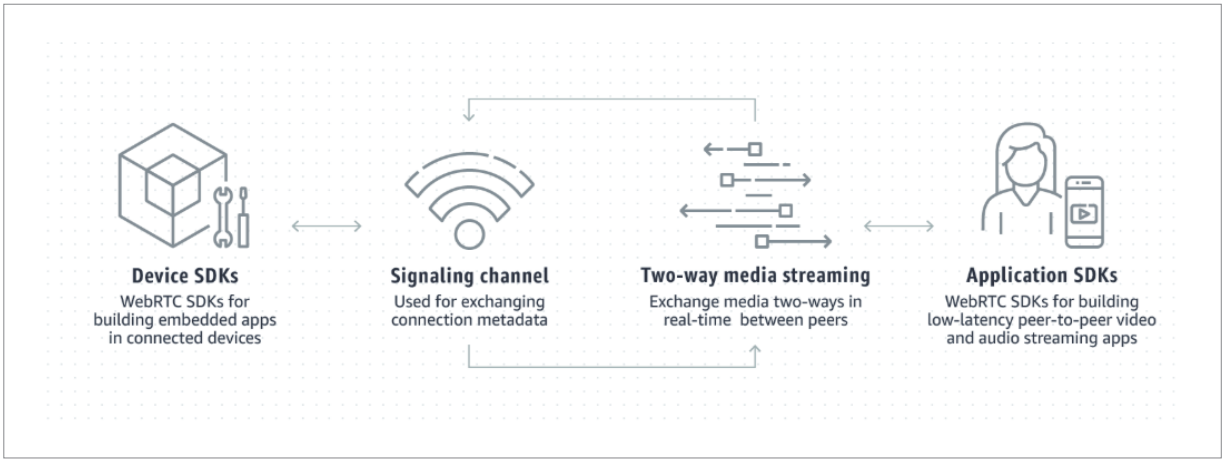
Amazon Kinesis Video Streams

Amazon Kinesis Video Streams의
WebRTC를 이용한 화상채팅 테스트

BESPIN GLOBAL
HELPING YOU ADOPT CLOUD.

노트북에 있는 웹캠(해상도 720p)과 AWS Kinesis Video Stream의 WebRTC을 이용하여 화상채팅을 테스트 해보았습니다.

1. 아키텍처



2. Amazon Kinesis Video Streams 개요

Amazon Kinesis Video Streams는 디바이스에서 AWS 클라우드로 라이브 비디오를 스트리밍하거나 실시간 비디오 처리 또는 배치 지향 비디오 분석을위한 애플리케이션을 구축하는 데 사용할 수 있는 완전 관리 형 AWS 서비스입니다.

Kinesis 비디오 스트림은 비디오 데이터를 위한 스토리지가 아닙니다. 클라우드를 통해 비디오 스트림을 실시간으로 시청할 수 있습니다. AWS Management Console에서 라이브 스트림을 모니터링하거나 Kinesis Video Streams API 라이브러리를 사용하여 라이브 비디오를 표시하는 자체 모니터링 애플리케이션을 개발할 수 있습니다.

Kinesis Video Streams를 사용하면 스마트 폰, 보안 카메라, 웹캠, 자동차에 내장된 카메라, 드론 및 기타 소스를 포함하여 수백만 개의 소스에서 방대한 양의 라이브 비디오 데이터를 캡처 할 수 있습니다. 오디오 데이터, 열 화상 이미지, 깊이 데이터, RADAR 데이터 등과 같은 비 비디오 시간 직렬화 데이터를 보낼 수도 있습니다. 이러한 소스에서 Kinesis 비디오 스트림으로 라이브 비디오 스트림으로 데이터를 프레임 단위로 액세스 할 수 있는 애플리케이션을 구축할 수 있습니다. 지연 시간이 짧은 처리를 위해 실시간으로. Kinesis 비디오 스트림은 소스에 구애받지 않습니다. 컴퓨터의 웹캠에서 비디오를 스트리밍 할 수 있습니다.

