



AWS 백서

AWS 기반의 실시간 통신



AWS 기반의 실시간 통신: AWS 백서

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

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

Table of Contents

요약	1
요약	1
소개	2
RTC 아키텍처의 기본 구성 요소	3
소프트스위치/PBX	3
SBC(Session Border Controller)	4
PSTN 연결	4
PSTN 게이트웨이	4
SIP 트렁크	4
미디어 게이트웨이(트랜스코더)	4
WebRTC 및 WebRTC 게이트웨이	4
AWS의 고가용성 및 확장성	7
활성-대기 상태 유지 서버 간 HA를 위한 부동 IP 패턴	8
RTC 솔루션에서의 적용 가능성	8
AWS에서의 구현	8
장점	9
제한 사항 및 확장성	9
WebRTC 및 SIP를 통한 확장성 및 HA를 위한 로드 밸런싱	9
RTC 아키텍처에서의 적용 가능성	10
Application Load Balancer 및 Auto Scaling을 사용한 AWS 기반 WebRTC의 로드 밸런싱	10
Network Load Balancer 또는 AWS Marketplace 제품을 사용한 SIP 구현	11
교차 리전 DNS 기반 로드 밸런싱 및 장애 조치	12
영구 스토리지를 사용한 데이터 내구성 및 HA	14
AWS Lambda, Amazon Route 53, 및 AWS Auto Scaling를 사용한 동적 조정	15
Kinesis Video Streams를 사용한 고가용성 WebRTC	15
Amazon Chime 음성 커넥터를 사용한 고가용성 SIP 트렁킹	16
현장 모범 사례	17
SIP 오버레이 생성	17
세부 모니터링 수행	18
로드 밸런싱에 DNS를 사용하고 장애 조치에 부동 IP를 사용	18
다중 가용 영역 사용	19
트래픽을 단일 가용 영역 내에 유지하고 EC2 배치 그룹을 사용	20
향상된 네트워킹 EC2 인스턴스 유형을 사용	20
보안 고려 사항	21

결론	22
기여자	23
문서 개정	24
고지 사항	25

AWS 기반의 실시간 통신

AWS에서 고가용성의 확장 가능한 실시간 통신(RTC) 워크로드를 설계하는 모범 사례

게시 날짜: 2020년 2월 13일([문서 개정](#))

요약

오늘날 많은 조직에서 실시간 음성, 메시징 및 멀티미디어 워크로드에 대한 비용을 절감하고 확장성을 확보하려고 합니다. 이 백서에서는 AWS에서의 실시간 통신 워크로드 관리에 대한 모범 사례를 간략하게 설명하고 이러한 요구 사항을 충족하는 참조 아키텍처를 제시합니다. 이 백서는 실시간 커뮤니케이션 분야 전문가가 이러한 워크로드에 대해 고가용성 및 확장성을 달성하는 방법에 대한 지침으로 사용할 수 있습니다.

소개

음성, 비디오 및 메시징을 채널로 사용하는 통신 애플리케이션은 많은 조직 및 해당 최종 사용자의 핵심 요구 사항 중 하나입니다. 이러한 실시간 통신(RTC) 워크로드에는 특정 대기 시간 및 가용성 요구 사항이 있으며 이러한 요구 사항은 관련 설계 모범 사례를 따라 충족할 수 있습니다. 과거에는 전용 리소스를 사용하는 기존 온프레미스 데이터 센터에 RTC 워크로드가 배포되었습니다.

그러나 급격히 증가하고 성숙하는 기능 세트 덕분에 엄격한 서비스 수준 요구 사항에도 불구하고 RTC 워크로드를 Amazon Web Services(AWS)에 배포하고 확장성, 탄력성 및 고가용성의 이점을 누릴 수 있습니다. 오늘날 여러 고객이 AWS, 파트너 및 오픈 소스 솔루션을 사용하여 비용 절감, 향상된 민첩성, 몇 분 만에 글로벌화할 수 있는 능력, AWS 서비스의 풍부한 기능으로 RTC 워크로드를 실행하고 있습니다.

