페스트 파일에서 식별됩니다. 로 ALWAYS 설정하면 Kinesis Video Streams 프래그먼트 번호가 속성 이름이 “kvs:fn”인 매니페스트 파일의 각 S 요소에 추가됩니다. DisplayFragmentNumber 이러한 프래그먼트 번호는 로깅에 사용하거나 다른 API (예: [및](#)) 와 함께 사용할 수 있습니다. GetMedia GetMediaForFragmentList 이러한 사용자 지정 속성을 활용하려면 사용자 지정 MPEG-DASH 미디어 플레이어에 필요합니다.

기본 값은 NEVER입니다.

타입: 문자열

유효 값: ALWAYS | NEVER

필수 여부: 아니요

[DisplayFragmentTimestamp](#)

MPEG-DASH 사양에 따라 매니페스트 파일에 있는 프래그먼트의 월 클럭 시간은 매니페스트 자체의 속성을 사용하여 도출할 수 있습니다. 그러나 일반적으로 MPEG-DASH 호환 미디어 플레이어는 미디어 타임라인의 격차를 제대로 처리하지 못합니다. Kinesis Video Streams는 매니페스트 파일의 미디어 타임라인을 조정하여 불연속성이 있는 미디어를 재생할 수 있도록 합니다. 따라서 매니페스트 파일에서 파생된 벽시계 시간이 정확하지 않을 수 있습니다. 로 설정하면 속성 이름이 “ALWAYSkvs:ts”인 매니페스트 파일의 각 S 요소에 정확한 프래그먼트 타임스탬프가 추가됩니다. DisplayFragmentTimestamp 이 사용자 지정 특성을 활용하려면 사용자 지정 MPEG-DASH 미디어 플레이어에 필요합니다.

기본 값은 NEVER입니다. [DASHFragmentSelector](#)SERVER_TIMESTAMP인 경우 타임스탬프는 서버 시작 타임스탬프가 됩니다. 마찬가지로, [DASHFragmentSelector](#) is가 PRODUCER_TIMESTAMP 되면 타임스탬프가 프로듀서 시작 타임스탬프가 됩니다.

타입: 문자열

유효 값: ALWAYS | NEVER

필수 여부: 아니요

[Expires](#)

요청된 세션이 만료될 때까지의 시간 (초). 이 값은 300 (5분) 에서 43200 (12시간) 사이일 수 있습니다.

세션이 만료되면 해당 세션에 대해 새 전화를 걸거나 전화를 걸 `GetMP4MediaFragment` 수 없습니다. `GetDashManifest` `GetMP4InitFragment`

기본값은 300(5분)입니다.

유형: 정수

유효 범위: 최소값은 300입니다. 최대 값은 43200입니다.

필수 여부: 아니요

MaxManifestFragmentResults

MPEG-DASH 매니페스트에서 반환되는 최대 프래그먼트 수입니다.

인 `PlaybackMode` 경우 가장 최근의 프래그먼트가 이 값까지 반환됩니다. LIVE is인 경우 `PlaybackMode` 이 `ON_DEMAND` 최대 개수까지 가장 오래된 프래그먼트가 반환됩니다.

라이브 MPEG-DASH 매니페스트에서 사용할 수 있는 프래그먼트 수가 많으면 비디오 플레이어는 재생을 시작하기 전에 콘텐츠를 버퍼링하는 경우가 많습니다. 버퍼 크기를 늘리면 재생 지연 시간은 늘어나지만 재생 중에 리버퍼링이 발생할 가능성은 줄어듭니다. 라이브 MPEG-DASH 매니페스트에는 최소 3개의 프래그먼트와 최대 10개의 프래그먼트가 있는 것이 좋습니다.

기본값은 또는 인 경우 5개 프래그먼트이고, `PlaybackMode` 인 경우 1,000개입니다LIVE. `LIVE_REPLAY` `PlaybackMode` `ON_DEMAND`

최대 값인 1,000개의 프래그먼트는 1초 프래그먼트가 있는 스트림의 경우 16분 이상의 비디오에 해당하고, 10초 프래그먼트가 포함된 스트림의 경우 2시간 반 이상의 비디오에 해당합니다.

타입: Long

유효 범위: 최소값 1. 최대값은 5000입니다.

필수 여부: 아니요

PlaybackMode

실시간, 실시간 리플레이 또는 아카이브된 온디맨드 데이터 중 무엇을 검색할지 여부.

세 가지 유형의 세션의 특징은 다음과 같습니다.

- **LIVE:** 이 유형의 세션의 경우 MPEG-DASH 매니페스트는 사용 가능한 최신 프래그먼트로 계속 업데이트됩니다. 미디어 플레이어가 1초 간격으로 새 매니페스트를 검색하는 것이 좋습니다. 미

디어 플레이어에서 이러한 유형의 세션을 재생하면 일반적으로 사용자 인터페이스에 “실시간” 알림이 표시되며, 이 알림은 재생 창에서 표시할 위치를 선택할 수 있는 스크러버 컨트롤이 없습니다.

Note

LIVE모드에서는 프래그먼트 사이에 간격이 있더라도 (즉, 프래그먼트가 없는 경우) 사용 가능한 최신 프래그먼트가 MPEG-DASH 매니페스트에 포함됩니다. 이와 같은 간격이 생기면 미디어 플레이어가 멈추거나 재생이 갑자기 중단될 수 있습니다. 이 모드에서는 플레이리스트의 최신 프래그먼트보다 오래된 프래그먼트가 MPEG-DASH 매니페스트에 추가되지 않습니다. 후속 프래그먼트가 매니페스트에 추가된 후 누락된 프래그먼트를 사용할 수 있게 되면 이전 프래그먼트는 추가되지 않고 공백도 메워지지 않습니다.

- **LIVE_REPLAY:** 이 유형의 세션에서 MPEG-DASH 매니페스트는 지정된 시작 시간의 프래그먼트를 포함하여 시작한다는 점을 제외하면 LIVE 모드 업데이트 방식과 비슷하게 업데이트됩니다. 프래그먼트가 인제스트될 때 추가되는 대신 다음 프래그먼트의 지속 시간이 경과하면 프래그먼트가 추가됩니다. 예를 들어 세션의 프래그먼트 길이가 2초인 경우 2초마다 매니페스트에 새 프래그먼트가 추가됩니다. 이 모드는 이벤트가 감지된 시점부터 재생을 시작하고 세션 생성 시점에 아직 인제스트되지 않은 미디어를 계속 라이브 스트리밍할 수 있는 데 유용합니다. 이 모드는 모드에서 1,000개의 프래그먼트 한도에 제한되지 않고 이전에 보관된 미디어를 스트리밍하는 데에도 유용합니다. ON_DEMAND
- **ON_DEMAND:** 이 유형의 세션의 경우 MPEG-DASH 매니페스트에는 세션의 모든 프래그먼트가 지정된 수까지 포함됩니다. MaxManifestFragmentResults 매니페스트는 각 세션마다 한 번만 검색해야 합니다. 미디어 플레이어에서 이러한 유형의 세션을 재생하면 일반적으로 사용자 인터페이스에는 표시할 재생 창의 위치를 선택할 수 있는 스크러버 컨트롤이 표시됩니다.

모든 재생 모드에서 동일한 시작 타임스탬프를 가진 프래그먼트가 여러 개 있는 경우 프래그먼트 수가 더 큰 프래그먼트 (즉, 최신 프래그먼트)가 MPEG-DASH 매니페스트에 포함됩니다. FragmentSelectorType PRODUCER_TIMESTAMP 다른 프래그먼트는 포함되지 않습니다. 타임스탬프는 다르지만 지속 시간이 겹치는 프래그먼트는 여전히 MPEG-DASH 매니페스트에 포함됩니다. 이로 인해 미디어 플레이어에서 예상치 못한 동작이 발생할 수 있습니다.

기본값은 LIVE입니다.

타입: 문자열

유효 값: LIVE | LIVE_REPLAY | ON_DEMAND

필수 여부: 아니요

StreamARN

MPEG-DASH 매니페스트 URL을 검색할 스트림의 Amazon 리소스 이름 (ARN) 입니다.

또는 를 지정해야 합니다. StreamName StreamARN

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+

Required: No

StreamName

MPEG-DASH 매니페스트 URL을 검색할 스트림의 이름입니다.

또는 를 지정해야 합니다. StreamName StreamARN

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256입니다.

패턴: [a-zA-Z0-9_.-]+

필수 여부: 아니요

응답 구문

```

HTTP/1.1 200
Content-type: application/json

{
  "DASHStreamingSessionURL": "string"
}

```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

DASHStreamingSessionURL

미디어 플레이어가 MPEG-DASH 매니페스트를 검색하는 데 사용할 수 있는 URL (세션 토큰 포함).

타입: 문자열

Errors

모든 작업에 공통되는 오류에 대한 내용은 [일반적인 오류](#) 섹션을 참조하십시오.

ClientLimitExceededException

한도를 초과했기 때문에 Kinesis Video Streams가 요청 병목 현상을 일으켰습니다. 나중에 전화를 걸어 보세요. 제한에 대한 자세한 내용은 [Kinesis Video Streams](#) 제한을 참조하십시오.

HTTP 상태 코드: 400

InvalidArgumentException

지정된 파라미터는 제한을 초과하거나, 지원되지 않거나, 사용할 수 없습니다.

HTTP 상태 코드: 400

InvalidCodecPrivateDataException

비디오 스트림의 트랙 중 하나 이상에 있는 코덱 개인 데이터는 이 작업에 유효하지 않습니다.

HTTP 상태 코드: 400

MissingCodecPrivateDataException

비디오 스트림의 트랙 중 하나 이상에서 코덱 개인 데이터를 찾을 수 없습니다.

HTTP 상태 코드: 400

NoDataRetentionException

GetImages 데이터를 보관하지 않는 스트림 (즉, a가 0인 스트림) 에 DataRetentionInHours 대해 요청되었습니다.

HTTP 상태 코드: 400

NotAuthorizedException

상태 코드: 403, 호출자가 지정된 스트림에서 작업을 수행할 권한이 없거나 토큰이 만료되었습니다.

HTTP 상태 코드: 401

ResourceNotFoundException

GetImagesKinesis Video Streams에서 지정한 스트림을 찾을 수 없는 경우 이 오류가 발생합니다.

GetHLSStreamingSessionURL요청된 시간 범위 내에 LIVE_REPLAY 프래그먼트가 없는 스트림에 대해 a ON_DEMAND 또는 f가 있는 세션이 요청되거나 지난 30초 내에 LIVE 프래그먼트가 없는 스트림에 대해 a가 있는 세션이 요청되면 이 GetDASHStreamingSessionURL 오류가 발생합니다. PlaybackMode PlaybackMode

HTTP 상태 코드: 404

UnsupportedStreamMediaTypeException

재생 세션의 첫 번째 프래그먼트에 있는 트랙의 코덱 ID로는 미디어 유형 (예: h.264 또는 h.265 비디오 또는 AAC 또는 G.711 오디오) 을 결정할 수 없습니다. 트랙 1의 코덱 ID는 다음과 같아야 하고, 선택적으로 트랙 2의 코덱 ID는 다음과 같아야 합니다. V_MPEG/ISO/AVC A_AAC

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS 파이썬용 SDK](#)
- [AWS 루비 V3용 SDK](#)

GetHLSStreamingSessionURL

서비스: Amazon Kinesis Video Streams Archived Media

스트림의 HTTP 라이브 스트리밍 (HLS) URL을 검색합니다. 그런 다음 브라우저 또는 미디어 플레이어에서 URL을 열어 스트림 콘텐츠를 볼 수 있습니다.

StreamName 및 StreamARN 매개 변수는 모두 선택 사항이지만 이 API 작업을 StreamARN 호출할 때는 StreamName 또는 를 지정해야 합니다.

Amazon Kinesis 비디오 스트림에는 HLS를 통해 데이터를 제공하기 위한 다음과 같은 요구 사항이 있습니다.

- [비디오 재생 트랙 요구 사항](#).
- 데이터 보존이 0보다 커야 합니다.
- [각 프래그먼트의 비디오 트랙에는 H.264 형식의 경우 고급 비디오 코딩 \(AVC\), H.265 형식의 경우 HEVC \(MPEG-4 사양 ISO/IEC 14496-15\) 의 코덱 전용 데이터가 포함되어야 합니다.](#) 스트림 데이터를 지정된 형식에 맞게 조정하는 방법에 대한 자세한 내용은 [NAL 적응 플래그](#)를 참조하십시오.
- [각 프래그먼트의 오디오 트랙 \(있는 경우\)에는 AAC 형식 \(AAC 사양 ISO/IEC 13818-7\) 의 코덱 전용 데이터가 포함되어야 합니다.](#)

Kinesis Video Streams HLS 세션에는 프래그먼트된 MPEG-4 형식 (fMP4 또는 CMAF라고도 함) 또는 MPEG-2 형식 (TS 청크라고도 하며 HLS 사양에서도 지원) 의 프래그먼트가 포함되어 있습니다. [HLS 프래그먼트 유형에 대한 자세한 내용은 HLS 사양을 참조하십시오.](#)

다음 절차는 Kinesis Video Streams와 함께 HLS를 사용하는 방법을 보여줍니다.

1. GetDataEndpointAPI를 호출하여 엔드포인트를 확보하십시오. 그런 다음 GetHLSStreamingSessionURL [--endpoint-url](#) 파라미터를 사용하여 이 엔드포인트로 요청을 전송합니다.
2. 를 사용하여 HLS URL을 검색합니다. GetHLSStreamingSessionURL Kinesis Video Streams는 HLS 프로토콜을 사용하여 스트림의 콘텐츠에 액세스하는 데 사용할 HLS 스트리밍 세션을 생성합니다. GetHLSStreamingSessionURL 세션의 HLS 마스터 플레이리스트 (HLS를 통한 스트리밍에 필요한 루트 리소스) 에 대한 인증된 URL (암호화된 세션 토큰 포함) 을 반환합니다.

Note

승인되지 않은 주체가 액세스할 수 있는 곳에 이 토큰을 공유하거나 보관하지 마십시오. 토큰은 스트림 콘텐츠에 대한 액세스를 제공합니다. AWS 자격 증명에 사용하는 것과 동일한 방법으로 토큰을 보호하십시오.

재생 목록을 통해 제공되는 미디어는 요청된 스트림, 시간 범위 및 형식으로만 구성됩니다. 다른 미디어 데이터 (예: 요청된 창 밖의 프레임 또는 대체 비트레이트) 는 사용할 수 없습니다.