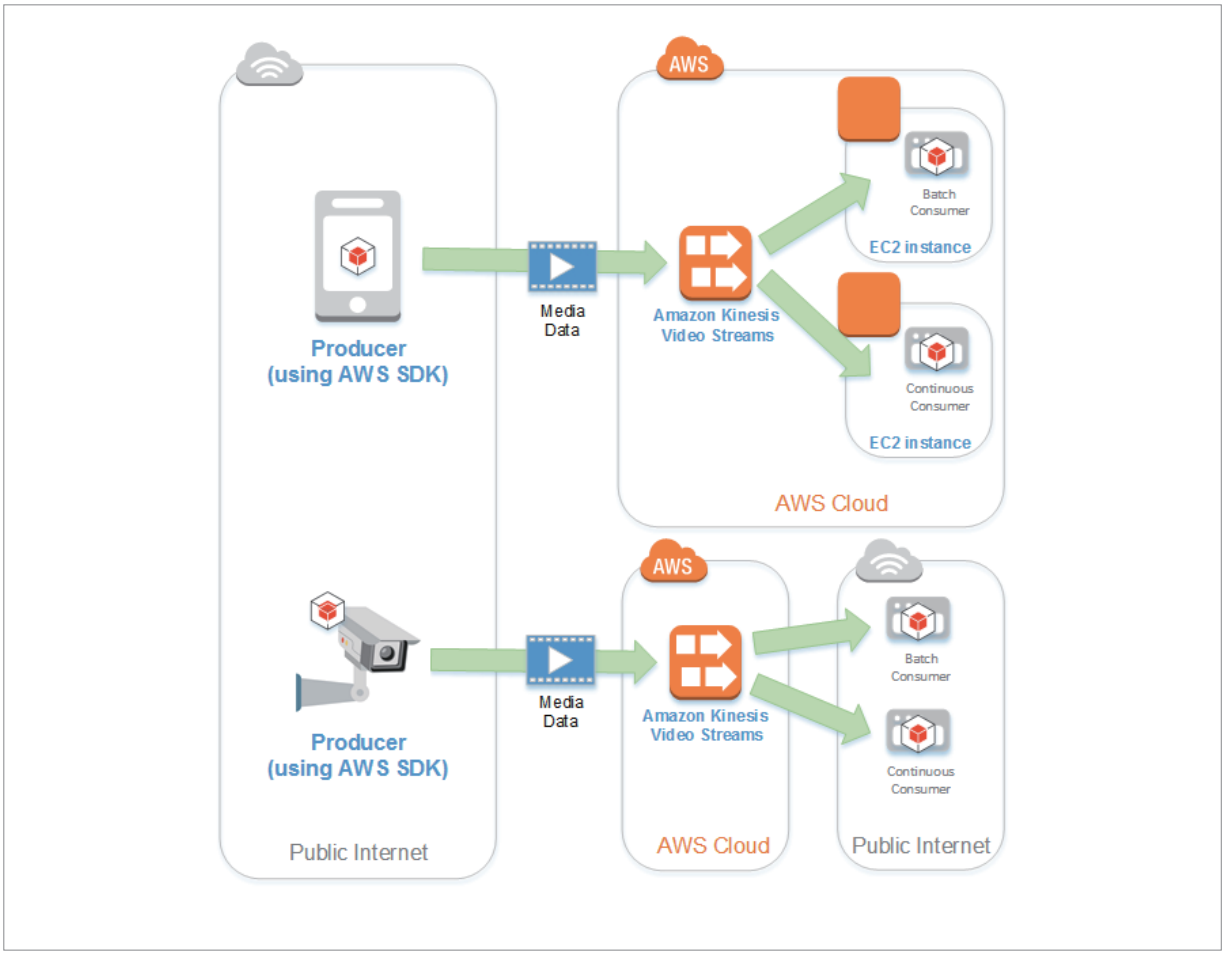
GStreamer 라이브러리 또는 RTSP를 사용하는 네트워크의 카메라에서 지정된 보존 기간 동안 미디어 데이터를 지속적으로 저장하도록 Kinesis 비디오 스트림을 구성할 수도 있습니다. Kinesis 비디오 스트림은 이 데이터를 자동으로 저장하고 저장시 암호화합니다. 또한 Kinesis Video Streams는 생산자 타임 스탬프와 처리 타임 스탬프를 기반으로 저장된 데이터를 타임 인덱스합니다. 비디오 데이터를 주기적으로 일괄 처리하는 응용 프로그램을 구축하거나 다양한 사용 사례에 대해 기록 데이터에 임시로 액세스해야하는 응용 프로그램을 만들 수 있습니다.

실시간 또는 배치 중심의 맞춤형 애플리케이션은 Amazon EC2 인스턴스에서 실행될 수 있습니다. 이러한 애플리케이션은 오픈 소스 딥 러닝 알고리즘을 사용하여 데이터를 처리하거나 Kinesis 비디오 스트림과 통합된 타사 애플리케이션을 사용할 수 있습니다.

2.1 Amazon Kinesis Video Streams 작동 방식

Amazon Kinesis Video Streams는 디바이스에서 AWS 클라우드로 라이브 비디오를 스트리밍하고 지속적으로 저장할 수 있는 완전 관리 형 AWS 서비스입니다. 그런 다음 실시간 비디오 처리를 위한 고유 한 응용 프로그램을 구축하거나 배치 지향 비디오 분석을 수행할 수 있습니다.

다음 다이어그램은 Kinesis 비디오 스트림 작동 방식에 대한 개요를 제공합니다.



다이어그램은 다음 구성 요소 간의 상호 작용을 보여줍니다.

· 프로듀서

데이터를 Kinesis 비디오 스트림에 넣는 모든 소스. 제작자는 보안 카메라, 신체 착용 카메라, 스마트 폰 카메라 또는 대시 보드 카메라와 같은 모든 비디오 생성 장치 일 수 있습니다. 제작자는 오디오 피드, 이미지 또는 RADAR 데이터와 같은 비 비디오 데이터를 보낼 수도 있습니다.