고객은 [Elastic Network Adapter\(ENA\)](#)를 통한 향상된 네트워킹, 최신 세대의 [Amazon Elastic Compute Cloud\(EC2\) 인스턴스](#)와 같은 AWS 기능을 활용하여 데이터 영역 개발 키트(DPDK), 단일 루트 I/O 가상화(SR-IOV), 대역폭 페이지, NVM Express(NVMe), NUMA(Non-Uniform Memory Access) 지원, [베어 메탈 인스턴스](#)를 통해 RTC 워크로드 요구 사항을 충족합니다. 이러한 인스턴스는 최대 100Gbps의 네트워크 대역폭 및 그에 상응하는 초당 패킷을 제공하여 네트워크 집약적 애플리케이션에 향상된 성능을 제공합니다. 확장을 위해 [Elastic Load Balancing](#)은 WebSocket 지원을 제공하는 [Application Load Balancer](#)와 초당 수백만 개의 요청을 처리할 수 있는 [Network Load Balancer](#)를 제공합니다. 네트워크 가속화를 위해 [AWS Global Accelerator](#)는 AWS에서 애플리케이션 엔드포인트에 대한 고정 진입점 역할을 하는 고정 IP 주소를 제공합니다. 또한 로드 밸런서에 대해 고정 IP 주소를 지원합니다. 대기 시간 감소, 비용 절감 및 대역폭 처리량 증가를 위해 [AWS Direct Connect](#)는 온프레미스에서 AWS로 전용 네트워크 연결을 설정합니다. 고가용성의 관리형 SIP 트렁킹은 [Amazon Chime 보이스 커넥터](#)가 제공합니다. [WebRTC가 포함된 Amazon Kinesis Video Streams](#)는 간편하게 실시간 양방향 미디어를 고가용성으로 스트리밍합니다.

이 백서에는 AWS에서 RTC 워크로드를 설정하는 방법과 최종 사용자 요구 사항을 충족하면서도 클라우드 최적화되도록 솔루션을 최적화하는 모범 사례를 보여주는 참조 아키텍처가 포함되어 있습니다. 이볼브드 패킷 코어(EPC)는 이 백서의 범위를 벗어나지만 세부 모범 사례는 가상 네트워크 기능(VNF)에 적용할 수 있습니다.

RTC 아키텍처의 기본 구성 요소

통신 산업에서 실시간 통신(RTC)은 일반적으로 대기 시간을 최소화한 두 엔드포인트 간의 라이브 미디어 세션을 말합니다. 이러한 세션은 다음과 관련될 수 있습니다.

- 두 당사자 간의 음성 세션(예: 전화 시스템, 모바일, VoIP)
- 인스턴트 메시징(예: 채팅, IRC)
- 라이브 비디오 세션(예: 화상 회의, 텔레프레즌스)

위의 각 솔루션에는 공통 구성 요소(예: 인증, 권한 부여 및 액세스 제어, 트랜스코딩, 버퍼링, 전달 등을 제공하는 구성 요소)와 전송되는 미디어 유형에 고유한 구성 요소(예: 브로드캐스트 서비스, 메시징 서버, 대기열 등)가 있습니다. 이 섹션에서는 음성 및 비디오 기반 RTC 시스템과 그림 1에 나와 있는 모든 관련 구성 요소를 정의하는 데 중점을 둡니다.

그림 1: RTC의 필수 아키텍처 구성 요소

주제

- [소프트스위치/PBX](#)
- [SBC\(Session Border Controller\)](#)
- [PSTN 연결](#)
- [미디어 게이트웨이\(트랜스코더\)](#)
- [WebRTC 및 WebRTC 게이트웨이](#)

소프트스위치/PBX

소프트스위치 또는 PBX는 음성 전화 시스템의 두뇌이며 다양한 구성 요소를 사용하여 기업 내부 또는 외부에서 음성 통화를 설정, 유지 관리 및 라우팅하기 위한 인텔리전스를 제공합니다. 기업의 모든 가입자는 전화를 받거나 걸려면 소프트웨어에 등록해야 합니다. 소프트웨어의 중요한 기능은 음성 네트워크 내의 다른 구성 요소를 사용하여 각 가입자를 추적하고 가입자에게 도달하는 방법을 추적하는 것입니다.