3. HLS 프로토콜을 지원하는 미디어 플레이어에 HLS 마스터 재생 목록의 URL (암호화된 세션 토큰 포함) 을 제공하십시오. Kinesis Video Streams를 사용하면 HLS 미디어 플레이리스트, 초기화 프래그먼트, 미디어 프래그먼트를 마스터 플레이리스트 URL을 통해 사용할 수 있습니다. 초기화 프래그먼트에는 스트림의 코덱 전용 데이터와 비디오 또는 오디오 디코더 및 렌더러를 설정하는 데 필요한 기타 데이터가 포함되어 있습니다. 미디어 조각에는 H.264로 인코딩된 비디오 프레임 또는 AAC로 인코딩된 오디오 샘플이 포함되어 있습니다.
4. 미디어 플레이어는 인증된 URL을 수신하고 스트림 메타데이터 및 미디어 데이터를 정상적으로 요청합니다. 미디어 플레이어가 데이터를 요청하면 다음 액션을 호출합니다.
 - **GetHLSMasterPlaylist:** 각 트랙의 **GetHLSMediaPlaylist** 액션 URL과 예상 비트레이트 및 해상도를 포함한 미디어 플레이어에 대한 추가 메타데이터가 포함된 HLS 마스터 재생 목록을 검색합니다.
 - **GetHLSMediaPlaylist:** HLS 미디어 재생 목록을 검색합니다. 이 목록에는 액션으로 MP4 초기화 프래그먼트에 액세스할 수 있는 URL과 액션으로 MP4 미디어 프래그먼트에 액세스할 수 있는 URL이 **GetMP4InitFragment** 포함되어 있습니다. **GetMP4MediaFragment** HLS 미디어 재생 목록에는 플레이어가 재생하는 데 필요한 스트림에 대한 메타데이터 (예: 스트리밍 여부) 도 포함되어 있습니다. **PlaybackMode LIVE ON_DEMAND** HLS 미디어 플레이리스트는 일반적으로 **a**가 설정된 세션에서는 정적입니다. **PlaybackType ON_DEMAND f**가 설정된 세션의 경우 HLS 미디어 플레이리스트가 새 프래그먼트로 계속 업데이트됩니다. **PlaybackType LIVE** 비디오 트랙을 위한 별도의 HLS 미디어 재생 목록과 특정 트랙의 MP4 미디어 URL을 포함하는 오디오 트랙 (해당하는 경우) 을 위한 별도의 HLS 미디어 재생 목록이 있습니다.
 - **GetMP4InitFragment:** MP4 초기화 프래그먼트를 검색합니다. 미디어 플레이어는 일반적으로 미디어 프래그먼트를 로드하기 전에 초기화 프래그먼트를 로드합니다. 이 프래그먼트에는 "" 및 **fytp "moov"** MP4 아톰과 미디어 플레이어 디코더를 초기화하는 데 필요한 하위 아톰이 포함되어 있습니다.

초기화 프래그먼트는 Kinesis 비디오 스트림의 프래그먼트와 일치하지 않습니다. 여기에는 미디어 플레이어가 미디어 프레임을 디코딩하는 데 필요한 스트림 및 해당 트랙의 코덱 전용 데이터만 포함됩니다.

- `GetMP4MediaFragment`: MP4 미디어 프래그먼트를 검색합니다. 이러한 프래그먼트에는 인코딩된 프래그먼트의 미디어 프레임과 타임스탬프가 들어 있는 `mdat` 및 `moof` MP4 원자와 해당 하위 원자가 포함됩니다.

Note

각 프래그먼트에 포함된 코덱 전용 데이터 (CPD)에는 프래그먼트를 올바르게 디코딩하는 데 필요한 프레임 속도, 해상도 및 인코딩 프로파일과 같은 코덱별 초기화 정보가 포함되어 있습니다. TS와 MP4 모두 스트리밍 세션 중에 CPD 변경이 지원됩니다. 따라서 재생을 중단하지 않고도 세션의 프래그먼트가 CPD에서 다른 정보를 가질 수 있습니다. 각 스트리밍 세션에서 CPD 변경은 500개만 허용됩니다.

Important

트랙 변경은 지원되지 않습니다. 트랙은 쿼리된 미디어 전체에서 일관성을 유지해야 합니다. 스트림의 프래그먼트가 비디오만 포함되다가 오디오와 비디오를 모두 포함하는 것으로 변경되거나 AAC 오디오 트랙이 A-Law 오디오 트랙으로 변경되면 스트리밍이 실패합니다.

이 작업으로 검색된 데이터에는 요금이 청구됩니다. 자세한 내용은 [요금](#)을 참조하십시오.

- `GettsFragment`: 스트림의 모든 트랙에 대한 초기화 데이터와 미디어 데이터를 모두 포함하는 MPEG TS 프래그먼트를 검색합니다.

Note

인 경우 스트림 미디어 `ContainerFormat` 대신 `MPEG_TS` 이 API를 사용하여 스트림 미디어를 검색합니다. `GetMP4InitFragment` `GetMP4MediaFragment`

이 작업으로 검색된 데이터에는 요금이 청구됩니다. 자세한 내용은 [Kinesis Video Streams 요금](#)을 참조하십시오.

스트리밍 세션 URL은 플레이어 간에 공유해서는 안 됩니다. 여러 미디어 플레이어가 세션을 공유하는 경우 서비스가 세션을 제한할 수 있습니다. 연결 제한에 대해서는 [Kinesis Video Streams](#) 제한을 참조하십시오.

GetMP4MediaFragment.OutgoingBytesAmazon CloudWatch 지표를 모니터링하여 미디어 플레이어가 소비하는 데이터의 양을 모니터링할 수 있습니다. Kinesis Video Streams를 모니터링하는 CloudWatch 데 사용하는 방법에 대한 자세한 내용은 Kinesis 비디오 스트림 [모니터링을 참조하십시오](#). 요금 정보는 [Amazon Kinesis Video Streams](#) 요금 [AWS 및](#) 요금을 참조하십시오. HLS 세션과 발신 AWS 데이터에 대한 요금이 모두 적용됩니다.

설명서 가이드의 비디오 재생 예시를 참조하십시오: [를 AWS CLI 사용하여 HLS 스트리밍 세션 URL을 검색할 수 있습니다](#). 및. [예: HTML에서 HLS를 사용하고 JavaScript](#)

HLS에 대한 자세한 내용은 [Apple 개발자 사이트의 HTTP 라이브 스트리밍](#)을 참조하십시오.

Important

Kinesis Video Streams 아카이브 미디어 API를 호출한 후 오류가 발생하는 경우 HTTP 상태 코드 및 응답 본문 외에도 다음과 같은 정보가 포함됩니다.

- x-amz-ErrorTypeHTTP 헤더 — HTTP 상태 코드가 제공하는 내용 외에도 보다 구체적인 오류 유형을 포함합니다.
- x-amz-RequestIdHTTP 헤더 — 문제를 신고하려는 경우 지원 팀에서 요청 ID를 제공하면 문제를 더 잘 진단할 수 있습니다. AWS

HTTP 상태 코드와 ErrorType 헤더를 모두 사용하여 오류를 재시도할 수 있는지 여부와 조건에 대해 프로그래밍 방식으로 결정할 수 있으며, 클라이언트 프로그래머가 성공적으로 다시 시도하기 위해 취해야 할 조치에 대한 정보도 제공할 수 있습니다.

[자세한 내용은 이 항목 하단의 오류 섹션 및 일반적인 오류를 참조하십시오.](#)

Request Syntax

```
POST /getHLSStreamingSessionURL HTTP/1.1
Content-type: application/json

{
  "ContainerFormat": "string",
  "DiscontinuityMode": "string",
```

```

    "DisplayFragmentTimestamp": "string",
    "Expires": number,
    "HLSFragmentSelector": {
      "FragmentSelectorType": "string",
      "TimestampRange": {
        "EndTimestamp": number,
        "StartTimestamp": number
      }
    },
    "MaxMediaPlaylistFragmentResults": number,
    "PlaybackMode": "string",
    "StreamARN": "string",
    "StreamName": "string"
  }

```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

ContainerFormat

미디어 패키징에 사용해야 하는 형식을 지정합니다. FRAGMENTED_MP4 컨테이너 형식을 지정하면 미디어가 MP4 조각 (FMP4 또는 CMAF) 으로 패키징됩니다. 패키징 오버헤드가 최소화되므로 이 패키징을 사용하는 것이 좋습니다. 다른 컨테이너 형식 옵션은 없습니다. MPEG_TS HLS는 출시 이후 MPEG TS 청크를 지원해 왔으며, 구형 HLS 플레이어에서 유일하게 지원되는 패키징이기도 합니다. MPEG TS는 일반적으로 5~ 25% 의 패키징 오버헤드를 가집니다. 즉, MPEG TS는 일반적으로 FMP4보다 5~ 25% 더 많은 대역폭과 비용이 필요합니다.

기본값은 FRAGMENTED_MP4입니다.

타입: 문자열

유효 값: FRAGMENTED_MP4 | MPEG_TS

필수 여부: 아니요

DiscontinuityMode

프래그먼트 간 불연속성을 표시하는 플래그가 미디어 재생 목록에 추가되는 시기를 지정합니다.

미디어 플레이어는 일반적으로 각 프래그먼트의 타임스탬프를 기반으로 재생할 미디어 콘텐츠의 타임라인을 만듭니다. 즉, 프래그먼트 사이에 겹치는 부분이나 간격이 있는 경우 (일반적으로 로 설정된 경우 [HLSFragmentSelector](#)SERVER_TIMESTAMP) 미디어 플레이어 타임라인의 일부 위치에서는 프래그먼트 사이에 작은 간격이 생기고 다른 곳에서는 프레임을 덮어씁니다. 미디어 플레이어 타임라인의 간격이 있으면 재생이 중단되고 겹치면 재생이 불안정해질 수 있습니다. 프래그먼트 사이에 불연속성 플래그가 있는 경우 미디어 플레이어가 타임라인을 재설정하여 이전 프래그먼트 바로 다음에 다음 프래그먼트가 재생될 것으로 예상됩니다.

지원되는 모드는 다음과 같습니다.

- ALWAYS: HLS 미디어 재생 목록의 모든 프래그먼트 사이에 불연속성 마커가 배치됩니다. 프래그먼트 타임스탬프가 ALWAYS 정확하지 않은 경우 값을 사용하는 것이 좋습니다.
- NEVER: 불연속성 마커는 어디에도 배치되지 않습니다. 미디어 플레이어 타임라인이 프로듀서 타임스탬프에 가장 정확하게 매핑되도록 NEVER 하려면 의 값을 사용하는 것이 좋습니다.
- ON_DISCONTINUITY: 50밀리초 이상의 간격이나 겹침이 있는 조각 사이에 불연속성 마커를 배치합니다. 대부분의 재생 시나리오에서는 미디어 타임라인에 심각한 문제 (예: 조각 누락) 가 있는 경우에만 미디어 플레이어 타임라인이 ON_DISCONTINUITY 재설정되도록 값을 사용하는 것이 좋습니다.

기본값은 로 설정된 ALWAYS [HLSFragmentSelector](#) NEVER 경우와 로 SERVER_TIMESTAMP PRODUCER_TIMESTAMP 설정된 시점입니다.

타입: 문자열

유효 값: ALWAYS | NEVER | ON_DISCONTINUITY

필수 여부: 아니요

[DisplayFragmentTimestamp](#)

프래그먼트 시작 타임스탬프를 HLS 미디어 재생 목록에 포함해야 하는 시기를 지정합니다. 일반적으로 미디어 플레이어는 플레이헤드 위치를 재생 세션의 첫 번째 프래그먼트 시작을 기준으로 한 시간으로 보고합니다. 하지만 HLS 미디어 재생 목록에 시작 타임스탬프가 포함된 경우 일부 미디어 플레이어는 프래그먼트 타임스탬프를 기반으로 현재 플레이헤드를 절대 시간으로 보고할 수 있습니다. 이는 시청자에게 미디어의 벽시계 시간을 보여주는 재생 경험을 만드는 데 유용할 수 있습니다.

기본값은 NEVER입니다. [HLSFragmentSelector](#)SERVER_TIMESTAMP인 경우 타임스탬프는 서버 시작 타임스탬프가 됩니다. 마찬가지로, [HLSFragmentSelector](#) is가 PRODUCER_TIMESTAMP 되면 타임스탬프가 프로듀서 시작 타임스탬프가 됩니다.

타입: 문자열

유효 값: ALWAYS | NEVER

필수 여부: 아니요

Expires

요청된 세션이 만료될 때까지의 시간 (초) 이 값은 300 (5분) 에서 43200 (12시간) 사이일 수 있습니다.

세션이 만료되면 해당 세션에 대해 `GetHLSMasterPlaylist`, `GetHLSMediaPlaylist`, `GetMP4InitFragment`, `GetMP4MediaFragment`, 또는 전화를 새로 걸 `GetTSFragment` 수 없습니다.

기본값은 300(5분)입니다.

유형: 정수

유효 범위: 최소값은 300입니다. 최대 값은 43200입니다.

필수 여부: 아니요

HLSFragmentSelector

요청된 프래그먼트의 시간 범위와 타임스탬프의 소스.

이 파라미터는 또는 인 경우 `PlaybackMode` 필수입니다. `ON_DEMAND LIVE_REPLAY` 있는 경우 이 매개변수는 `PlaybackMode` 선택사항입니다 `LIVE`. `PlaybackMode LIVE`인 경우는 설정할 `FragmentSelectorType` 수 있지만 `TimestampRange` 설정해서는 안 됩니다. `ON_DEMAND` 또는 `PlaybackMode LIVE_REPLAY` 인 경우 둘 다 `FragmentSelectorType` `TimestampRange` 설정해야 합니다.

유형: [HLSFragmentSelector](#) 객체

필수 여부: 아니요

MaxMediaPlaylistFragmentResults

HLS 미디어 플레이리스트에서 반환되는 최대 프래그먼트 수입니다.

인 `PlaybackMode` 경우 가장 최근의 프래그먼트가 이 값까지 반환됩니다. `LIVE is`인 경우 `PlaybackMode` 이 `ON_DEMAND` 최대 개수까지 가장 오래된 프래그먼트가 반환됩니다.

라이브 HLS 미디어 재생 목록에서 사용할 수 있는 프래그먼트 수가 많으면 비디오 플레이어가 재생을 시작하기 전에 콘텐츠를 버퍼링하는 경우가 많습니다. 버퍼 크기를 늘리면 재생 지연 시간은 늘어나지만 재생 중에 리버퍼링이 발생할 가능성은 줄어듭니다. 라이브 HLS 미디어 재생 목록에는 최소 3개의 프래그먼트와 최대 10개의 프래그먼트가 있는 것이 좋습니다.

기본값은 LIVE 또는 인 경우 5개 프래그먼트이고 LIVE_REPLAY, PlaybackMode 인 경우 1,000개입니다. PlaybackMode ON_DEMAND

최대 값인 5,000개의 프래그먼트는 1초 프래그먼트가 있는 스트림의 경우 80분 이상의 비디오에 해당하고, 10초 프래그먼트가 있는 스트림의 경우 13시간 이상의 비디오에 해당합니다.

타입: Long

유효 범위: 최소값 1. 최대값은 5000입니다.

필수 여부: 아니요

PlaybackMode

실시간, 실시간 리플레이 또는 아카이브된 온디맨드 데이터 중 무엇을 검색할지 여부.

세 가지 유형의 세션의 특징은 다음과 같습니다.

- **LIVE:** 이 유형의 세션의 경우 HLS 미디어 재생 목록은 최신 프래그먼트가 제공되는 대로 지속적으로 업데이트됩니다. 미디어 플레이어가 1초 간격으로 새 재생 목록을 검색하는 것이 좋습니다. 미디어 플레이어에서 이러한 유형의 세션을 재생하면 일반적으로 사용자 인터페이스에 “실시간” 알림이 표시되며, 이 알림은 재생 창에서 표시할 위치를 선택할 수 있는 스크러버 컨트롤이 없습니다.