단일 제작자가 하나 이상의 비디오 스트림을 생성할 수 있습니다. 예를 들어 비디오 카메라는 비디오 데이터를 한 Kinesis 비디오 스트림으로 푸시하고 오디오 데이터를 다른 Kinesis 비디오 스트림으로 푸시 할 수 있습니다.

Kinesis Video Streams Producer 라이브러리 – 장치에 설치 및 구성할 수 있는 사용하기 쉬운 소프트웨어 및 라이브러리 세트. 이러한 라이브러리를 사용하면 실시간, 몇 초 동안 버퍼링 한 후 또는 실제 미디어 업로드와 같은 다양한 방식으로 비디오를 안전하게 연결하고 안정적으로 스트리밍 할 수 있습니다.

· Kinesis 비디오 스트림

실시간 비디오 데이터를 전송하고, 선택적으로 저장하며, 실시간 및 배치 또는 애드혹 기반으로 데이터를 소비할 수 있는 리소스입니다. 일반적인 구성에서 Kinesis 비디오 스트림에는 하나의 제작자만 데이터를 게시합니다.

스트림은 깊이 감지 피드, RADAR 피드 등과 같은 오디오, 비디오 및 유사한 시간 인코딩 된 데이터 스트림을 전달할 수 있습니다. AWS Management Console을 사용하거나 프로그래밍 방식으로 AWS SDK를 사용하여 Kinesis 비디오 스트림을 생성합니다.

여러 개의 독립적 인 애플리케이션이 Kinesis 비디오 스트림을 병렬로 사용할 수 있습니다.

· 소비자

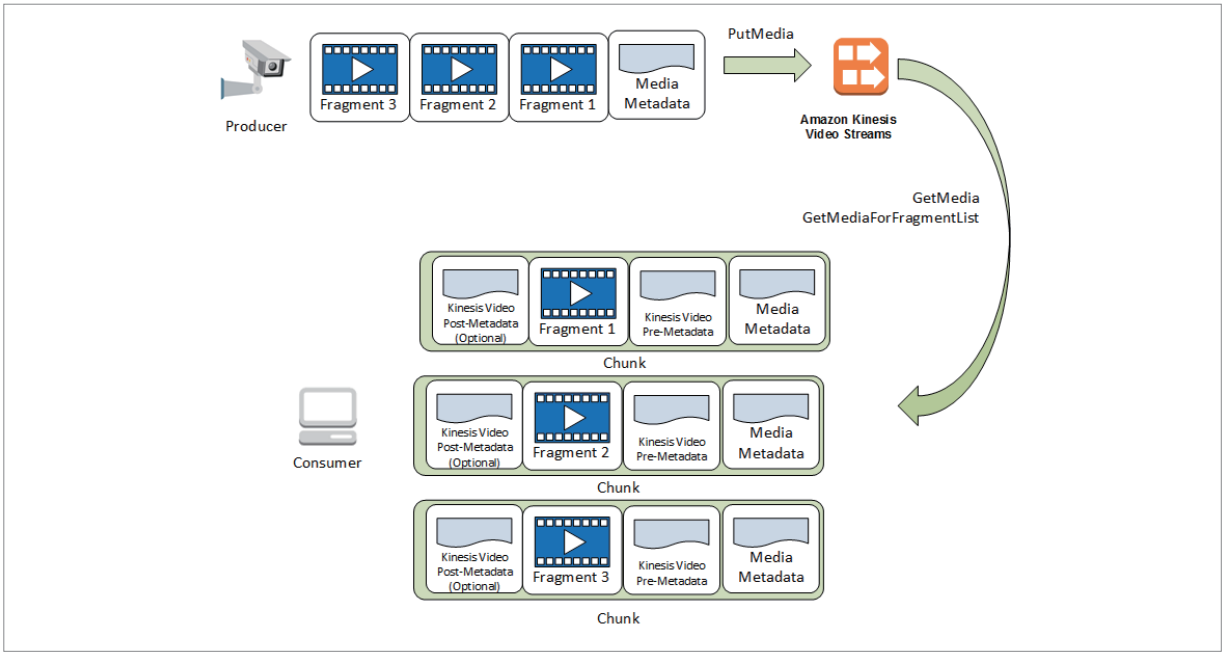
Kinesis 비디오 스트림에서 프래그먼트 및 프레임과 같은 데이터를 가져와서 보고 처리하거나 분석합니다. 일반적으로 이러한 소비자를 Kinesis 비디오 스트림 애플리케이션이라고 합니다. Kinesis 비디오 스트림에서 실시간으로 데이터를 소비 및 처리하는 애플리케이션을 작성하거나 지연 시간이 짧은 처리가 필요하지 않은 경우 데이터가 지속적으로 저장되고 시간 색인화 된 후 애플리케이션을 작성할 수 있습니다. 이러한 소비자 애플리케이션을 생성하여 Amazon EC2 인스턴스에서 실행할 수 있습니다.

Kinesis 비디오 스트림 파서 라이브러리 – Kinesis 비디오 스트림 애플리케이션이 지연 시간이 짧은 방식으로 Kinesis 비디오 스트림에서 미디어를 안정적으로 가져올 수 있습니다. 또한 미디어의 프레임 경계를 분석하여 응용 프로그램이 프레임 자체를 처리하고 분석하는 데 집중할 수 있도록 합니다.