SBC(Session Border Controller)

세션 경계 컨트롤러(SBC)는 음성 네트워크의 엣지에 위치하며 모든 수신 및 발신 트래픽(제어 및 데이터 영역 모두)을 추적합니다. SBC의 주요 책임 중 하나는 음성 시스템을 악의적인 사용으로부터 보호하는 것입니다. SBC는 외부 연결을 위해 SIP(Session Initiation Protocol) 트렁크와 상호 연결하는 데 사용할 수 있습니다. 일부 SBC는 코덱을 한 형식에서 다른 형식으로 변환하기 위한 트랜스코딩 기능도 제공합니다. 마지막으로 대부분의 SBC는 방화벽 네트워크에서도 통화가 설정되도록 하는 데 도움이 되는 NAT 순회 기능도 제공합니다.

PSTN 연결

VoIP(보이스 오버 IP) 솔루션은 PSTN 게이트웨이 및 SIP 트렁크를 사용하여 레거시 PSTN 네트워크에 연결합니다.

PSTN 게이트웨이

PSTN(Public Switched Telephone Network) 게이트웨이는 신호(SIP와 SS7 사이) 및 미디어(코덱 트랜스코딩을 사용하여 RTP와 시분할 다중화 [TDM] 사이)를 변환합니다. PSTN 게이트웨이는 항상 PSTN 네트워크에 가까운 엣지에 위치합니다.

SIP 트렁크

SIP 트렁크에서 기업은 TDM(SS7 기반) 네트워크로의 통화를 종료하지 않고 기업과 통신사 간의 흐름은 IP를 통해 유지됩니다. 대부분의 SIP 트렁크는 SBC를 사용하여 설정됩니다. 기업은 특정 범위의 IP 주소, 포트를 허용하는 등 통신사의 사전 정의된 보안 규칙에 동의해야 합니다.

미디어 게이트웨이(트랜스코더)

일반적인 음성 솔루션은 다양한 유형의 코덱을 허용합니다. 몇 가지 일반적인 코덱은 북미에서는 G.711 μ -law이고, 북미 이외 지역에서 G.711 A-law, G.729 및 G.722입니다. 서로 다른 두 코덱을 사용하는 두 디바이스가 통신하면 미디어 서버가 디바이스 간의 코덱 흐름을 변환합니다. 즉, 미디어 게이트웨이는 미디어를 처리하고 최종 디바이스가 서로 통신할 수 있게 합니다.

WebRTC 및 WebRTC 게이트웨이

웹 실시간 통신(WebRTC)에서는 API를 사용하여 웹 브라우저에서 통화를 설정하거나 백엔드 서버에서 리소스를 요청할 수 있습니다. 이 기술은 클라우드 기술을 염두에 두고 설계되었으므로 통화를 설정

하는 데 사용할 수 있는 다양한 API를 제공합니다. 모든 음성 솔루션(SIP 포함)이 이러한 API를 지원하는 것은 아니기 때문에 WebRTC 게이트웨이는 API 호출을 SIP 메시지로 또는 그 반대로 변환하는 데 필요합니다.

그림 2에서는고가용성 WebRTC 아키텍처의 설계 패턴을 보여줍니다. WebRTC 클라이언트에서 수신되는 트래픽은 Auto Scaling 그룹의 일부인 EC2 인스턴스에서 실행되는 WebRTC에서 Amazon Application Load Balancer를 통해 로드 밸런싱됩니다.