Note

LIVE모드에서는 프래그먼트 사이에 간격이 있더라도 (즉, 프래그먼트가 없는 경우) 사용 가능한 최신 프래그먼트가 HLS 미디어 재생 목록에 포함됩니다. 이와 같은 간격이 생기면 미디어 플레이어가 멈추거나 재생이 갑자기 중단될 수 있습니다. 이 모드에서는 플레이리스트의 최신 프래그먼트보다 오래된 프래그먼트는 HLS 미디어 플레이리스트에 추가되지 않습니다. 다음 프래그먼트가 재생 목록에 추가된 후 누락된 프래그먼트를 사용할 수 있게 되면 이전 프래그먼트는 추가되지 않고 공백도 채워지지 않습니다.

- **LIVE_REPLAY:** 이 유형의 세션에서 HLS 미디어 재생 목록은 지정된 시작 시간의 프래그먼트를 포함하여 시작한다는 점을 제외하면 LIVE 모드 업데이트 방식과 비슷하게 업데이트됩니다. 프래그먼트가 인제스트될 때 추가되는 대신 다음 프래그먼트의 지속 시간이 경과하면 프래그먼트가 추가됩니다. 예를 들어 세션의 프래그먼트 길이가 2초인 경우 2초마다 새 프래그먼트가 미디

어 재생 목록에 추가됩니다. 이 모드는 이벤트가 감지된 시점부터 재생을 시작하고 세션 생성 시점에 아직 인제스트되지 않은 미디어를 계속 라이브 스트리밍할 수 있는 데 유용합니다. 이 모드는 모드에서 1,000개의 프래그먼트 한도에 제한되지 않고 이전에 보관된 미디어를 스트리밍하는 데에도 유용합니다. ON_DEMAND

- **ON_DEMAND**: 이 유형의 세션의 경우 HLS 미디어 재생 목록에는 세션의 모든 프래그먼트가 지정된 수까지 포함됩니다. MaxMediaPlaylistFragmentResults 재생 목록은 각 세션마다 한 번만 검색해야 합니다. 미디어 플레이어에서 이러한 유형의 세션을 재생하면 일반적으로 사용자 인터페이스에는 표시할 재생 창의 위치를 선택할 수 있는 스크러버 컨트롤이 표시됩니다.

모든 재생 모드에서 동일한 시작 타임스탬프를 가진 프래그먼트가 여러 개 있는 경우 프래그먼트 번호가 가장 큰 프래그먼트 (즉, 최신 프래그먼트)가 HLS 미디어 재생 목록에 포함됩니다. FragmentSelectorType PRODUCER_TIMESTAMP 다른 프래그먼트는 포함되지 않습니다. 타임스탬프는 다르지만 지속 시간이 겹치는 프래그먼트는 여전히 HLS 미디어 재생 목록에 포함됩니다. 이로 인해 미디어 플레이어에서 예상치 못한 동작이 발생할 수 있습니다.

기본값은 LIVE입니다.

타입: 문자열

유효 값: LIVE | LIVE_REPLAY | ON_DEMAND

필수 여부: 아니요

StreamARN

HLS 마스터 플레이리스트 URL을 검색할 스트림의 Amazon 리소스 이름 (ARN).

또는 를 지정해야 합니다. StreamName StreamARN

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+

Required: No

StreamName

HLS 마스터 재생목록 URL을 검색할 스트림의 이름.

또는 를 지정해야 합니다. StreamName StreamARN

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256입니다.

패턴: [a-zA-Z0-9_.-]+

필수 여부: 아니요

응답 구문

```
HTTP/1.1 200
Content-type: application/json

{
  "HLSStreamingSessionURL": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

[HLSStreamingSessionURL](#)

미디어 플레이어가 HLS 마스터 플레이리스트를 검색하는 데 사용할 수 있는 URL (세션 토큰 포함).

타입: 문자열

Errors

모든 작업에 공통되는 오류에 대한 내용은 [일반적인 오류](#) 섹션을 참조하십시오.

ClientLimitExceededException

한도를 초과했기 때문에 Kinesis Video Streams가 요청 병목 현상을 일으켰습니다. 나중에 전화를 걸어 보세요. 제한에 대한 자세한 내용은 [Kinesis Video Streams](#) 제한을 참조하십시오.

HTTP 상태 코드: 400

InvalidArgumentException

지정된 파라미터는 제한을 초과하거나, 지원되지 않거나, 사용할 수 없습니다.

HTTP 상태 코드: 400

InvalidCodecPrivateDataException

비디오 스트림의 트랙 중 하나 이상에 있는 코덱 개인 데이터는 이 작업에 유효하지 않습니다.

HTTP 상태 코드: 400

MissingCodecPrivateDataException

비디오 스트림의 트랙 중 하나 이상에서 코덱 개인 데이터를 찾을 수 없습니다.

HTTP 상태 코드: 400

NoDataRetentionException

GetImages 데이터를 보관하지 않는 스트림 (즉, a가 0인 스트림)에 DataRetentionInHours 대해 요청되었습니다.

HTTP 상태 코드: 400

NotAuthorizedException

상태 코드: 403, 호출자가 지정된 스트림에서 작업을 수행할 권한이 없거나 토큰이 만료되었습니다.

HTTP 상태 코드: 401

ResourceNotFoundException

GetImages Kinesis Video Streams에서 지정한 스트림을 찾을 수 없는 경우 이 오류가 발생합니다.

GetHLSStreamingSessionURL 요청된 시간 범위 내에 LIVE_REPLAY 프래그먼트가 없는 스트림에 대해 a ON_DEMAND 또는 f가 있는 세션이 요청되거나 지난 30초 내에 LIVE 프래그먼트가 없는 스트림에 대해 a가 있는 세션이 요청되면 이 GetDASHStreamingSessionURL 오류가 발생합니다. PlaybackMode PlaybackMode

HTTP 상태 코드: 404

UnsupportedStreamMediaTypeException

재생 세션의 첫 번째 프래그먼트에 있는 트랙의 코덱 ID로는 미디어 유형 (예: h.264 또는 h.265 비디오 또는 AAC 또는 G.711 오디오)을 결정할 수 없습니다. 트랙 1의 코덱 ID는 다음과 같아야 하고, 선택적으로 트랙 2의 코덱 ID는 다음과 같아야 합니다. V_MPEG/ISO/AVC A_AAC

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

GetImages

서비스: Amazon Kinesis Video Streams Archived Media

지정된 시간 범위, 샘플링 간격 및 이미지 형식 구성의 각 타임스탬프에 해당하는 이미지 목록을 검색합니다.

Note

먼저 `GetDataEndpoint` API를 호출하여 엔드포인트를 가져와야 합니다. 그런 다음 `GetImages` [--endpoint-url](#) 파라미터를 사용하여 이 엔드포인트로 요청을 전송합니다.

[비디오 재생 트랙 요구 사항](#).

Request Syntax

```
POST /getImages HTTP/1.1
Content-type: application/json

{
  "EndTimeStamp": number,
  "Format": "string",
  "FormatConfig": {
    "string": "string"
  },
  "HeightPixels": number,
  "ImageSelectorType": "string",
  "MaxResults": number,
  "NextToken": "string",
  "SamplingInterval": number,
  "StartTimeStamp": number,
  "StreamARN": "string",
  "StreamName": "string",
  "WidthPixels": number
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

EndTimeStamp

생성할 이미지 범위의 종료 타임스탬프입니다. 시간 범위가 StartTimestamp 2초에서 300초 이상인 StartTimestamp 경우 시간 범위를 받게 됩니다. EndTimestamp IllegalArgumentException

유형: 타임스탬프

필수 여부: 예

Format

이미지를 인코딩하는 데 사용할 형식입니다.

타입: 문자열

유효 값: JPEG | PNG

필수 사항 여부: 예

FormatConfig

이미지 생성 시 적용할 수 있는 추가 매개 변수가 포함된 키-값 쌍 구조의 목록입니다.

FormatConfig키는 이미지를 생성하는 JPEGQuality 데 사용할 JPEG 품질 키를 나타내는입니다. FormatConfig값은 1에서 100 사이의 정수를 허용합니다. 값이 1이면 품질이 떨어지고 압축률이 가장 좋은 이미지가 생성됩니다. 값이 100인 경우 이미지는 최상의 품질로 압축률을 낮춘 상태로 생성됩니다. 값을 제공하지 않으면 JPEGQuality 키의 기본값은 80으로 설정됩니다.

유형: 문자열-문자열 맵

맵 항목: 최대 항목 수 1개입니다.

유효한 키: JPEGQuality

값 길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 256.

값 패턴: `^[a-zA-Z_0-9]+`

필수 여부: 아니요

HeightPixels

WidthPixels 매개변수와 함께 사용되는 출력 이미지의 높이입니다. HeightPixels 및 WidthPixels 매개변수를 모두 제공하면 이미지가 지정된 가로 세로 비율에 맞게 늘어납니다. HeightPixels 매개변수만 제공하면 원래纵横비를 사용하여 WidthPixels 비율을 계산합니다. 두 매개변수를 모두 제공하지 않으면 원본 이미지 크기가 반환됩니다.

타입: 정수

유효 범위: 최소값 1. 최대값은 2160입니다.

필수 여부: 아니요

ImageSelectorType

이미지 생성에 사용할 서버 또는 프로듀서 타임스탬프의 출처.

타입: 문자열

유효 값: PRODUCER_TIMESTAMP | SERVER_TIMESTAMP

필수 사항 여부: 예

MaxResults

API에서 반환되는 최대 이미지 수입니다.

Note

기본 제한은 API 응답당 25개 이미지입니다. 이 값보다 MaxResults 큰 값을 제공하면 페이지 크기가 25가 됩니다. 모든 추가 결과에는 페이지가 매겨집니다.

타입: Long

유효 범위: 최소값은 1입니다. 최대값 100.

필수 여부: 아니요

NextToken

다음 이미지 세트의 페이지 매김을 시작할 위치를 지정하는 토큰입니다. 이전에 잘린 GetImages:NextToken 응답에서 가져온 것입니다.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 4096자입니다.

패턴: `[a-zA-Z0-9+/\]{0,2}`

Required: No

SamplingInterval

스트림에서 이미지를 생성해야 하는 시간 간격 (밀리초) 입니다. 제공할 수 있는 최소값은 200ms (초당 5개 이미지) 입니다. 타임스탬프 범위가 샘플링 간격보다 작으면 가능한 경우 의 이미지가 `startTimestamp` 반환됩니다.

유형: 정수

필수 여부: 예

StartTimestamp

이미지를 생성해야 하는 시작점입니다. 이미지가 반환되는 타임스탬프의 전체 범위 내에 `StartTimestamp` 있어야 합니다.

유형: 타임스탬프

필수 여부: 예

StreamARN

이미지를 검색할 스트림의 Amazon 리소스 이름 (ARN). 또는 를 지정해야 합니다. `StreamName`
`StreamARN`

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_-.-]+/[0-9]+`

Required: No

StreamName

이미지를 검색할 스트림의 이름. `StreamName`또는 를 지정해야 합니다`StreamARN`.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256입니다.

패턴: [a-zA-Z0-9_.-]+

Required: No

WidthPixels

HeightPixels 매개변수와 함께 사용되는 출력 이미지의 너비입니다. WidthPixels 및 HeightPixels 매개 변수를 모두 제공하면 이미지가 지정된 가로 세로 비율에 맞게 늘어납니다. WidthPixels 매개변수만 제공하거나 매개변수만 ValidationException 제공하면 a가 발생합니다. HeightPixels 두 매개 변수를 모두 제공하지 않으면 스트림의 원본 이미지 크기가 반환됩니다.

타입: 정수

유효 범위: 최소값 1. 최대값은 3840입니다.

필수 여부: 아니요

응답 구문

```
HTTP/1.1 200
Content-type: application/json

{
  "Images": [
    {
      "Error": "string",
      "ImageContent": "string",
      "TimeStamp": number
    }
  ],
  "NextToken": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

Images

비디오 스트림에서 생성된 이미지 목록. 지정된 타임스탬프에 사용할 수 있는 미디어가 없는 경우 출력에 NO_MEDIA 오류가 표시됩니다. 이미지를 생성하는 동안 오류가 발생하는 경우 출력에 이미지 누락의 원인으로 MEDIA_ERROR 나열됩니다.

유형: [Image](#) 객체 어레이

NextToken

추가 이미지를 가져오기 위한 요청에 사용된 암호화된 토큰.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 4096자입니다.

패턴: `[a-zA-Z0-9+/-]{0,2}`

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 [일반적인 오류](#) 섹션을 참조하세요.

ClientLimitExceededException

한도를 초과했기 때문에 Kinesis Video Streams가 요청 병목 현상을 일으켰습니다. 나중에 전화를 걸어 보세요. 제한에 대한 자세한 내용은 [Kinesis Video Streams](#) 제한을 참조하십시오.

HTTP 상태 코드: 400

InvalidArgumentException

지정된 파라미터는 제한을 초과하거나, 지원되지 않거나, 사용할 수 없습니다.

HTTP 상태 코드: 400

NotAuthorizedException

상태 코드: 403, 호출자가 지정된 스트림에서 작업을 수행할 권한이 없거나 토큰이 만료되었습니다.

HTTP 상태 코드: 401

ResourceNotFoundException

GetImagesKinesis Video Streams에서 지정한 스트림을 찾을 수 없는 경우 이 오류가 발생합니다.

GetHLSStreamingSessionURL요청된 시간 범위 내에 LIVE_REPLAY 프래그먼트가 없는 스트림에 대해 a ON_DEMAND 또는 f가 있는 세션이 요청되거나 지난 30초 내에 LIVE 프래그먼트가 없는 스트림에 대해 a가 있는 세션이 요청되면 이 GetDASHStreamingSessionURL 오류가 발생합니다. PlaybackMode PlaybackMode

HTTP 상태 코드: 404

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS 파이썬용 SDK](#)
- [AWS 루비 V3용 SDK](#)

GetMediaForFragmentList

서비스: Amazon Kinesis Video Streams Archived Media

Amazon Kinesis 비디오 스트림에 보관된 데이터에서 프래그먼트 목록 (프래그먼트 번호로 지정) 의 미디어를 가져옵니다.

Note

먼저 `GetDataEndpoint` API를 호출하여 엔드포인트를 가져와야 합니다. 그런 다음 `GetMediaForFragmentList` [--endpoint-url](#) 파라미터를 사용하여 이 엔드포인트로 요청을 전송합니다.

제한에 대해서는 [Kinesis Video Streams](#) 제한을 참조하십시오.

Important

Kinesis Video Streams 아카이브 미디어 API를 호출한 후 오류가 발생하는 경우 HTTP 상태 코드 및 응답 본문 외에도 다음과 같은 정보가 포함됩니다.

- `x-amz-ErrorTypeHTTP` 헤더 — HTTP 상태 코드가 제공하는 내용 외에도 보다 구체적인 오류 유형을 포함합니다.
- `x-amz-RequestIdHTTP` 헤더 — 문제를 신고하려는 경우 지원 팀에서 요청 ID를 제공하면 문제를 더 잘 진단할 수 있습니다. AWS

HTTP 상태 코드와 `ErrorType` 헤더를 모두 사용하여 오류를 재시도할 수 있는지 여부와 조건에 대해 프로그래밍 방식으로 결정할 수 있으며, 클라이언트 프로그래머가 성공적으로 다시 시도하기 위해 취해야 할 조치에 대한 정보도 제공할 수 있습니다.

[자세한 내용은 이 항목 하단의 오류 섹션 및 일반적인 오류를 참조하십시오.](#)

Request Syntax