2.2 Kinesis 비디오 스트림 API 및 생산자 라이브러리 지원

Kinesis 비디오 스트림은 스트림을 생성 및 관리하고 미디어 데이터를 스트림에서 읽거나 쓸 수 있는 API를 제공합니다. Kinesis Video Streams 콘솔은 관리 기능 외에도 실시간 및 주문형 비디오 재생을 지원합니다. Kinesis Video Streams는 또한 애플리케이션 코드에서 미디어 소스에서 데이터를 추출하고 Kinesis 비디오 스트림에 업로드 할 수 있는 생산자 라이브러리 세트를 제공합니다.

다음 다이어그램은 이러한 API 호출 중 조각 및 청크의 데이터 흐름을 보여줍니다.



2.3 Kinesis 비디오 스트림 재생

다음 방법을 사용하여 Kinesis 비디오 스트림을 볼 수 있습니다.

· GetMedia

GetMediaAPI를 사용하여 Kinesis 비디오 스트림을 처리하는 자체 응용 프로그램을 구축합니다. GetMedia는 지연 시간이 짧은 실시간 API입니다. GetMedia를 사용하는 플레이어를 만들려면 GetMediaAPI를 직접 빌드해야 합니다. GetMediaAPI를 사용하여 Kinesis 비디오 스트림을 표시하는 응용 프로그램을 개발하는 방법에 대한 자세한 내용 GetMedia은 스트림 파서 라이브러리를 참조하십시오.

· HLS

HTTP 라이브 스트리밍 (HLS)업계 표준 HTTP 기반 미디어 스트리밍 통신 프로토콜입니다. HLS를 사용하여 실시간 재생 또는 보관된 비디오를 위해 Amazon Kinesis 비디오 스트림을 볼 수 있습니다.

라이브 재생에 HLS를 사용할 수 있습니다. 지연 시간은 일반적으로 3 초에서 5 초 사이이지만 사용 사례, 플레이어 및 네트워크 상태에 따라 1 초에서 10 초 사이입니다. Video.js 와 같은 타사 플레이어를 사용할 수 있습니다. 또는 Google Shaka Player를 사용하여 프로그래밍 방식 또는 수동으로 HLS 스트리밍 세션 URL을 제공하여 비디오 스트림을 표시합니다. Apple Safari의 위치 표시 줄에 HLS 스트리밍 세션 URL을 입력하여 비디오를 재생할 수도 있습니다.

· MPEG-DASH

DASH (Dynamic Adaptive Streaming over HTTP) MPEG-DASH라고도하는 기존의 HTTP 웹 서버에서 제공되는 인터넷을 통해 미디어 컨텐츠를 고품질로 스트리밍 할 수 있는 적응 형 비트 전송률 스트리밍 프로토콜입니다.

라이브 재생에 MPEG-DASH를 사용할 수 있습니다. 지연 시간은 일반적으로 3 초에서 5 초 사이이지만 사용 사례, 플레이어 및 네트워크 상태에 따라 1 초에서 10 초 사이입니다. 타사 플레이어 (예: dash.js)를 사용할 수 있습니다. 또는 Google Shaka Player MPEG-DASH 스트리밍 세션 URL을 프로그래밍 방식 또는 수동으로 제공하여 비디오 스트림을 표시합니다.

2.4 HLS 재생을 위한 Kinesis 비디오 스트림 클라이언트 설정

HLS로 스트리밍 비디오에 액세스하려면 먼저 Kinesis Video Streams 클라이언트 (서비스 엔드 포인트를 검색) 및 아카이브 된 미디어 클라이언트 (HLS 스트리밍 세션을 검색)를 생성 및 구성하십시오. 응용 프로그램은 HTML 페이지의 입력 상자에서 필요한 값을 검색합니다.

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/aws-sdk/2.278.1/aws-sdk.min.js"></script>
...

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

2.5 Kinesis 비디오 스트림과 함께 스트리밍 메타 데이터 사용

Amazon Kinesis Video Streams Producer SDK를 사용하여 개별 조각 수준에서 메타 데이터를 Kinesis 비디오 스트림에 포함할 수 있습니다. Kinesis 비디오 스트림의 메타 데이터는 변경 가능한 키-값 쌍입니다. 이를 사용하여 프래그먼트의 내용을 설명하거나 실제 프래그먼트와 함께 전송해야하는 관련 센서 판독 값을 임베드하거나 다른 사용자 정의 요구를 충족시킬 수 있습니다. 메타 데이터는 GetMedia 또는 GetMediaFor-FragmentList의 일부로 제공됩니다. 스트림 보존 기간의 전체 기간 동안 프래그먼트와 함께 저장됩니다. 소비하는 애플리케이션은 Kinesis 비디오 스트림 파서 라이브러리를 사용하여 메타 데이터를 기반으로 읽고 처리하고 조치를 취할 수 있습니다.