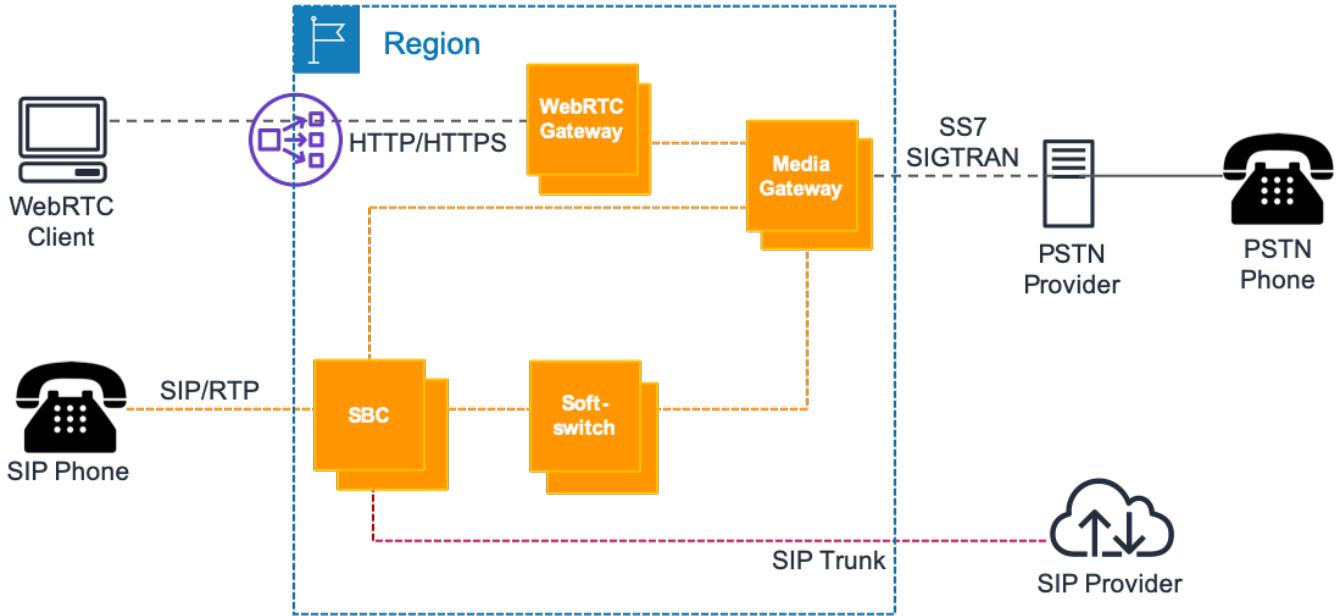


그림 2: 음성용 RTC 시스템의 기본 토폴로지

SIP 및 RTP 트래픽의 또 다른 설계 패턴은 가용 영역 전체에 걸쳐 Amazon EC2 기반 SBC 쌍을 활성 수동 모드로 사용하는 것입니다(그림 3). 여기서 DNS를 사용할 수 없는 장애가 발생할 경우 인스턴스 간에 탄력적 IP 주소를 동적으로 이동할 수 있습니다.