```
POST /getMediaForFragmentList HTTP/1.1
Content-type: application/json

{
  "Fragments": [ "string" ],
```

```

  "StreamARN": "string",
  "StreamName": "string"
}

```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

[Fragments](#)

미디어를 검색할 때 사용할 프래그먼트 수 목록입니다. 이 값은 를 사용하여 검색합니다.

[ListFragments](#)

유형: 문자열 어레이

배열 멤버: 최소 항목 수는 1입니다. 최대 항목 수는 1000입니다.

길이 제약: 최소 길이 1. 최대 길이 128.

패턴: `^[0-9]+$`

필수 사항 여부: Yes

[StreamARN](#)

프래그먼트 미디어를 검색할 스트림의 Amazon 리소스 이름 (ARN). 이 파라미터 또는 파라미터를 지정하십시오. StreamName

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

[StreamName](#)

프래그먼트 미디어를 검색할 스트림의 이름. 이 파라미터 또는 파라미터를 지정합니다.

StreamARN

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256입니다.

패턴: [a-zA-Z0-9_.-]+

필수 여부: 아니요

응답 구문

```
HTTP/1.1 200
Content-Type: ContentType
```

Payload

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

응답에 다음 HTTP 헤더가 반환됩니다.

ContentType

요청된 미디어의 콘텐츠 유형.

길이 제약: 최소 길이 1. 최대 길이 128.

패턴: ^[a-zA-Z0-9_\.\-]+\$

응답은 다음 내용을 HTTP 본문으로 반환합니다.

Payload

Kinesis Video Streams가 반환하는 페이로드에는 지정된 스트림의 청크 시퀀스입니다. 청크에 대한 자세한 내용은 [을 참조하십시오.](#) [PutMedia](#) Kinesis Video Streams가 호출에서 `GetMediaForFragmentList` 반환하는 청크에는 다음과 같은 추가 마트로스카 (MKV) 태그도 포함됩니다.

- `AWS_KINESISVIDEO_FRAGMENT_NUMBER` - 청크에 반환된 프래그먼트 번호입니다.
- `AWS_KINESISVIDEO_서버_SIDE_타임스탬프` - 프래그먼트의 서버 측 타임스탬프입니다.
- `AWS_KINESISVIDEO_프로듀서_사이드_타임스탬프` - 프래그먼트의 프로듀서 측 타임스탬프.

예외가 발생할 경우 다음 태그가 포함됩니다.

- `AWS_KINESISVIDEO_FRAGMENT_NUMBER` - 예외가 발생한 프래그먼트의 수입입니다.
- `AWS_KINESISVIDEO_EXCEPTION_ERROR_CODE` - 오류의 정수 코드입니다.
- `AWS_KINESISVIDEO_EXCEPTION_MESSAGE` - 예외에 대한 텍스트 설명입니다.

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 [일반적인 오류](#) 섹션을 참조하세요.

ClientLimitExceededException

한도를 초과했기 때문에 Kinesis Video Streams가 요청 병목 현상을 일으켰습니다. 나중에 전화를 걸어 보세요. 제한에 대한 자세한 내용은 [Kinesis Video Streams](#) 제한을 참조하십시오.

HTTP 상태 코드: 400

InvalidArgumentException

지정된 파라미터는 제한을 초과하거나, 지원되지 않거나, 사용할 수 없습니다.

HTTP 상태 코드: 400

NotAuthorizedException

상태 코드: 403, 호출자가 지정된 스트림에서 작업을 수행할 권한이 없거나 토큰이 만료되었습니다.

HTTP 상태 코드: 401

ResourceNotFoundException

GetImagesKinesis Video Streams에서 지정한 스트림을 찾을 수 없는 경우 이 오류가 발생합니다.

GetHLSStreamingSessionURL요청된 시간 범위 내에 LIVE_REPLAY 프래그먼트가 없는 스트림에 대해 a ON_DEMAND 또는 f가 있는 세션이 요청되거나 지난 30초 내에 LIVE 프래그먼트가 없는 스트림에 대해 a가 있는 세션이 요청되면 이 GetDASHStreamingSessionURL 오류가 발생합니다. PlaybackMode PlaybackMode

HTTP 상태 코드: 404

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

ListFragments

서비스: Amazon Kinesis Video Streams Archived Media

아카이브된 데이터 내에서 지정된 스트림 및 타임스탬프 범위의 [Fragment](#) 객체 목록을 반환합니다.

리스팅 프래그먼트는 결국 일관성을 유지합니다. 즉, 제작자가 프래그먼트가 지속된다는 승인을 받더라도 요청에서 결과가 즉시 반환되지 않을 수 있습니다. ListFragments 하지만 일반적으로 1초 이내에 결과를 확인할 수 있습니다.

Note

먼저 GetDataEndpoint API를 호출하여 엔드포인트를 가져와야 합니다. 그런 다음 ListFragments [--endpoint-url](#) 파라미터를 사용하여 이 엔드포인트로 요청을 전송합니다.

Important

Kinesis Video Streams 아카이브 미디어 API를 호출한 후 오류가 발생하는 경우 HTTP 상태 코드 및 응답 본문 외에도 다음과 같은 정보가 포함됩니다.

- `x-amz-ErrorTypeHTTP` 헤더 — HTTP 상태 코드가 제공하는 내용 외에도 보다 구체적인 오류 유형을 포함합니다.
- `x-amz-RequestIdHTTP` 헤더 — 문제를 신고하려는 경우 지원 팀에서 요청 ID를 제공하면 문제를 더 잘 진단할 수 있습니다. AWS

HTTP 상태 코드와 ErrorType 헤더를 모두 사용하여 오류를 재시도할 수 있는지 여부와 조건에 대해 프로그래밍 방식으로 결정할 수 있으며, 클라이언트 프로그래머가 성공적으로 다시 시도하기 위해 취해야 할 조치에 대한 정보도 제공할 수 있습니다.

[자세한 내용은 이 항목 하단의 오류 섹션 및 일반적인 오류를 참조하십시오.](#)

Request Syntax

```
POST /listFragments HTTP/1.1
Content-type: application/json

{
  "FragmentSelector": {
```

```

    "FragmentSelectorType": "string",
    "TimestampRange": {
      "EndTimeStamp": number,
      "StartTimeStamp": number
    }
  },
  "MaxResults": number,
  "NextToken": "string",
  "StreamARN": "string",
  "StreamName": "string"
}

```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

FragmentSelector

반환할 프래그먼트 범위의 타임스탬프 범위와 타임스탬프 출처를 설명합니다.

Note

이는 API에 NextToken 전달되지 않은 경우에만 필요합니다.

유형: FragmentSelector 객체

필수 항목 여부: 아니요

MaxResults

반환할 총 프래그먼트 수입니다. 사용 가능한 총 조각 수가 에 지정된 값보다 많으면 페이지 매김을 재개하는 데 사용할 수 있는 ListFragmentsNextToken:가 출력에 제공됩니다. max-results

기본 값은 100입니다.

타입: Long

유효한 범위: 최소값은 1입니다. 최대값은 1000입니다.

필수 여부: 아니요

NextToken

페이지 매김을 시작할 위치를 지정하기 위한 토큰입니다. 다음은 이전에 잘린 [ListFragmentsNextToken](#) 응답의 [:입나다](#).

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 4096자입니다.

패턴: `[a-zA-Z0-9+/\+=]{0,2}`

Required: No

StreamARN

프래그먼트 목록을 검색할 스트림의 Amazon 리소스 이름 (ARN). 이 파라미터 또는 파라미터를 지정하십시오. `StreamName`

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

StreamName

프래그먼트 목록을 검색할 스트림의 이름. 이 파라미터 또는 파라미터를 지정합니다. `StreamARN`

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256입니다.

패턴: `[a-zA-Z0-9_.-]+`

필수 여부: 아니요

응답 구문

```
HTTP/1.1 200
Content-type: application/json
```

```
{
  "Segments": [
    {
      "SegmentLengthInMilliseconds": number,
      "SegmentNumber": "string",
      "SegmentSizeInBytes": number,
      "ProducerTimestamp": number,
      "ServerTimestamp": number
    }
  ],
  "NextToken": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

Segments

선택 기준을 충족하는 스트림의 아카이브된 Segment 객체 목록. 결과는 특정 순서가 없으며, 여러 페이지에 걸쳐 표시되더라도 마찬가지입니다.

스트림에 선택 기준을 충족하는 프래그먼트가 없는 경우 빈 목록이 반환됩니다.

유형: Segment 객체 어레이

NextToken

반환된 목록이 잘린 경우 작업은 이 토큰을 반환하여 다음 결과 페이지를 검색하는 데 사용합니다. 이 값은 null 반환할 결과가 더 이상 없는 경우입니다.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 4096자입니다.

패턴: [a-zA-Z0-9+/-]{0,2}

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

ClientLimitExceededException

한도를 초과했기 때문에 Kinesis Video Streams가 요청 병목 현상을 일으켰습니다. 나중에 전화를 걸어 보세요. 제한에 대한 자세한 내용은 [Kinesis Video Streams](#) 제한을 참조하십시오.

HTTP 상태 코드: 400

InvalidArgumentException

지정된 파라미터는 제한을 초과하거나, 지원되지 않거나, 사용할 수 없습니다.

HTTP 상태 코드: 400

NotAuthorizedException

상태 코드: 403, 호출자가 지정된 스트림에서 작업을 수행할 권한이 없거나 토큰이 만료되었습니다.

HTTP 상태 코드: 401

ResourceNotFoundException

GetImagesKinesis Video Streams에서 지정한 스트림을 찾을 수 없는 경우 이 오류가 발생합니다.

GetHLSStreamingSessionURL요청된 시간 범위 내에 LIVE_REPLAY 프래그먼트가 없는 스트림에 대해 a ON_DEMAND 또는 f가 있는 세션이 요청되거나 지난 30초 내에 LIVE 프래그먼트가 없는 스트림에 대해 a가 있는 세션이 요청되면 이 GetDASHStreamingSessionURL 오류가 발생합니다. PlaybackMode PlaybackMode

HTTP 상태 코드: 404

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)

- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

Amazon Kinesis Video Kinesis Video Kin

다음 작업이 지원됩니다.

- [GetIceServerConfig](#)
- [SendAlexaOfferToMaster](#)

GetIceServerConfig

서비스: Amazon Kinesis Video Signaling Channels

참고: 이 API를 사용하기 전에 API를 호출하여 HTTPS 엔드포인트를 요청해야 합니다.

GetSignalingChannelEndpoint 그런 다음 GetIceServerConfig API 요청에서 엔드포인트와 리전을 지정합니다.

WebRTC 연결을 구성하는 데 사용할 수 있는 URI, 사용자 이름 및 암호를 포함한 ICE (대화형 연결 설정) 서버 구성 정보를 가져옵니다. ICE 구성 요소는 이 구성 정보를 사용하여 NAT (TURN) 릴레이 주변의 Traversal Using Relays를 통한 인증을 포함하여 WebRTC 연결을 설정합니다.

TURN은 애플리케이션의 연결성을 개선하는 데 사용되는 프로토콜입니다. peer-to-peer TURN은 클라우드 기반 릴레이 서비스를 제공함으로써 한 명 이상의 피어가 직접 연결을 할 수 없는 경우에도 연결을 설정할 수 있도록 합니다. peer-to-peer 자세한 내용은 [TURN 서비스 액세스를 위한 REST API를 참조하십시오](#).

피어 중 한 명이 신호 채널을 통해 직접 peer-to-peer 연결을 설정할 수 없는 경우 이 API를 호출하여 대체 메커니즘을 설정할 수 있습니다. 이 API를 호출하려면 신호 채널의 Amazon 리소스 이름 (ARN) 을 지정해야 합니다.

Request Syntax

```
POST /v1/get-ice-server-config HTTP/1.1
Content-type: application/json
```

```
{
  "ChannelARN": "string",
  "ClientId": "string",
  "Service": "string",
  "Username": "string"
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

ChannelARN

구성된 피어 간 peer-to-peer 연결에 사용할 신호 채널의 ARN입니다.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

필수 사항 여부: Yes

ClientId

시청자의 고유 식별자. 신호 채널 내에서 고유해야 합니다.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256입니다.

패턴: `[a-zA-Z0-9_.-]+`

Required: No

Service

원하는 서비스를 지정합니다. 현재는 TURN 이 (가) 유일한 유효한 값입니다.

타입: 문자열

유효 값: TURN

필수 여부: 아니요

Username

자격 증명에 연결할 선택적 사용자 ID.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256입니다.

패턴: `[a-zA-Z0-9_.-]+`

필수 여부: 아니요

응답 구문

```
HTTP/1.1 200
Content-type: application/json

{
  "IceServerList": [
    {
      "Password": "string",
      "Ttl": number,
      "Uris": [ "string" ],
      "Username": "string"
    }
  ]
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

IceServerList

ICE 서버 정보 개체 목록.

타입: [IceServer](#) 객체 배열

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 [일반적인 오류](#) 섹션을 참조하세요.

ClientLimitExceededException

허용된 클라이언트 호출 한도를 초과했기 때문에 요청이 병목 현상이 발생했습니다. 나중에 전화를 걸어 보세요.

HTTP 상태 코드: 400

InvalidArgumentException

이 입력 매개변수의 값이 유효하지 않습니다.

HTTP 상태 코드: 400

InvalidClientException

지정된 클라이언트가 유효하지 않습니다.

HTTP 상태 코드: 400

NotAuthorizedException

발신자는 이 작업을 수행할 권한이 없습니다.

HTTP 상태 코드: 401

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 404

SessionExpiredException

클라이언트 세션이 만료된 경우 클라이언트가 연결되면 세션은 45분 동안 유효합니다. 메시지를 계속 보내고 받으려면 클라이언트가 채널에 다시 연결해야 합니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS 파이썬용 SDK](#)
- [AWS 루비 V3용 SDK](#)

SendAlexaOfferToMaster

서비스: Amazon Kinesis Video Signaling Channels

Note

이 API를 사용하기 전에 API를 호출하여 엔드포인트를 가져와야 합니다.

GetSignalingChannelEndpoint 그런 다음 SendAlexaOfferToMaster API 요청에서 엔드포인트와 리전을 지정합니다.

이 API를 사용하면 WebRTC 지원 디바이스를 Alexa 디스플레이 디바이스와 연결할 수 있습니다. 호출되면 Alexa SDP (세션 설명 프로토콜) 오퍼가 마스터 피어에 전송됩니다. 마스터가 지정된 신호 채널에 연결되는 즉시 오퍼가 전달됩니다. 이 API는 연결된 마스터로부터 SDP 응답을 반환합니다. 마스터가 시그널링 채널에 연결되지 않은 경우 메시지가 만료될 때까지 재배송 요청이 이루어집니다.