메타 데이터를 스트림에 단편으로 임베드 할 수있는 두 가지 모드가 있습니다.
비 지속적: 발생한 비즈니스 별 기준에 따라 스트림의 조각에 임시로 메타 데이터를 첨부할 수 있습니다. 예를 들어 모션을 감지하고 모션을 포함하는 해당 프래그먼트에 메타 데이터를 추가 한 스마트 카메라가 있습니다. 다음 형식으로 조각에 메타 데이터를 적용할 수 있습니다 Motion = true.

지속적: 메타 데이터를 지속적인 요구에 따라 스트림의 연속적인 연속 조각에 첨부할 수 있습니다. 예를 들어 모든 조각과 관련된 현재 위도 및 경도 좌표를 Kinesis 비디오 스트림으로 보내는 스마트 카메라가 그 예입니다. 다음 형식으로 모든 조각에 메타 데이터를 적용할 수 있습니다.
Lat= 47.608013N, Long = -122.335167W

응용 프로그램의 요구에 따라 두 모드 모두에서 메타 데이터를 동시에 동일한 조각에 첨부할 수 있습니다. 임베드 된 메타 데이터에는 감지된 활동, 추적 된 활동, GPS 좌표 또는 스트림의 단편과 연관시킬 기타 사용자 정의 데이터가 포함될 수 있습니다. 메타 데이터는 키-값 문자열 쌍으로 인코딩 됩니다.

3. WebRTC 개요

WebRTC는 간단한 API를 통해 브라우저와 모바일 애플리케이션에서 실시간 통신 (RTC)을 가능하게 하는 개방형 기술 사양입니다. 연결된 피어 간의 실시간 데이터 교환을 위해 피어링 기술을 사용하고 사람과 사람의 상호 작용에 필요한 지연 시간이 짧은 미디어 스트리밍을 제공합니다. WebRTC 사양에는 대화식 연결 설정을 포함한 IETF 프로토콜 세트가 포함됩니다. NAT 주변의 릴레이를 사용하여 순회 (TURN)및 NAT에 대한 세션 탐색 유틸리티 (STUN) 신뢰할 수 있고 안전한 실시간 미디어 및 데이터 스트리밍을 위한 프로토콜 사양 외에도 P2P 연결을 설정합니다.

Amazon Kinesis 비디오 스트림완벽하게 관리되는 기능으로 표준 준수 WebRTC 구현을 제공합니다. WebRTC와 함께 Amazon Kinesis Video Streams를 사용하여 카메라 IoT 디바이스와 WebRTC 호환 모바일 또는 웹 플레이어간에 미디어를 안전하게 라이브 스트리밍하거나 양방향 오디오 또는 비디오 상호 작용을 수행할 수 있습니다. 완벽하게 관리되는 기능으로서, 신호 및 미디어 중계 서버와 같은 WebRTC 관련 클라우드 인프라를 구축, 운영 또는 확장할 필요가 없으므로 애플리케이션 및 장치에서 미디어를 안전하게 스트리밍 할 수 있습니다.

WebRTC와 함께 Kinesis Video Streams를 사용하면 라이브 P2P 미디어 스트리밍을 위한 응용 프로그램 또는 카메라 IoT 장치, 웹 브라우저 및 모바일 장치 간의 실시간 오디오 또는 비디오 대화 형 작업을 다양한 사용 사례를 위해 쉽게 구축할 수 있습니다.

3.1 WebRTC 관련 주요 용어 및 기술 개념

Signaling channel

응용 프로그램이 신호 메시지를 교환하여 피어 투 피어 연결을 검색, 설정, 제어 및 종료 할 수 있도록 하는 리소스입니다. 신호 메시지는 두 응용 프로그램이 서로 교환하여 피어 투 피어 연결을 설정하는 메타 데이터입니다. 이 메타 데이터에는 미디어 코덱 및 코덱 매개 변수와 같은 로컬 미디어 정보가 포함됩니다.

스트리밍 응용 프로그램은 신호 채널과의 지속적인 연결을 유지하고 다른 응용 프로그램이 연결될 때까지 기다릴 수 있습니다. 또는 라이브 스트림 미디어가 필요한 경우에만 신호 채널에 연결할 수 있습니다. 시그널링 채널을 통해 애플리케이션은 여러 뷰어에 연결하는 하나의 마스터 개념을 사용하여 일대 다 모델로 서로 연결할 수 있습니다. ConnectAsMasterAPI를 제공하고 시청자를 기다립니다. 그런 다음 ConnectAsViewerAPI 를 호출하여 뷰어 책임을 가정하여 최대 10 개의 애플리케이션이 해당 시그널링 채널에 연결할 수 있습니다. 마스터 및 뷰어 응용 프로그램이 신호 채널에 연결되면 라이브 미디어 스트리밍을 위한 피어 투 피어 연결을 설정하기 위해 서로 신호 메시지를 보낼 수 있습니다.

Peer

WebRTC를 사용하여 Kinesis 비디오 스트림을 통해 실시간 양방향 스트리밍을 위해 구성된 모든 장치 또는 애플리케이션 (예: 모바일 또는 웹 애플리케이션, 웹캠, 주택 보안 카메라, 베이비 모니터 등).