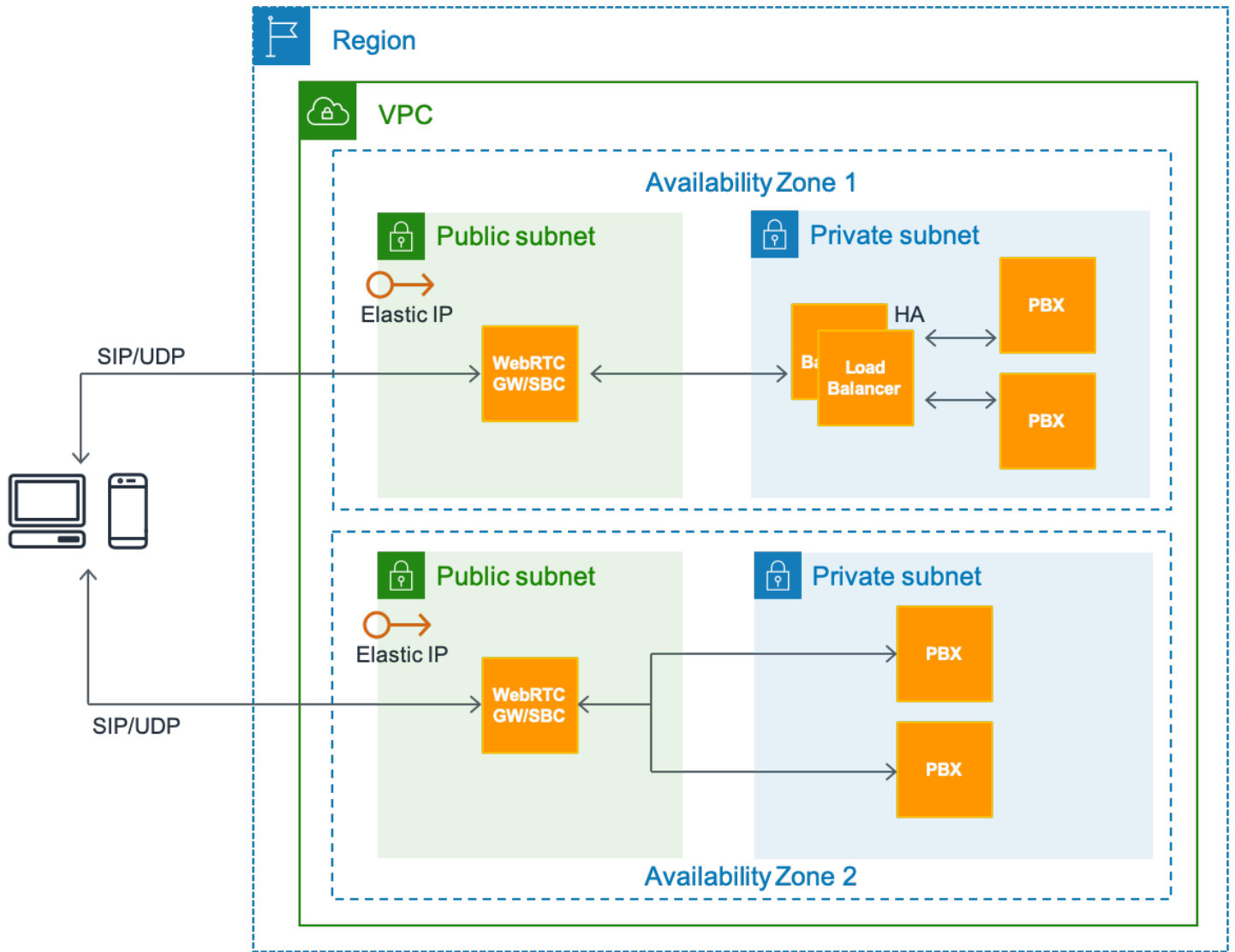


그림 3: VPC에서 Amazon EC2를 사용하는 RTC 아키텍처

AWS의 고가용성 및 확장성

대부분의 실시간 통신 제공업체는 99.9~99.999%의 가용성을 제공하는 서비스 수준을 유지합니다. 원하는 고가용성(HA) 정도에 따라 애플리케이션의 전체 수명 주기에서 갈수록 더 정교한 조치를 취해야 합니다. 강력한 수준의 고가용성을 달성하려면 다음 가이드라인을 권장합니다.

- 단일 장애 지점이 없도록 시스템을 설계합니다. 상태 유지 구성 요소와 무상태 구성 요소 모두에 대해 자동화된 모니터링, 장애 감지 및 장애 조치 메커니즘을 사용합니다.
- 단일 장애 지점(SPOF)은 일반적으로 N+1 또는 2N 이중화 구성을 통해 제거됩니다. 여기서 N+1은 활성-활성 노드 간의 로드 밸런싱을 통해 달성되고 2N은 활성-대기 구성의 노드 쌍으로 달성됩니다.
- AWS는 확장 가능하고 로드 밸런싱된 클러스터 또는 활성-대기 쌍을 사용하는 두 접근 방식을 통해 HA를 달성할 수 있는 몇 가지 방법을 제공합니다.
- 시스템 가용성을 올바르게 계측하고 테스트합니다.
- 장애에 대응, 완화 및 복구하기 위한 수동 메커니즘에 대한 운영 절차를 준비합니다.

이 섹션에서는 AWS에서 제공하는 기능을 사용하여 단일 장애 지점을 제거하는 방법을 중점적으로 다룹니다. 특히 이 섹션은 플랫폼에서 고가용성 실시간 통신 애플리케이션을 구축할 수 있는 일부 핵심 AWS 기능 및 설계 패턴을 설명합니다.