Request Syntax

```
POST /v1/send-alex-a-offer-to-master HTTP/1.1
```

```
Content-type: application/json
```

```
{
  "ChannelARN": "string",
  "MessagePayload": "string",
  "SenderClientId": "string"
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

ChannelARN

Alexa와 마스터 피어가 통신하는 데 사용하는 신호 채널의 Amazon 리소스 이름 (ARN).

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

필수 사항 여부: Yes

MessagePayload

base64로 인코딩된 SDP는 콘텐츠를 제공합니다.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 10,000개입니다.

패턴: `[a-zA-Z0-9+/=]+`

필수 사항 여부: Yes

SenderId

발신자 클라이언트의 고유 식별자입니다.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256입니다.

패턴: `[a-zA-Z0-9_.-]+`

필수 항목 여부: 예

응답 구문

```
HTTP/1.1 200
Content-type: application/json

{
  "Answer": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반환합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

Answer

base64로 인코딩된 SDP 응답 콘텐츠.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 10,000개입니다.

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 [일반적인 오류](#) 섹션을 참조하세요.

ClientLimitExceededException

허용된 클라이언트 통화 한도를 초과했기 때문에 요청이 병목 현상이 발생했습니다. 나중에 전화를 걸어 보세요.

HTTP 상태 코드: 400

InvalidArgumentException

이 입력 매개변수의 값이 유효하지 않습니다.

HTTP 상태 코드: 400

NotAuthorizedException

호출자는 이 작업을 수행할 권한이 없습니다.

HTTP 상태 코드: 401

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 404

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)

- [AWS SDK for C++](#)
- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS 파이썬용 SDK](#)
- [AWS 루비 V3용 SDK](#)

Amazon Kinesis Video Streams

다음 작업이 Amazon Kinesis Video Streams

- [JoinStorageSession](#)

JoinStorageSession

서비스: Amazon Kinesis Video WebRTC Storage

Note

이 API를 사용하기 전에 API를 호출하여 WEBRTC 엔드포인트를 요청해야 합니다. GetSignalingChannelEndpoint 그런 다음 JoinStorageSession API 요청에서 엔드포인트와 지역을 지정합니다.

진행 중인 단방향 비디오 및/또는 다방향 오디오 WebRTC 세션에 입력 채널용 비디오 제작 장치로 참여하십시오. 채널에 대한 기존 세션이 없는 경우 새 스트리밍 세션을 만들고 신호 채널의 Amazon 리소스 이름 (ARN) 을 제공해야 합니다.

현재 이 SINGLE_MASTER 유형의 비디오 제작 디바이스는 오디오와 비디오 미디어를 스트림으로 인제스트할 수 있습니다. 비디오 제작 장치만 세션에 참여하고 미디어를 녹화할 수 있습니다.

Important

현재 WebRTC 통합에는 오디오 및 비디오 트랙이 모두 필요합니다.
현재 요구 사항:

- 비디오 트랙: H.264
- 오디오 트랙: 오픈스

Kinesis 비디오 스트림의 결과 인제스트된 비디오는 H.264 비디오 및 AAC 오디오와 같은 매개 변수를 가집니다.

마스터 참가자가 WebRTC를 통해 연결을 협상하면 수집된 미디어 세션이 Kinesis 비디오 스트림에 저장됩니다. 그러면 여러 시청자가 Playback API를 통해 실시간 미디어를 재생할 수 있습니다.

또한 재생, 이미지 생성 HLS 등과 같은 DASH 기존 Kinesis Video Streams 기능을 인제스트된 [GetImagesWebRTC](#) 미디어와 함께 사용할 수 있습니다.

Note

S3 이미지 전송 및 알림은 현재 지원되지 않습니다.

Note

채널의 세션에 비디오 제작 디바이스 클라이언트를 하나만 연결할 수 있다고 가정해 보겠습니다. 한 명 이상의 클라이언트가 특정 채널의 세션에 동영상 제작 장치로 참여하는 경우 가장 최근의 클라이언트 요청이 우선합니다.

추가 정보

- 무력화 - 이 API는 무력하지 않습니다.
- 재시도 동작 - 이는 새 API 호출로 간주됩니다.
- 동시 통화 - 동시 통화가 허용됩니다. 제안은 각 호출당 한 번씩 전송됩니다.

Request Syntax

```
POST /joinStorageSession HTTP/1.1
Content-type: application/json
```

```
{
  "channelArn": "string"
}
```

URI 요청 파라미터

요청은 URI 파라미터를 사용하지 않습니다.

요청 본문

요청은 JSON 형식으로 다음 데이터를 받습니다.

channelArn

신호 채널의 Amazon 리소스 이름(ARN)입니다.

유형: String

패턴: `^arn:(aws[a-zA-Z-]*):kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+$`

필수 항목 여부: 예

응답 구문

```
HTTP/1.1 200
```

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 반환합니다.

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 [일반적인 오류](#) 섹션을 참조하세요.

AccessDeniedException

이 작업을 수행하는 데 필요한 권한이 없습니다.

HTTP 상태 코드: 403

ClientLimitExceededException

허용된 클라이언트 호출 한도를 초과했기 때문에 Kinesis Video Streams가 요청 병목 현상을 일으켰습니다. 나중에 전화를 걸어 보세요.

HTTP 상태 코드: 400

InvalidArgumentException

이 입력 매개변수의 값이 유효하지 않습니다.

HTTP 상태 코드: 400

ResourceNotFoundException

지정된 리소스를 찾을 수 없습니다.

HTTP 상태 코드: 404

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)

- [AWS Go v2를 위한 SDK](#)
- [AWS Java V2용 SDK](#)
- [AWS V3용 SDK JavaScript](#)
- [AWS PHP V3용 SDK](#)
- [AWS Python용 SDK](#)
- [AWS 루비 V3용 SDK](#)

데이터 유형

Amazon Kinesis 비디오 스트림은 다음과 같은 데이터 유형을 지원합니다.

- [ChannelInfo](#)
- [ChannelNameCondition](#)
- [DeletionConfig](#)
- [EdgeAgentStatus](#)
- [EdgeConfig](#)
- [ImageGenerationConfiguration](#)
- [ImageGenerationDestinationConfig](#)
- [LastRecorderStatus](#)
- [LastUploaderStatus](#)
- [ListEdgeAgentConfigurationsEdgeConfig](#)
- [LocalSizeConfig](#)
- [MappedResourceConfigurationListItem](#)
- [MediaSourceConfig](#)
- [MediaStorageConfiguration](#)
- [NotificationConfiguration](#)
- [NotificationDestinationConfig](#)
- [RecorderConfig](#)
- [ResourceEndpointListItem](#)
- [ScheduleConfig](#)
- [SingleMasterChannelEndpointConfiguration](#)

- [SingleMasterConfiguration](#)
- [StreamInfo](#)
- [StreamNameCondition](#)
- [Tag](#)
- [UploaderConfig](#)

Amazon Kinesis 비디오 스트림 미디어는 다음 데이터 유형을 지원합니다.

- [StartSelector](#)

Amazon Kinesis 비디오 스트림 아카이브 미디어는 다음 데이터 유형을 지원합니다.

- [ClipFragmentSelector](#)
- [ClipTimestampRange](#)
- [DASHFragmentSelector](#)
- [DASHTimestampRange](#)
- [Fragment](#)
- [FragmentSelector](#)
- [HLSFragmentSelector](#)
- [HLSTimestampRange](#)
- [Image](#)
- [TimestampRange](#)

Amazon Kinesis 비디오 시그널링 채널은 다음과 같은 데이터 유형을 지원합니다.

- [IceServer](#)

Amazon Kinesis Video WebRTC 스토리지는 다음과 같은 데이터 유형을 지원합니다.

Amazon Kinesis Video Streams

Amazon Kinesis 비디오 스트림은 다음과 같은 데이터 유형을 지원합니다.

- [ChannellInfo](#)

- [ChannelNameCondition](#)
- [DeletionConfig](#)
- [EdgeAgentStatus](#)
- [EdgeConfig](#)
- [ImageGenerationConfiguration](#)
- [ImageGenerationDestinationConfig](#)
- [LastRecorderStatus](#)
- [LastUploaderStatus](#)
- [ListEdgeAgentConfigurationsEdgeConfig](#)
- [LocalSizeConfig](#)
- [MappedResourceConfigurationListItem](#)
- [MediaSourceConfig](#)
- [MediaStorageConfiguration](#)
- [NotificationConfiguration](#)
- [NotificationDestinationConfig](#)
- [RecorderConfig](#)
- [ResourceEndpointListItem](#)
- [ScheduleConfig](#)
- [SingleMasterChannelEndpointConfiguration](#)
- [SingleMasterConfiguration](#)
- [StreamInfo](#)
- [StreamNameCondition](#)
- [Tag](#)
- [UploaderConfig](#)

ChannelInfo

서비스: Amazon Kinesis Video Streams

시그널링 채널의 메타데이터와 속성을 캡슐화하는 구조입니다.

내용

ChannelARN

신호 채널의 Amazon 리소스 이름(ARN)입니다.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

ChannelName

시그널링 채널의 이름.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256입니다.

패턴: `[a-zA-Z0-9_.-]+`

Required: No

ChannelStatus

신호 채널의 현재 상태.

타입: 문자열

유효 값: CREATING | ACTIVE | UPDATING | DELETING

필수 여부: 아니요

ChannelType

신호 채널의 유형.

타입: 문자열

유효 값: SINGLE_MASTER | FULL_MESH

필수 여부: 아니요

CreationTime

시그널링 채널이 생성된 시간.

유형: 타임스탬프

필수 여부: 아니요

SingleMasterConfiguration

SINGLE_MASTER채널 유형에 대한 컨피그레이션이 포함된 구조입니다.

유형: [SingleMasterConfiguration](#)객체

필수 항목 여부: 아니요

Version

신호 채널의 현재 버전입니다.

유형: 문자열

길이 제한: 최소 길이는 1. 최대 길이는 64.

패턴: [a-zA-Z0-9]+

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

ChannelNameCondition

서비스: Amazon Kinesis Video Streams

ListSignalingChannelsAPI의 선택적 입력 파라미터입니다. ListSignalingChannels호출하는 동안 이 파라미터를 지정하면 API는 에서 ChannelNameCondition 지정한 조건을 충족하는 채널만 반환합니다.

내용

ComparisonOperator

비교 연산자. 현재는 이름이 지정된 접두사로 시작하는 신호 채널을 찾는 BEGINS_WITH 연산자만 지정할 수 있습니다.

타입: 문자열

유효 값: BEGINS_WITH

필수 여부: 아니요

ComparisonValue

비교할 값입니다.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256입니다.

패턴: [a-zA-Z0-9_.-]+

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DeletionConfig

서비스: Amazon Kinesis Video Streams

Edge Agent에서 스트림의 연결을 삭제하는 데 필요한 구성 세부 정보입니다.

내용

DeleteAfterUpload

Kinesis Video Stream 클라우드에 업로드된 미디어를 삭제하도록 표시할지 여부를 나타내는 데 사용되는 boolean 값입니다. 삭제 구성 값 중 하나라도 로 설정된 경우 (예: 또는 의 한도에 도달한 경우) 미디어 파일을 삭제할 수 있습니다. true EdgeRetentionInHours MaxLocalMediaSizeInMB

기본값이 로 true 설정되어 있으므로 미디어 파일이 클라우드에 처음 업로드되기 전에 삭제되지 않도록 업로더 일정을 구성하십시오 AWS .

타입: 부울

필수 항목 여부: 아니요

EdgeRetentionInHours

Edge Agent의 스트림에 데이터를 보존하려는 시간. 보존 시간의 기본값은 720시간이며, 이는 30일로 변환됩니다.

타입: 정수

유효 범위: 최소값 1. 최대값은 720입니다.

필수 여부: 아니요

LocalSizeConfig

에지 구성을 삭제하는 데 필요한 로컬 크기 값입니다.

유형: [LocalSizeConfig](#)객체

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

EdgeAgentStatus

서비스: Amazon Kinesis Video Streams

Edge Agent의 레코더 및 업로더 작업에 대한 최신 상태 세부 정보가 포함된 개체입니다. 이 정보를 사용하여 Edge Agent의 현재 상태를 확인할 수 있습니다.

내용

LastRecorderStatus

스트림 에지 레코딩 작업의 최신 상태.

유형: [LastRecorderStatus](#) 객체

필수 항목 여부: 아니요

LastUploaderStatus

스트림의 엣지 투 클라우드 업로더 작업의 최신 상태.

유형: [LastUploaderStatus](#) 객체

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

EdgeConfig

서비스: Amazon Kinesis Video Streams

Edge Agent IoT Greengrass 구성 요소와 동기화하는 데 사용할 스트림의 에지 구성에 대한 설명입니다. Edge Agent 구성 요소는 온프레미스의 IoT Hub 장치 설정에서 실행됩니다.

내용

HubDeviceArn

스트림의 “사물 인터넷 (IoT) 사물” Arn.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: `arn:[a-z\d-]+:iot:[a-z0-9-]+:[0-9]+:thing/[a-zA-Z0-9_.-]+`

필수 사항 여부: Yes

RecorderConfig

레코더 구성은 카메라에서 스트리밍되는 로컬 미디어 파일에 액세스하기 위한 자격 증명으로 사용되는 로컬 `MediaSourceConfig` 세부 정보로 구성됩니다.

유형: [RecorderConfig](#) 객체

필수 여부: 예

DeletionConfig

삭제 구성은 삭제에 사용된 보존 시간 (`EdgeRetentionInHours`) 과 로컬 크기 구성 (`LocalSizeConfig`) 세부 정보로 구성됩니다.

유형: [DeletionConfig](#) 객체

필수 항목 여부: 아니요

UploaderConfig

업로더 구성에는 녹화된 미디어 파일의 업로드 작업을 Edge Agent에서 Kinesis Video Stream으로 예약하는 데 사용되는 `ScheduleExpression` 세부 정보가 포함되어 있습니다.

유형: [UploaderConfig](#) 객체

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

ImageGenerationConfiguration

서비스: Amazon Kinesis Video Streams

KVS 이미지 전송에 필요한 정보가 들어 있는 구조입니다. null인 경우 스트림에서 구성이 삭제됩니다.

내용

DestinationConfig

고객에게 이미지를 전달하는 데 필요한 정보가 포함된 구조입니다.

유형: [ImageGenerationDestinationConfig](#) 객체

필수 여부: 예

Format

허용되는 이미지 형식.

타입: 문자열

유효 값: JPEG | PNG

필수 사항 여부: 예

ImageSelectorType

이미지 생성에 사용할 서버 또는 프로듀서 타임스탬프의 출처.

타입: 문자열

유효 값: SERVER_TIMESTAMP | PRODUCER_TIMESTAMP

필수 사항 여부: 예

SamplingInterval

스트림에서 이미지를 생성해야 하는 시간 간격 (밀리초) 입니다. 제공할 수 있는 최소값은 200ms입니다. 타임스탬프 범위가 샘플링 간격보다 작으면 가능한 경우 의 이미지가 StartTimestamp 반환됩니다.

유형: 정수

유효 범위: 최소값 3000입니다. 최대값은 20000입니다.

필수 여부: 예

Status

ContinuousImageGenerationConfigurationsAPI의 활성화 또는 비활성화 여부를 나타냅니다.

타입: 문자열

유효 값: ENABLED | DISABLED

필수 사항 여부: 예

FormatConfig

이미지 생성 시 적용할 수 있는 추가 파라미터가 포함된 키-값 쌍 구조 목록입니다.