Master

연결을 시작하고 신호 채널의 연결된 뷰어와 미디어를 검색하고 교환할 수 있는 기능을 통해 신호 채널에 연결된 피어입니다.

*** 현재 신호 채널은 하나의 마스터 만 가질 수 있습니다.

Viewer
신호 채널의 마스터와 만 미디어를 검색하고 교환할 수 있는 기능을 통해 신호 채널에 연결된 피어입니다. 시청자는 지정된 신호 채널을 통해 다른 시청자를 발견하거나 상호 작용할 수 없습니다. 신호 채널에는 최대 10 명의 연결된 뷰어가 있을 수 있습니다.

Session Traversal Utilities for NAT (STUN)
공개 주소를 발견하고 라우터에서 피어와의 직접 연결을 방해하는 제한 사항을 판별하는 데 사용되는 프로토콜입니다.

Traversal Using Relays around NAT (TURN)
TURN 서버와의 연결을 열고 해당 서버를 통해 모든 정보를 릴레이하여 대칭 NAT 제한 사항을 무시하는 데 사용되는 서버입니다.

Session Description Protocol (SDP)
데이터가 전송되면 두 피어가 서로를 이해할 수 있도록 해상도, 형식, 코덱, 암호화 등과 같은 연결의 멀티미디어 컨텐츠를 설명하는 표준입니다.

SDP Offer
세션을 작성하거나 수정하기 위해 세션 설명을 생성하는 에이전트가 보낸 SDP 메시지. 원하는 미디어 통신의 측면을 설명합니다.

SDP Answer
제안자로부터 받은 제안에 대한 응답으로 응답자가 보낸 SDP 메시지. 대답은 허용되는 측면을 나타냅니다. 예를 들어, 오퍼의 모든 오디오 및 비디오 스트림이 승인된 경우.

Interactive Connectivity Establishment (ICE)
웹 브라우저가 피어와 연결할 수 있도록 하는 프레임 워크입니다.

ICE Candidate
송신 피어가 통신하는 데 사용할 수 있는 방법입니다.

3.2 STUN, TURN 및 ICE가 함께 작동하는 방법

WebRTC 피어를 사용하여 양방향 미디어 스트리밍 (예: 비디오 채팅 응용 프로그램)을 피어링하는 두 피어 A와 B의 시나리오를 살펴보겠습니다.

A가 B에게 전화를 걸려면 어떻게 됩니까?

B의 애플리케이션에 연결하려면 A의 애플리케이션이 SDP 오퍼를 생성해야 합니다. SDP 오퍼에는 사용할 코덱, 오디오 또는 비디오 세션 등을 포함하여 세션 A의 애플리케이션이 설정하려는 세션에 대한 정보가 포함됩니다. 또한 B의 애플리케이션이 사용하는 IP 및 포트 쌍인 ICE 후보 목록도 포함합니다. A에 연결하는 데 사용할 수 있습니다.

ICE 후보 목록을 작성하기 위해 A의 애플리케이션은 STUN 서버에 일련의 요청을 작성합니다. 서버는 요청을 시작한 퍼블릭 IP 주소와 포트 쌍을 반환합니다. A의 애플리케이션은 각 쌍을 ICE 후보 목록에 추가합니다. 즉, ICE 후보를 수집합니다. A의 신청이 ICE 후보 수집을 마치면 SDP를 반환할 수 있습니다.

다음으로 A의 애플리케이션은 이러한 애플리케이션이 통신하는 신호 채널을 통해 SDP를 B의 애플리케이션으로 전달해야 합니다. 이 교환에 대한 전송 프로토콜은 WebRTC 표준에 지정되어 있지 않습니다. HTTPS, 보안 WebSocket 또는 기타 통신 프로토콜을 통해 수행할 수 있습니다.

이제 B의 응용 프로그램은 SDP 응답을 생성해야 합니다. B의 응용 프로그램은 이전 단계에서 사용된 동일한 단계 A를 따릅니다. ICE 후보 등을 수집합니다. B의 응용 프로그램은이 SDP 답변을 A의 응용 프로그램에 반환해야 합니다.

A와 B가 SDP를 교환한 후 일련의 연결 확인을 수행합니다. 각 애플리케이션의 ICE 알고리즘은 상대방의 SDP에서 수신 한 목록에서 후보 IP/포트 쌍을 가져와서 STUN 요청을 보냅니다. 다른 응용 프로그램에서 응답이 반환되면 원래 응용 프로그램에서 검사가 성공한 것으로 간주하고 해당 IP/포트 쌍을 유효한 ICE 후보로 표시합니다.

모든 IP/포트 쌍에서 연결 확인이 완료된 후 응용 프로그램은 나머지 유효한 쌍 중 하나를 협상하고 사용하기로 결정합니다. 쌍을 선택하면 응용 프로그램 간에 미디어가 흐르기 시작합니다.

애플리케이션 중 하나가 연결 확인을 통과 한 IP/포트 쌍을 찾을 수 없는 경우 STUN 요청을 TURN 서버에 보내서 미디어 릴레이 주소를 얻습니다. 릴레이 주소는 공용 IP 주소 및 포트로, 릴레이 주소를 설정하기 위해 응용 프로그램과 주고받는 패킷을 전달합니다. 이 릴레이 주소는 후보 목록에 추가되고 신호 채널을 통해 교환됩니다.

4. WebRTC를 이용한 화상채팅 구현해보기

GitHub 리포지토리
https://github.com/awslabs/amazon-kinesis-video-streams-webrtc-sdk-js

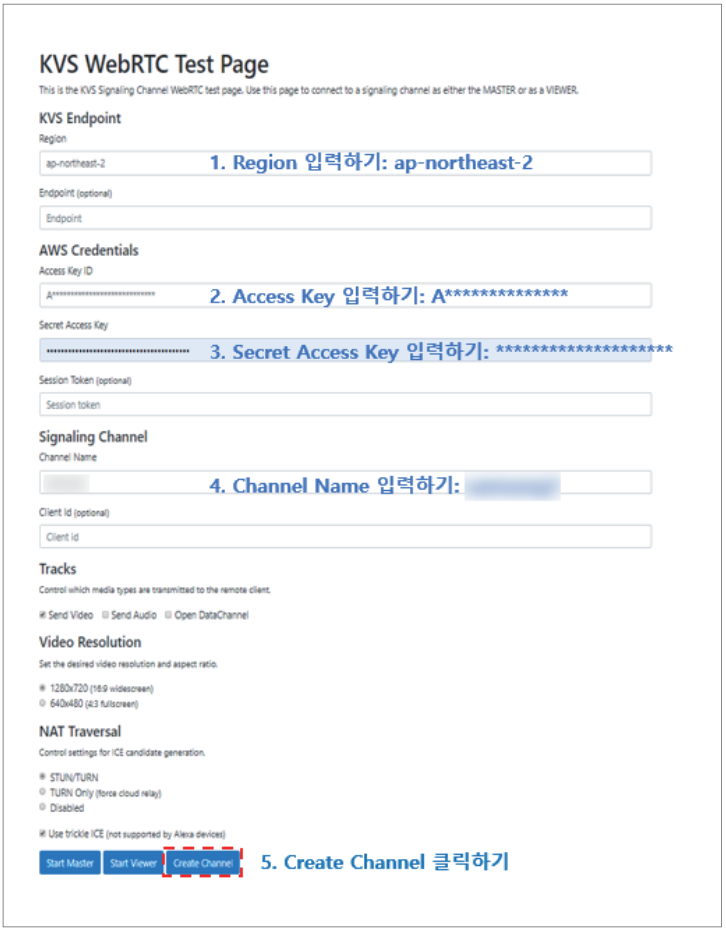
위 GitHub 리포지토리는 WebRTC 테스트 페이지가 있어서 Kinesis 비디오 스트림을 호스팅하여 새 채널을 생성하거나 기존 신호 채널에 연결하여 마스터 또는 뷰어로 사용할 수 있습니다.

4.1 사전 준비 사항

- 이 데모를 실행하기 위한 AWS 계정, 계정의 액세스 키와 시크릿 액세스 키
- 연결할 신호 채널의 이름
- 오디오, 비디오 또는 둘 다를 보낼지에 대한 여부
- 웹캠이 달려 있는 노트북 2대