FormatConfig키는 이미지를 생성하는 JPEGQuality 데 사용할 JPEG 품질 키를 나타내는입니다. FormatConfig값은 1에서 100 사이의 정수를 허용합니다. 값이 1이면 품질이 떨어지고 압축률이 가장 좋은 이미지가 생성됩니다. 값이 100인 경우 이미지는 최상의 품질로 압축률을 낮춘 상태로 생성됩니다. 값을 제공하지 않으면 JPEGQuality 키의 기본값은 80으로 설정됩니다.

유형: 문자열-문자열 맵

맵 항목: 최대 항목 수 1개입니다.

유효한 키: JPEGQuality

값 길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 256.

값 패턴: `^[a-zA-Z_0-9]+`

필수 여부: 아니요

HeightPixels

WidthPixels매개변수와 함께 사용되는 출력 이미지의 높이입니다. HeightPixels및 WidthPixels 매개 변수를 모두 제공하면 이미지가 지정된 가로 세로 비율에 맞게 늘어납니다. HeightPixels매개변수만 제공하면 원래 종횡비를 사용하여 WidthPixels 비율을 계산합니다. 두 매개 변수를 모두 제공하지 않으면 원본 이미지 크기가 반환됩니다.

타입: 정수

유효 범위: 최소값 1. 최대값은 2160입니다.

필수 여부: 아니요

WidthPixels

HeightPixels 매개변수와 함께 사용되는 출력 이미지의 너비입니다. WidthPixels 및 HeightPixels 매개 변수를 모두 제공하면 이미지가 지정된 가로 세로 비율에 맞게 늘어납니다. WidthPixels 매개변수만 제공하면 원래纵横비를 사용하여 HeightPixels 비율을 계산합니다. 두 매개 변수를 모두 제공하지 않으면 원본 이미지 크기가 반환됩니다.

타입: 정수

유효 범위: 최소값 1. 최대값은 3840입니다.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

ImageGenerationDestinationConfig

서비스: Amazon Kinesis Video Streams

고객에게 이미지를 제공하는 데 필요한 정보가 들어 있는 구조입니다.

내용

DestinationRegion

이미지가 전송될 S3 버킷의 AWS 지역. 스트림이 위치한 지역과 DestinationRegion 일치해야 합니다.

타입: 문자열

길이 제한: 최소 길이 9. 최대 길이는 14입니다.

패턴: `^[a-z]+(-[a-z]+)?-[a-z]+-[0-9]$`

필수 사항 여부: Yes

Uri

이미지가 전송될 위치를 식별하는 통합 리소스 식별자 (URI).

유형: 문자열

길이 제약: 최소 길이는 1. 최대 길이는 255.

패턴: `^[a-zA-Z_0-9]+:(//)?(^[^/]+)/?([^*]*)$`

필수 여부: 예

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

LastRecorderStatus

서비스: Amazon Kinesis Video Streams

스트림의 엣지 레코딩 작업의 최신 상태입니다.

내용

JobStatusDetails

레코더 작업의 최신 상태에 대한 설명.

타입: 문자열

필수사항: 아니요

LastCollectedTime

레코더 작업이 마지막으로 실행되고 미디어가 로컬 디스크에 저장된 타임스탬프입니다.

유형: 타임스탬프

필수 여부: 아니요

LastUpdatedTime

레코더 상태가 마지막으로 업데이트된 타임스탬프입니다.

유형: 타임스탬프

필수 여부: 아니요

RecorderStatus

최신 레코더 작업 상태.

타입: 문자열

유효 값: SUCCESS | USER_ERROR | SYSTEM_ERROR

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

LastUploaderStatus

서비스: Amazon Kinesis Video Streams

스트림의 엣지 투 클라우드 업로더 작업의 최신 상태

내용

JobStatusDetails

업로더 작업의 최신 상태에 대한 설명.

타입: 문자열

필수사항: 아니요

LastCollectedTime

업로더 작업이 마지막으로 실행되고 미디어가 클라우드에 수집된 타임스탬프입니다.

유형: 타임스탬프

필수 여부: 아니요

LastUpdatedTime

업로더 상태가 마지막으로 업데이트된 시점의 타임스탬프입니다.

유형: 타임스탬프

필수 여부: 아니요

UploaderStatus

최신 업로더 작업의 상태.

타입: 문자열

유효 값: SUCCESS | USER_ERROR | SYSTEM_ERROR

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

ListEdgeAgentConfigurationsEdgeConfig

서비스: Amazon Kinesis Video Streams

단일 스트림의 에지 구성에 대한 설명입니다.

내용

CreationTime

스트림이 엣지 구성을 처음 생성한 시점의 타임스탬프입니다.

유형: 타임스탬프

필수 여부: 아니요

EdgeConfig

Edge Agent IoT Greengrass 구성 요소와 동기화하는 데 사용할 스트림의 에지 구성에 대한 설명입니다. Edge Agent 구성 요소는 온프레미스의 IoT Hub 장치 설정에서 실행됩니다.

유형: [EdgeConfig](#) 객체

필수 항목 여부: 아니요

FailedStatusDetails

생성된 장애 상태에 대한 설명.

타입: 문자열

필수 사항: 아니요

LastUpdatedTime

스트림이 엣지 구성을 마지막으로 업데이트한 시점의 타임스탬프입니다.

유형: 타임스탬프

필수 여부: 아니요

StreamARN

스트림의 Amazon 리소스 이름(ARN)입니다.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

StreamName

스트림의 이름입니다.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256입니다.

패턴: `[a-zA-Z0-9_.-]+`

Required: No

SyncStatus

스트림 에지 구성의 현재 동기화 상태.

타입: 문자열

유효 값: SYNCING | ACKNOWLEDGED | IN_SYNC | SYNC_FAILED | DELETING | DELETE_FAILED | DELETING_ACKNOWLEDGED

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

LocalSizeConfig

서비스: Amazon Kinesis Video Streams

Edge Agent에서 스트림에 저장하려는 미디어의 최대 크기 (MaxLocalMediaSizeInMB) 와 스트림의 최대 크기에 도달했을 때 사용해야 하는 전략 (StrategyOnFullSize) 을 포함하는 구성 세부 정보입니다.

내용

MaxLocalMediaSizeInMB

Edge Agent에서 스트림용으로 저장하려는 미디어의 전체 최대 크기.

유형: 정수

유효 범위: 최소값은 64입니다. 최대값은 200,000달러입니다.

필수 여부: 아니요

StrategyOnFullSize

스트림 MaxLocalMediaSizeInMB 한도에 도달했을 때 수행하는 전략.

타입: 문자열

유효 값: DELETE_OLDEST_MEDIA | DENY_NEW_MEDIA

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

MappedResourceConfigurationListItem

서비스: Amazon Kinesis Video Streams

미디어 스토리지 구성 속성을 캡슐화하거나 포함하는 구조입니다.

내용

ARN

스트림과 연결된 Kinesis 비디오 스트림 리소스의 Amazon 리소스 이름 (ARN).

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+

Required: No

Type

kinesis 비디오 스트림에 대한 관련 리소스의 유형.

타입: 문자열

필수 항목 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

MediaSourceConfig

서비스: Amazon Kinesis Video Streams

카메라로 스트리밍되는 미디어 파일에 액세스하는 데 필요한 자격 증명 (MediaUriSecretArn 및 MediaUriType) 으로 구성된 구성 세부 정보입니다.

내용

MediaUriSecretArn

AWS Secrets Manager ARN에서 카메라의 사용자 이름과 암호 또는 로컬 미디어 파일 위치를 확인할 수 있습니다.

타입: 문자열

길이 제약: 최소 길이는 20. 최대 길이는 2,048.

패턴: `arn:[a-z\d-]+:secretsmanager:[a-z0-9-]+:[0-9]+:secret:[a-zA-Z0-9_.-]+`

필수 사항 여부: Yes

MediaUriType

통합 리소스 식별자 (URI) 유형. 이 FILE_URI 값은 로컬 미디어 파일을 스트리밍하는 데 사용할 수 있습니다.

Note

미리보기는 RTSP_URI 미디어 소스 URI 형식만 지원합니다.

타입: 문자열

유효 값: RTSP_URI | FILE_URI

필수 여부: 예

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

MediaStorageConfiguration

서비스: Amazon Kinesis Video Streams

미디어 스토리지 구성 속성을 캡슐화하거나 포함하는 구조입니다.

- 를 활성화하면 StorageStatus 데이터가 제공된 위치에 저장됩니다. StreamARN WebRTC 통합이 작동하려면 스트림에 데이터 보존이 활성화되어 있어야 합니다.
- 를 비활성화하면 StorageStatus 데이터가 저장되지 않으므로 StreamARN 파라미터가 필요하지 않습니다.

내용

Status

미디어 스토리지 구성 상태.

타입: 문자열

유효 값: ENABLED | DISABLED

필수 사항 여부: 예

StreamARN

스트림의 Amazon 리소스 이름(ARN)입니다.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)

- [AWS 루비 V3용 SDK](#)

NotificationConfiguration

서비스: Amazon Kinesis Video Streams

이 API를 사용하여 스트림에서 프래그먼트를 사용할 수 있게 될 때 Amazon Simple Notification Service (Amazon SNS) 알림을 구성할 수 있습니다. 이 파라미터가 null인 경우 구성이 스트림에서 삭제됩니다.

자세한 내용은 [Kinesis Video Streams의 알림](#)을 참조하십시오.

내용

DestinationConfig

고객에게 알림을 전달하는 데 필요한 대상 정보.

유형: [NotificationDestinationConfig](#) 객체

필수 여부: 예

Status

알림 구성이 활성화되었는지 비활성화되었는지를 나타냅니다.

타입: 문자열

유효 값: ENABLED | DISABLED

필수 여부: 예

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

NotificationDestinationConfig

서비스: Amazon Kinesis Video Streams

고객에게 알림을 전달하는 데 필요한 정보가 들어 있는 구조입니다.

내용

Uri

이미지가 배송될 위치를 식별하는 유니폼 리소스 식별자 (URI).

유형: 문자열

길이 제약: 최소 길이는 1. 최대 길이는 255.

패턴: `^[a-zA-Z_0-9]+:(//)?([/]+)/?([^*]*)$`

필수 여부: 예

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

RecorderConfig

서비스: Amazon Kinesis Video Streams

레코더 구성은 카메라에서 스트리밍되는 로컬 미디어 파일에 액세스하기 위한 자격 증명으로 사용되는 로컬 MediaSourceConfig 세부 정보로 구성됩니다.

내용

MediaSourceConfig

카메라로 스트리밍되는 미디어 파일에 액세스하는 데 필요한 자격 증명 (MediaUriSecretArn 및 MediaUriType) 으로 구성된 구성 세부 정보입니다.

유형: [MediaSourceConfig](#) 객체

필수 여부: 예

ScheduleConfig

로 구성된 ScheduleExpression 구성과 카메라 또는 로컬 미디어 파일에서 Edge Agent로 녹화하기 위한 일정을 지정하는 DurationInMinutes 세부 정보입니다. ScheduleExpression 속성을 제공하지 않으면 Edge Agent는 항상 녹화 모드로 설정됩니다.

유형: [ScheduleConfig](#) 객체

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

ResourceEndpointListItem

서비스: Amazon Kinesis Video Streams

GetSignalingChannelEndpointAPI에서 반환되는 신호 채널의 엔드포인트를 설명하는 객체입니다.

미디어 서버 엔드포인트는 WEBRTC 프로토콜에 해당합니다.

내용

Protocol

GetSignalingChannelEndpointAPI에서 반환한 신호 채널의 프로토콜입니다.

타입: 문자열

유효 값: WSS | HTTPS | WEBRTC

필수 여부: 아니요

ResourceEndpoint

API에서 반환한 신호 채널의 엔드포인트. GetSignalingChannelEndpoint

타입: 문자열

필수 항목 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

ScheduleConfig

서비스: Amazon Kinesis Video Streams

이 API를 사용하면 카메라 또는 로컬 미디어 파일이 Edge Agent에 녹화해야 하는 기간을 지정할 수 있습니다. 는 ScheduleExpression 및 DurationInMinutes 속성으로 ScheduleConfig 구성됩니다.

에 ScheduleConfig 이 (가) 제공되지 않은 경우 Edge Agent는 항상 레코딩 모드로 설정됩니다. RecorderConfig

에 ScheduleConfig 이 (가) 제공되지 않은 경우 Edge Agent는 정기적으로 (1시간마다) 업로드합니다. UploaderConfig

내용

DurationInSeconds

미디어를 녹화하는 총 시간입니다. ScheduleExpression속성을 제공한 경우 DurationInSeconds 속성도 지정해야 합니다.

유형: 정수

유효한 범위: 최소값은 60. 최대값은 3600입니다.

필수 여부: 예

ScheduleExpression

카메라 또는 로컬 미디어 파일에서 Edge Agent에 녹화하도록 작업을 예약하는 Quartz cron 표현식입니다. 에 ScheduleExpression 가 제공되지 않는 경우 Edge Agent는 항상 녹화 모드로 설정됩니다. RecorderConfig

Quartz에 대한 자세한 내용은 [크론 트리거 튜토리얼](#) 페이지를 참조하여 유효한 표현식과 사용법을 이해하십시오.

타입: 문자열

길이 제약: 최소 길이는 11입니다. 최대 길이는 100.

패턴: [^\n]{11,100}

필수 여부: 예

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

SingleMasterChannelEndpointConfiguration

서비스: Amazon Kinesis Video Streams

SINGLE_MASTER 채널 유형에 대한 엔드포인트 구성을 포함하는 객체입니다.

내용

Protocols

이 속성은 이 SINGLE_MASTER 신호 채널을 통한 통신의 특성을 결정하는 데 사용됩니다. 이 API를 지정하면 이 WSS API는 웹소켓 엔드포인트를 반환합니다. 이 지정된 경우 이 HTTPS API는 HTTPS 엔드포인트를 반환합니다.

유형: 문자열 어레이

배열 구성원: 최소수는 1개입니다. 최대 항목 수 5개.

유효 값: WSS | HTTPS | WEBRTC

필수 여부: 아니요

Role

이 속성은 이 SINGLE_MASTER 신호 채널의 메시징 권한을 결정하는 데 사용됩니다. 를 지정하면 이 MASTER API는 클라이언트가 이 신호 채널의 모든 시청자로부터 제안을 받고 답변을 보내는 데 사용할 수 있는 엔드포인트를 반환합니다. 를 지정하면 이 VIEWER API는 클라이언트가 이 신호 채널을 통해 다른 MASTER 클라이언트에게 오퍼를 보내는 데만 사용할 수 있는 엔드포인트를 반환합니다.

타입: 문자열

유효 값: MASTER | VIEWER

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

SingleMasterConfiguration

서비스: Amazon Kinesis Video Streams

SINGLE_MASTER 채널 유형에 대한 구성을 포함하는 구조입니다.

내용

MessageTtlSeconds

신호 채널이 전달되지 않은 메시지를 폐기하기 전에 보관하는 기간 (초) 입니다. 이 값을 업데이트 하는 데 사용합니다 [UpdateSignalingChannel](#).

유형: 정수

유효한 범위: 최소값은 5. 최대값은 120입니다.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

StreamInfo

서비스: Amazon Kinesis Video Streams

Kinesis 비디오 스트림을 설명하는 객체입니다.

내용

CreationTime

스트림이 생성된 시기를 나타내는 타임스탬프입니다.

유형: 타임스탬프

필수 여부: 아니요

DataRetentionInHours

스트림이 데이터를 유지하는 기간(시간)입니다.

유형: 정수

유효 범위: 최소값 0.

필수 여부: 아니요

DeviceName

스트림과 연결되는 디바이스의 이름입니다.

유형: 문자열

길이 제약: 최소 길이는 1. 최대 길이 128.

패턴: [a-zA-Z0-9_.-]+

Required: No

KmsKeyId

Kinesis Video Streams가 스트림의 데이터를 암호화하는 데 사용하는 AWS Key Management Service (AWS KMS) 키의 ID입니다.

유형: 문자열

길이 제약: 최소 길이는 1. 최대 길이는 2,048.

패턴: .+

Required: No

MediaType

스트림의 MediaType입니다.

유형: 문자열

길이 제약: 최소 길이는 1. 최대 길이 128.

패턴: [\w\-\.\.]+/[\w\-\.\.]+(,[\w\-\.\.]+/[\w\-\.\.]+)*

Required: No

Status

스트림의 상태.

타입: 문자열

유효 값: CREATING | ACTIVE | UPDATING | DELETING

필수 여부: 아니요

StreamARN

스트림의 Amazon 리소스 이름(ARN)입니다.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+

Required: No

StreamName

스트림의 이름입니다.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256입니다.

패턴: [a-zA-Z0-9_.-]+

Required: No

Version

스트림 버전입니다.

유형: 문자열

길이 제한: 최소 길이는 1. 최대 길이는 64.

패턴: [a-zA-Z0-9]+

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

StreamNameCondition

서비스: Amazon Kinesis Video Streams

스트림을 나열할 때 스트림이 반환되기 위해 충족해야 하는 조건을 지정합니다 (ListStreamsAPI 참조). 조건에는 비교 연산과 값이 있습니다. 현재는 이름이 지정된 접두사로 시작하는 스트림을 찾는 BEGINS_WITH 연산자만 지정할 수 있습니다.

내용

ComparisonOperator

비교 연산자. 현재는 이름이 지정된 접두사로 시작하는 스트림을 찾는 BEGINS_WITH 연산자만 지정할 수 있습니다.

타입: 문자열

유효 값: BEGINS_WITH

필수 여부: 아니요

ComparisonValue

비교할 값입니다.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256입니다.

패턴: [a-zA-Z0-9_.-]+

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

Tag

서비스: Amazon Kinesis Video Streams

지정된 신호 채널과 관련된 키 및 값 쌍입니다.

내용

Key

지정된 신호 채널과 연결된 태그의 키입니다.

유형: 문자열

길이 제약: 최소 길이는 1. 최대 길이 128.

패턴: $^([\p{L}\p{Z}\p{N}_\cdot:/=\+\-@]^*)\$$

필수 사항 여부: Yes

Value

지정된 신호 채널과 관련된 태그의 값입니다.

타입: 문자열

길이 제약: 최소 길이는 0. 최대 길이는 256입니다.

패턴: $[\p{L}\p{Z}\p{N}_\cdot:/=\+\-@]^*$

필수 여부: 예

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

UploaderConfig

서비스: Amazon Kinesis Video Streams

로 구성된 ScheduleExpression 구성과 카메라 또는 로컬 미디어 파일에서 Edge Agent로 녹화하기 위한 일정을 지정하는 DurationInMinutes 세부 정보입니다. 예 ScheduleConfig 이 (가) 제공되지 않은 경우 Edge Agent는 정기적으로 (1시간마다) 업로드합니다. UploaderConfig

내용

ScheduleConfig

카메라 또는 로컬 미디어 파일에서 Edge Agent로 녹화하는 일정을 지정하는 DurationInMinutes 세부 정보와 함께 구성된 구성입니다. ScheduleExpression 여기에 이 ScheduleConfig UploaderConfig 정보가 제공되지 않은 경우 Edge Agent는 정기적으로 (1시간마다) 업로드합니다.

유형: [ScheduleConfig](#) 객체

필수 여부: 예

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

Amazon Kinesis Video Streams

Amazon Kinesis Video Streams 미디어에서 지원하는 데이터 유형은 다음과 같습니다.

- [StartSelector](#)

StartSelector

서비스: Amazon Kinesis Video Streams Media

Kinesis 비디오 스트림에서 GetMedia API가 미디어 데이터 반환을 시작하도록 하려는 청크를 식별합니다. 다음과 같은 옵션을 사용하여 시작 청크를 식별할 수 있습니다.

- 최신 (또는 가장 오래된) 청크를 선택합니다.
- 특정 청크를 식별하십시오. 프래그먼트 번호나 타임스탬프 (서버 또는 프로듀서) 를 제공하여 특정 청크를 식별할 수 있습니다.
- 각 청크의 메타데이터에는 연속 토큰이 Matroska (MKV) 태그 () 로 포함되어 있습니다. `AWS_KINESISVIDEO_CONTINUATION_TOKEN` 이전 GetMedia 요청이 종료된 경우 다음 요청에서 이 태그 값을 사용할 수 있습니다. GetMedia 그러면 API는 마지막 API가 종료된 지점부터 청크를 반환하기 시작합니다.

내용

StartSelectorType

Kinesis 비디오 스트림에서 데이터 가져오기를 시작하려는 프래그먼트를 식별합니다.

- `지금` - 스트림의 최신 청크부터 시작하세요.
- `가장 빠른 시간` - 스트림에서 사용할 수 있는 가장 빠른 청크부터 시작합니다.
- `FRAGMENT_NUMBER` - 특정 프래그먼트 뒤에 오는 청크부터 시작합니다. `AfterFragmentNumber` 파라미터도 지정해야 합니다.
- `PRODUCER_TIMESTAMP` 또는 `SERVER_TIMESTAMP` - 지정된 프로듀서 또는 서버 타임스탬프가 있는 프래그먼트가 포함된 청크부터 시작합니다. 를 추가하여 `StartTimestamp` 타임스탬프를 지정합니다.
- `CONTINUATION_TOKEN` - 지정된 연속 토큰을 사용하여 읽습니다.

Note

현재, 가장 빠른 또는 `CONTINUATION_TOKEN`을 로 선택하면 에 추가 `startSelectorType` 정보가 제공되지 않습니다. `startSelector`

타입: 문자열

유호 값: FRAGMENT_NUMBER | SERVER_TIMESTAMP | PRODUCER_TIMESTAMP | NOW | EARLIEST | CONTINUATION_TOKEN

필수 사항 여부: 예

AfterFragmentNumber

GetMediaAPI에서 프래그먼트 반환을 시작할 프래그먼트 번호를 지정합니다.

유형: 문자열

길이 제약: 최소 길이는 1. 최대 길이 128.

패턴: `^[0-9]+$`

Required: No

ContinuationToken

Kinesis Video Streams가 이전 응답에서 반환한 연속 토큰입니다GetMedia. 그런 다음 연속 토큰으로 식별된 청크로 GetMedia API가 시작됩니다.

유형: 문자열

길이 제약: 최소 길이는 1. 최대 길이 128.

패턴: `^[a-zA-Z0-9_\.\\-]+$`

Required: No

StartTimestamp

타임스탬프 값. 이 값은 PRODUCER_TIMESTAMP 또는 SERVER_TIMESTAMP를 로 선택하는 경우 필요합니다. startSelectorType 그런 다음 GetMedia API는 지정된 타임스탬프가 있는 프래그먼트가 포함된 청크로 시작합니다.

유형: 타임스탬프

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)

- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

Amazon Kinesis Video Streams

Amazon Kinesis Video Streams 아카이브된 미디어에서 지원하는 데이터 유형은 다음과 같습니다.

- [ClipFragmentSelector](#)
- [ClipTimestampRange](#)
- [DASHFragmentSelector](#)
- [DASHTimestampRange](#)
- [Fragment](#)
- [FragmentSelector](#)
- [HLSFragmentSelector](#)
- [HLSTimestampRange](#)
- [Image](#)
- [TimestampRange](#)

ClipFragmentSelector

서비스: Amazon Kinesis Video Streams Archived Media

프래그먼트 범위의 타임스탬프 범위와 타임스탬프 출처를 설명합니다.

프로듀서 타임스탬프가 중복된 프래그먼트는 중복이 제거됩니다. 즉, 프로듀서가 실제 클록 시간과 거의 같은 프로듀서 타임스탬프를 사용하여 프래그먼트 스트림을 생성하는 경우 요청된 타임스탬프 범위 내에 있는 모든 프래그먼트가 클립에 포함됩니다. 동일한 시간 범위와 매우 다른 시점 내에 일부 프래그먼트를 인제스트하면 가장 오래된 인제스트 프래그먼트 컬렉션만 반환됩니다.

내용

FragmentSelectorType

사용할 타임스탬프의 출처 (서버 또는 프로듀서)

타입: 문자열

유효 값: PRODUCER_TIMESTAMP | SERVER_TIMESTAMP

필수 사항 여부: 예

TimestampRange

반환할 타임스탬프의 범위.

유형: [ClipTimestampRange](#) 객체

필수 여부: 예

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

ClipTimestampRange

서비스: Amazon Kinesis Video Streams Archived Media

프래그먼트를 반환할 타임스탬프 범위.

내용

EndTimestamp

요청된 미디어의 타임스탬프 범위 끝입니다.

이 값은 지정된 시간으로부터 24시간 이내여야 StartTimestamp 하며 해당 StartTimestamp 값보다 이후여야 합니다. FragmentSelectorType 요청의 SERVER_TIMESTAMP 경우 이 값은 과거의 값이어야 합니다.

이 값에는 포함됩니다. 프래그먼트의 (시작) 타임스탬프와 EndTimestamp 비교됩니다. 값보다 먼저 시작해서 EndTimestamp 값을 넘어서는 프래그먼트는 세션에 포함됩니다.

유형: 타임스탬프

필수 여부: 예

StartTimestamp

프래그먼트를 반환할 타임스탬프 범위의 시작 타임스탬프입니다.

정확히 시작 시점 또는 그 이후에 시작하는 프래그먼트만 세션에 포함됩니다 StartTimestamp. 이전에 StartTimestamp 시작해서 그 이후에도 계속되는 프래그먼트는 세션에 포함되지 않습니다. 그렇다면 FragmentSelectorType 스트림 헤드보다 StartTimestamp 나중이어야 합니다. SERVER_TIMESTAMP

유형: 타임스탬프

필수 여부: 예

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DASHFragmentSelector

서비스: Amazon Kinesis Video Streams Archived Media

요청된 미디어의 타임스탬프 범위와 타임스탬프의 출처를 포함합니다.

내용

FragmentSelectorType

요청된 미디어의 타임스탬프 소스.

FragmentSelectorType가 [getDash StreamingSession URL:로 PRODUCER_TIMESTAMP](#) 설정된 경우 지정된 [FragmentSelector: PlaybackMode TimestampRange](#) 내에서 프로듀서 타임스탬프로 인제스트된 첫 번째 프래그먼트가 미디어 재생 목록에 포함됩니다. ON_DEMAND LIVE_REPLAY 또한 [TimestampRange](#) 인제스트된 첫 번째 프래그먼트 바로 다음 ([GetDash URL: 값까지](#)) 내에 프로듀서 타임스탬프가 있는 프래그먼트도 포함됩니다. [StreamingSession MaxManifestFragmentResults](#)

프로듀서 타임스탬프가 중복된 프래그먼트는 중복이 제거됩니다. 즉, 프로듀서가 실제 클록 시간과 거의 동일한 프로듀서 타임스탬프를 사용하여 프래그먼트 스트림을 생성하는 경우 MPEG-DASH 매니페스트에는 요청된 타임스탬프 범위 내의 모든 프래그먼트가 포함됩니다. 동일한 시간 범위와 매우 다른 시점 내에 일부 프래그먼트를 인제스트하는 경우 가장 오래된 인제스트 프래그먼트 컬렉션만 반환됩니다.

FragmentSelectorType가 [getDash StreamingSession URL:로](#) 설정된 경우 프로듀서 타임스탬프는 PlaybackMode LIVE MP4 PRODUCER_TIMESTAMP 프래그먼트와 중복 제거에 사용됩니다. 하지만 서버 타임스탬프를 기반으로 가장 최근에 인제스트된 프래그먼트는 MPEG-DASH 매니페스트에 포함됩니다. 즉, 과거에 인제스트된 프래그먼트에 현재 값이 포함된 프로듀서 타임스탬프가 있더라도 HLS 미디어 플레이리스트에는 포함되지 않습니다.

기본값은 SERVER_TIMESTAMP입니다.

타입: 문자열

유효 값: PRODUCER_TIMESTAMP | SERVER_TIMESTAMP

필수 여부: 아니요

TimestampRange

요청된 미디어의 타임스탬프 범위의 시작과 끝

있는 경우 PlaybackType 이 값은 없어야 합니다. LIVE

유형: [DASHTimestampRange](#) 객체

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

DASHTimestampRange

서비스: Amazon Kinesis Video Streams Archived Media

요청된 미디어의 타임스탬프 범위의 시작과 끝

있는 경우 PlaybackType 이 값은 없어야 합니다. LIVE

의 값은 DASHTimestampRange 모두 포함됩니다. 정확히 시작 시간 또는 그 이후에 시작하는 프래그먼트는 세션에 포함됩니다. 시작 시간 이전에 시작하여 시작 시간 이후에도 계속되는 프래그먼트는 세션에 포함되지 않습니다.

내용

EndTimeStamp

요청된 미디어의 타임스탬프 범위 종료 이 값은 지정된 시간으로부터 24시간 이내여야 StartTimestamp 하며 해당 StartTimestamp 값보다 이후여야 합니다.

FragmentSelectorType요청의 SERVER_TIMESTAMP 경우 이 값은 과거의 값이어야 합니다.

이 EndTimeStamp 값은 모드에는 필수이지만 ON_DEMAND 모드에는 선택사항입니다.

LIVE_REPLAY 를 LIVE_REPLAY 모드로 설정하지 않으면 EndTimeStamp 세션이 만료될 때까지 세션에 새로 인제스트된 프래그먼트가 계속 포함됩니다.

Note

이 값에는 포함됩니다. 프래그먼트의 (시작) 타임스탬프와 EndTimeStamp 비교됩니다. 값보다 먼저 시작해서 EndTimeStamp 값을 넘어서는 프래그먼트는 세션에 포함됩니다.

유형: 타임스탬프

필수 여부: 아니요

StartTimestamp

요청된 미디어의 타임스탬프 범위 시작

DASHTimestampRange값이 지정된 경우 해당 StartTimestamp 값이 필요합니다.

정확히 시작 시점 또는 그 이후에 StartTimestamp 시작하는 프래그먼트만 세션에 포함됩니다. 이전에 StartTimestamp 시작해서 그 이후에도 계속되는 프래그먼트는 세션에 포함되지 않습니다.

다. 그렇다면 `FragmentSelectorType` 스트림 헤드보다 `StartTimestamp` 나중이어야 합니다.
`SERVER_TIMESTAMP`

유형: 타임스탬프

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

Fragment

서비스: Amazon Kinesis Video Streams Archived Media

비디오 세그먼트나 기타 시간 구분 데이터를 나타냅니다.