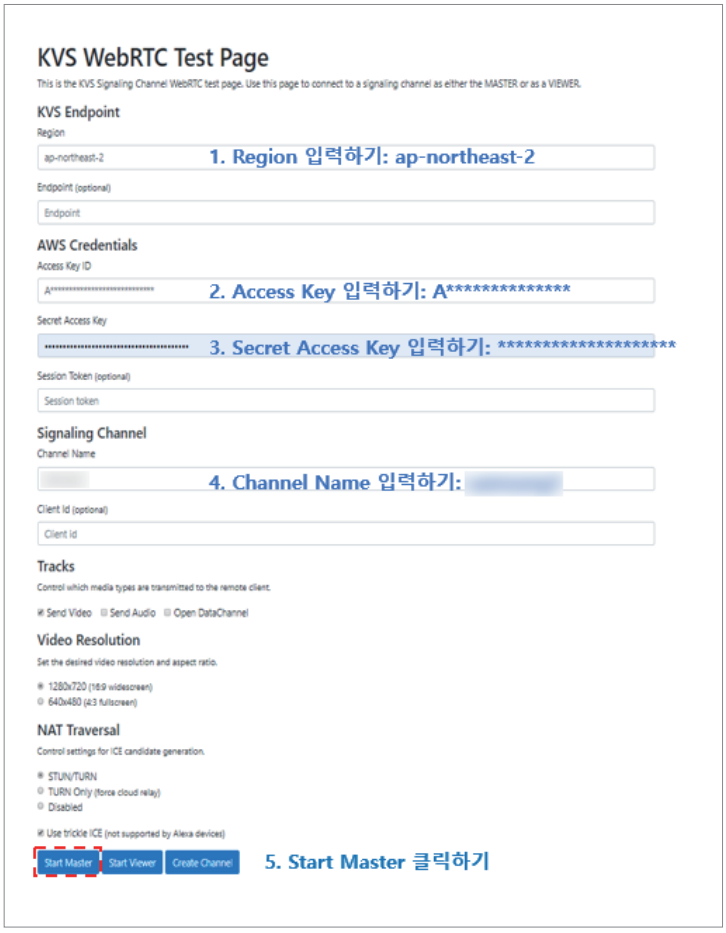
4.2 신규 채널 생성하기

- 데모 페이지
  https://awslabs.github.io/amazon-kinesis-video-streams-webrtc-sdk-js/examples/index.html
  을 클릭하여 웹페이지를 오픈합니다.



4.3 노트북에서 Master 실행하기

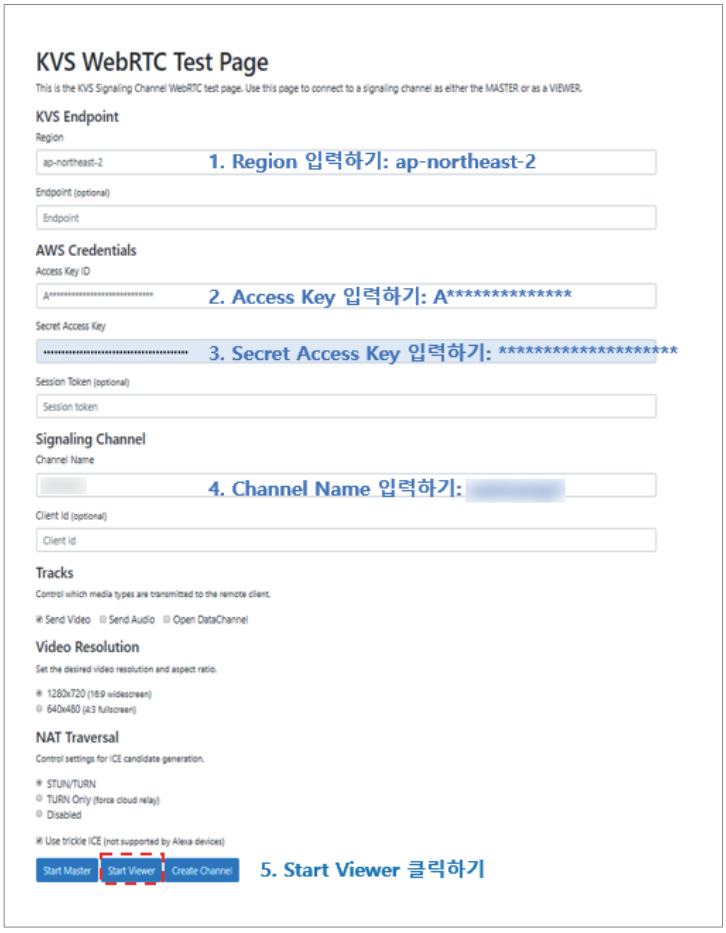
· 데모 페이지
https://awslabs.github.io/amazon-kinesis-video-streams-webrtc-sdk-js/examples/index.html
을 클릭하여 웹페이지를 오픈합니다.



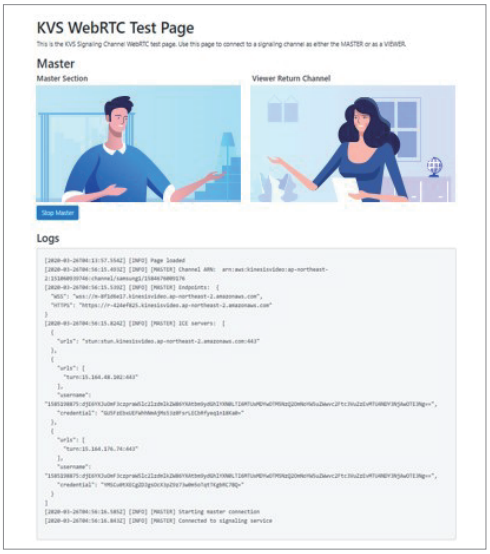
· Start Master를 실행하면 아래와 같이 웹캠에서 화면이 나오게됩니다.

4.4 노트북에서 Master 실행하기

· 다른 노트북에서 데모 페이지
https://awslabs.github.io/amazon-kinesis-video-streams-webrtc-sdk-js/examples/index.html
을 클릭하여 웹페이지를 오픈합니다.



· 다른 노트북에서 Start Viewer를 실행하면 아래와 같이 웹캠에서 양쪽 모두에서 화면이 나옵니다.
· Start View를 실행하고 화면이 나오기까지 약 20~30초가량 시간이 소요됩니다.



· 웹페이지에 Region, Access Key, Secret Access Key, Chanel Name을 입력할 때 공백이 들어가지 않도록 주의하도록합니다.

Source

aws.amazon.com에서 제공하는 Technical Document 및 AWS Lab 자료를 참고하여 수행한 결과입니다.