내용

FragmentLengthInMilliseconds

프래그먼트와 관련된 재생 시간 또는 기타 시간 값.

유형: Long

필수 여부: 아니요

FragmentNumber

프래그먼트의 고유 식별자. 이 값은 수집 순서에 따라 일정하게 증가합니다.

유형: 문자열

길이 제약: 최소 길이는 1. 최대 길이 128.

패턴: $^{[0-9]+}$

Required: No

FragmentSizeInBytes

프래그먼트 및 포함된 미디어 데이터에 대한 정보를 포함한 전체 프래그먼트 크기.

유형: Long

필수 여부: 아니요

ProducerTimestamp

프래그먼트에 해당하는 프로듀서의 타임스탬프 (밀리초 단위).

유형: 타임스탬프

필수 여부: 아니요

ServerTimestamp

프래그먼트에 해당하는 AWS 서버의 타임스탬프 (밀리초 단위).

유형: 타임스탬프

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

FragmentSelector

서비스: Amazon Kinesis Video Streams Archived Media

프래그먼트 범위의 타임스탬프 범위와 타임스탬프 출처를 설명합니다.

시작 타임스탬프가 지정된 시작 시간보다 크거나 같고 종료 시간보다 작거나 같은 프래그먼트만 반환됩니다. 예를 들어 스트림에 다음과 같은 시작 타임스탬프가 있는 프래그먼트가 포함된 경우:

- 00:00:00
- 00:00:02
- 00:00:04
- 00:00:06

시작 시간이 00:00:01 이고 종료 시간이 00:00:04 인 프래그먼트 셀렉터 범위는 시작 시간이 00:00:02 및 00:00:04 인 프래그먼트를 반환합니다.

내용

FragmentSelectorType

사용할 타임스탬프의 출처 (서버 또는 프로듀서).

타입: 문자열

유효 값: PRODUCER_TIMESTAMP | SERVER_TIMESTAMP

필수 사항 여부: 예

TimestampRange

반환할 타임스탬프의 범위.

유형: [TimestampRange](#) 객체

필수 여부: 예

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)

- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

HLSFragmentSelector

서비스: Amazon Kinesis Video Streams Archived Media

요청된 미디어의 타임스탬프 범위와 타임스탬프의 출처를 포함합니다.

내용

FragmentSelectorType

요청된 미디어의 타임스탬프 소스.

FragmentSelectorType가 [GetHLS StreamingSession URL:로 PRODUCER_TIMESTAMP 설정된 경우 지정된 FragmentSelector: PlaybackMode TimestampRange](#) 내에서 프로듀서 타임스탬프로 인제스트된 첫 번째 프래그먼트가 미디어 재생 목록에 포함됩니다. ON_DEMAND LIVE_REPLAY 또한 [TimestampRange](#) 인제스트된 첫 번째 프래그먼트 바로 다음 ([GetHLS URL: 값까지](#)) 내에 프로듀서 타임스탬프가 있는 프래그먼트도 포함됩니다. [StreamingSession MaxMediaPlaylistFragmentResults](#)

프로듀서 타임스탬프가 중복된 프래그먼트는 중복이 제거됩니다. 즉, 프로듀서가 실제 클록 시간과 거의 동일한 프로듀서 타임스탬프를 사용하여 프래그먼트 스트림을 생성하는 경우 HLS 미디어 재생 목록에는 요청된 타임스탬프 범위 내의 모든 프래그먼트가 포함됩니다. 일부 프래그먼트가 동일한 시간 범위와 매우 다른 시점 내에 인제스트되는 경우 가장 오래된 인제스트 프래그먼트 컬렉션만 반환됩니다.

FragmentSelectorType가 [GetHLS StreamingSession PlaybackMode URL:로](#) 설정된 경우 제작자 타임스탬프는 LIVE MP4 PRODUCER_TIMESTAMP 프래그먼트와 중복 제거에 사용됩니다. 하지만 서버 타임스탬프를 기반으로 가장 최근에 인제스트된 프래그먼트는 HLS 미디어 재생 목록에 포함됩니다. 즉, 과거에 인제스트된 프래그먼트에 현재 값이 포함된 프로듀서 타임스탬프가 있더라도 HLS 미디어 플레이리스트에는 포함되지 않습니다.

기본값은 SERVER_TIMESTAMP입니다.

타입: 문자열

유효 값: PRODUCER_TIMESTAMP | SERVER_TIMESTAMP

필수 여부: 아니요

TimestampRange

요청된 미디어의 타임스탬프 범위의 시작과 끝

있는 경우 PlaybackType 이 값은 없어야 합니다. LIVE

유형: [HLSTimestampRange](#) 객체

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

HLSTimestampRange

서비스: Amazon Kinesis Video Streams Archived Media

요청된 미디어의 타임스탬프 범위의 시작과 끝.

있는 경우 PlaybackType 이 값은 없어야 합니다. LIVE

내용

EndTimeStamp

요청된 미디어의 타임스탬프 범위 끝입니다. 이 값은 지정된 시간으로부터 24시간 이내여야 StartTimestamp 하며 해당 StartTimestamp 값보다 이후여야 합니다.

FragmentSelectorType 요청의 SERVER_TIMESTAMP 경우 이 값은 과거의 값이어야 합니다.

이 EndTimestamp 값은 모드에는 필수이지만 ON_DEMAND 모드에는 선택사항입니다.

LIVE_REPLAY 를 LIVE_REPLAY 모드로 설정하지 않으면 EndTimestamp 세션이 만료될 때까지 세션에 새로 인제스트된 프래그먼트가 계속 포함됩니다.

Note

이 값에는 포함됩니다. 프래그먼트의 (시작) 타임스탬프와 EndTimestamp 비교됩니다. 값 보다 먼저 시작해서 EndTimestamp 값을 넘어서는 프래그먼트는 세션에 포함됩니다.

유형: 타임스탬프

필수 여부: 아니요

StartTimestamp

요청된 미디어의 타임스탬프 범위 시작

HLSTimestampRange 값이 지정된 경우 해당 StartTimestamp 값이 필요합니다.

정확히 시작 시점 또는 그 이후에 StartTimestamp 시작하는 프래그먼트만 세션에 포함됩니다. 이전에 StartTimestamp 시작해서 그 이후에도 계속되는 프래그먼트는 세션에 포함되지 않습니다. 그렇다면 FragmentSelectorType 스트림 헤드보다 StartTimestamp 나중이어야 합니다.

SERVER_TIMESTAMP

유형: 타임스탬프

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

Image

서비스: Amazon Kinesis Video Streams Archived Media

TimestampError, 및 를 포함하는 구조체ImageContent.

내용

Error

시도할 수 없는 오류로 인해 제공된 타임스탬프의 이미지가 추출되지 않았을 때 표시되는 오류 메시지입니다. 다음과 같은 경우 오류가 반환됩니다.

- 지정된 미디어가 존재하지 않습니다Timestamp.
- 지정된 시간 동안의 미디어는 이미지를 추출할 수 없습니다. 이 경우 미디어는 오디오 전용이거나 잘못된 미디어가 인제스트되었습니다.

타입: 문자열

유효 값: NO_MEDIA | MEDIA_ERROR

필수 여부: 아니요

ImageContent

Base64로 인코딩된 Image 객체의 속성입니다.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 6291456입니다.

필수 여부: 아니요

TimeStamp

비디오 스트림에서 이미지를 추출하는 데 사용되는 Image 객체의 속성입니다. 이 필드는 이미지의 간격을 관리하거나 페이지 매김 창을 더 잘 이해하는 데 사용됩니다.

유형: 타임스탬프

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

TimestampRange

서비스: Amazon Kinesis Video Streams Archived Media

프래그먼트를 반환할 타임스탬프 범위.

내용

EndTimestamp

프래그먼트를 반환할 타임스탬프 범위의 종료 타임스탬프입니다.

유형: 타임스탬프

필수 여부: 예

StartTimestamp

프래그먼트를 반환할 타임스탬프 범위의 시작 타임스탬프입니다.

유형: 타임스탬프

필수 여부: 예

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

Amazon Kinesis Video

다음 데이터 유형이 지원됩니다.

- [IceServer](#)

IceServer

서비스: Amazon Kinesis Video Signaling Channels

ICE 서버 연결 데이터의 구조입니다.

내용

Password

ICE 서버에 로그인하기 위한 암호.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256입니다.

패턴: [a-zA-Z0-9_.-]+

Required: No

Ttl

사용자 이름과 암호가 유효한 기간 (초).

유형: 정수

유효 범위: 최소값은 30입니다. 최대값은 86400입니다.

필수 여부: 아니요

Uris

[I-D에 지정된 형식의 URI 배열. `petithuguenin-behave-turn-uris` 스펙](#) 이 URI는 TURN 서버에 연결하는 데 사용할 수 있는 다양한 주소 및/또는 프로토콜을 제공합니다.

유형: 문자열 어레이

길이 제약: 최소 길이 1. 최대 길이는 256.

필수 여부: 아니요

Username

ICE 서버에 로그인하기 위한 사용자 이름.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256입니다.

패턴: [a-zA-Z0-9_.-]+

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- [AWS SDK for C++](#)
- [AWS Java V2용 SDK](#)
- [AWS 루비 V3용 SDK](#)

Amazon Kinesis Video Streams

다음 데이터 유형이 Amazon Kinesis Video Streams에서 다음 데이터 유형이 지원됩니다.

일반적인 오류

이 단원에는 모든 AWS 서비스의 API 작업에 대한 일반 오류가 나와 있습니다. 이 서비스의 API 작업에 대한 오류는 해당 API 작업 항목을 참조하십시오.

AccessDeniedException

이 작업을 수행할 수 있는 충분한 액세스 권한이 없습니다.

HTTP 상태 코드: 400

IncompleteSignature

요청 서명이 AWS 표준을 준수하지 않습니다.

HTTP 상태 코드: 400

InternalFailure

알 수 없는 오류, 예외 또는 장애 때문에 요청 처리가 실패했습니다.

HTTP 상태 코드: 500

InvalidAction

요청된 동작 또는 작업이 유효하지 않습니다. 작업을 올바르게 입력했는지 확인합니다.

HTTP 상태 코드: 400

InvalidClientId

제공된 X.509 인증서 또는 AWS 액세스 키 ID가 AWS의 레코드에 존재하지 않습니다.

HTTP 상태 코드: 403

NotAuthorized

이 작업을 수행하려면 권한이 있어야 합니다.

HTTP 상태 코드: 400

OptInRequired

AWS 액세스 키 ID는 서비스에 대한 구독이 필요합니다.

HTTP 상태 코드: 403

RequestExpired

요청이 요청상의 날짜 스탬프로부터 15분 이상, 또는 요청 만료 날짜(예: 미리 서명된 URL)로부터 15분 이상 경과한 후 서비스에 도달했거나, 요청상의 날짜 스탬프가 15분 이상 미래입니다.

HTTP 상태 코드: 400

ServiceUnavailable

서버의 일시적 장애로 인해 요청이 실패하였습니다.

HTTP 상태 코드: 503

ThrottlingException

요청 제한 때문에 요청이 거부되었습니다.

HTTP 상태 코드: 400

ValidationError

입력이 AWS 서비스에서 지정한 제약에 충족되지 않습니다.

HTTP 상태 코드: 400

공통 파라미터

다음 목록에는 모든 작업이 쿼리 문자열을 사용하여 Signature Version 4 요청에 서명하는 데 사용하는 파라미터가 포함되어 있습니다. 작업별 파라미터는 그 작업에 대한 항목에 나열되어 있습니다. 서명 버전 4에 대한 자세한 내용은 IAM 사용 설명서의 [AWS API 요청 서명](#)을 참조하십시오.

Action

수행할 작업입니다.

유형: 문자열

필수 항목 여부: 예

Version

요청이 작성되는 API 버전으로 YYYY-MM-DD 형식으로 표시됩니다.

유형: 문자열

필수 항목 여부: 예

X-Amz-Algorithm

요청 서명을 생성하는 데 사용된 해시 알고리즘입니다.

조건: HTTP 권한 부여 헤더 대신 쿼리 문자열에 인증 정보를 포함하는 경우 이 파라미터를 지정합니다.

유형: 문자열

유효한 값: AWS4-HMAC-SHA256

필수 항목 여부: 조건부

X-Amz-Credential

자격 증명 범위 값이며 액세스 키, 날짜, 대상으로 하는 리전, 요청하는 서비스 및 종료 문자열("aws4_request")이 포함된 문자열입니다. 값은 다음 형식으로 표시됩니다. access_key/YYYYMMDD/region/service/aws4_request.

자세한 내용은 IAM 사용 설명서의 [서명된 AWS API 요청 생성](#)을 참조하십시오.

조건: HTTP 권한 부여 헤더 대신 쿼리 문자열에 인증 정보를 포함하는 경우 이 파라미터를 지정합니다.

유형: 문자열

필수 항목 여부: 조건부

X-Amz-Date

서명을 만드는 데 사용되는 날짜입니다. 형식은 ISO 8601 기본 형식('YYYYMMDD'T'HHMMSS'Z') 이어야 합니다. 예를 들어 다음 날짜 시간은 유효한 X-Amz-Date 값: 20120325T120000Z.

조건: X-Amz-Date는 모든 요청에서 옵션이지만 서명 요청에 사용되는 날짜보다 우선할 때 사용 됩니다. 날짜 헤더가 ISO 8601 기본 형식으로 지정된 경우 X-Amz-Date가 필요하지 않습니다. X-Amz-Date를 사용하는 경우 항상 Date 헤더의 값을 재정의합니다. 자세한 내용은 IAM 사용 설명서의 [AWS API 요청 서명 요소를](#) 참조하십시오.

유형: 문자열

필수 항목 여부: 조건부

X-Amz-Security-Token

toAWS Security Token Service (AWS STS) 에 대한 호출을 통해 받은 임시 보안 토큰입니다. 의 임시 보안 인증 정보를 지원하는 서비스 목록은 IAM 사용 설명서 의 [IAM와 함께AWS 서비스 작동하는 서비스 목록은 IAM](#) 사용 설명서 의 와 함께 작동하는 서비스 에서 확인할 수 있습니다.AWS STS

조건: 의 임시 보안 인증 정보를 사용하는 경우AWS STS, 보안 토큰을 포함해야 합니다.

유형: 문자열

필수 항목 여부: 조건부

X-Amz-Signature

서명할 문자열과 파생된 서명 키에서 계산된 16진수로 인코딩된 서명을 지정합니다.

조건: HTTP 권한 부여 헤더 대신 쿼리 문자열에 인증 정보를 포함하는 경우 이 파라미터를 지정합니다.

유형: 문자열

필수 항목 여부: 조건부

X-Amz-SignedHeaders

표준 요청의 일부로 포함된 모든 HTTP 헤더를 지정합니다. 서명된 헤더 지정에 대한 자세한 내용은 IAM 사용 설명서의 [서명된AWS API 요청 생성을](#) 참조하십시오.

조건: HTTP 권한 부여 헤더 대신 쿼리 문자열에 인증 정보를 포함하는 경우 이 파라미터를 지정합니다.

유형: 문자열

필수 항목 여부: 조건부

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.