주제

- [활성-대기 상태 유지 서버 간 HA를 위한 부동 IP 패턴](#)
- [WebRTC 및 SIP를 통한 확장성 및 HA를 위한 로드 밸런싱](#)
- [교차 리전 DNS 기반 로드 밸런싱 및 장애 조치](#)
- [영구 스토리지를 사용한 데이터 내구성 및 HA](#)
- [AWS Lambda, Amazon Route 53, 및 AWS Auto Scaling를 사용한 동적 조정](#)
- [Kinesis Video Streams를 사용한 고가용성 WebRTC](#)
- [Amazon Chime 음성 커넥터를 사용한 고가용성 SIP 트렁킹](#)

활성-대기 상태 유지 서버 간 HA를 위한 부동 IP 패턴

부동 IP 설계 패턴은 활성-대기 하드웨어 노드 쌍(미디어 서버) 간에 자동 장애 조치를 수행하는 잘 알려진 메커니즘입니다. 활성 노드에 고정 보조 가상 IP 주소가 할당됩니다. 활성 노드와 대기 노드 간의 지속적인 모니터링이 장애를 감지합니다. 활성 노드에 장애가 발생하면 모니터링 스크립트가 가상 IP를 준비된 대기 노드에 할당하고 대기 노드가 기본 활성 노드의 기능을 대신합니다. 이러한 방식으로 가상 IP는 활성 노드와 대기 노드 간에 이동합니다.

주제

- [RTC 솔루션에서의 적용 가능성](#)
- [AWS에서의 구현](#)
- [장점](#)
- [제한 사항 및 확장성](#)

RTC 솔루션에서의 적용 가능성

N개의 노드로 구성된 활성-활성 클러스터와 같이 동일한 구성 요소의 활성 인스턴스를 여러 개 사용하는 것이 항상 가능하지는 않습니다. 활성-대기 구성은 HA에 가장 적합한 메커니즘입니다. 예를 들어 미디어 서버 또는 회의 서버, 심지어 SBC 또는 데이터베이스 서버와 같은 RTC 솔루션의 상태 유지 구성 요소는 활성-대기 설정에 적합합니다. SBC 또는 미디어 서버에는 특정 시점에 여러 개의 장기 실행 세션 또는 활성 채널이 있으며, SBC 활성 인스턴스에 장애가 발생할 경우 부동 IP로 인해 클라이언트 측 구성 없이 엔드포인트가 대기 노드에 다시 연결할 수 있습니다.

AWS에서의 구현

AWS에서 Amazon Elastic Compute Cloud(Amazon EC2), Amazon EC2 API, 탄력적 IP 주소의 핵심 기능과 Amazon EC2의 보조 프라이빗 IP 주소 지원을 사용하여 이 패턴을 구현할 수 있습니다.

1. 프라이머리 노드와 세컨더리 노드의 역할을 맡을 두 개의 EC2 인스턴스를 시작합니다. 프라이머리 노드는 기본적으로 활성 상태로 간주됩니다.
2. 추가 보조 프라이빗 IP 주소를 기본 EC2 인스턴스에 할당합니다.
3. 가상 IP(VIP)와 유사한 탄력적 IP 주소는 보조 프라이빗 주소와 연결됩니다. 이 보조 프라이빗 주소는 외부 엔드포인트가 애플리케이션에 액세스하는 데 사용하는 주소입니다.
4. 보조 IP 주소를 기본 네트워크 인터페이스에 별칭으로 추가하려면 약간의 OS 구성이 필요합니다.
5. 애플리케이션을 이 탄력적 IP 주소에 바인딩해야 합니다. Asterisk 소프트웨어의 경우 고급 Asterisk SIP 설정을 통해 바인딩을 구성할 수 있습니다.

6. 각 노드에서 모니터링 스크립트(사용자 지정, Linux 기반 KeepAlive, Corosync 등)를 실행하여 피어 노드의 상태를 모니터링합니다. 현재 활성 노드에 장애가 발생하면 피어는 이 장애를 감지하고 Amazon EC2 API를 호출하여 보조 프라이빗 IP 주소를 자신에게 재할당합니다.
7. 따라서 보조 프라이빗 IP 주소와 연결된 VIP에서 수신 대기하던 애플리케이션은 엔드포인트에서 대기 노드를 통해 사용할 수 있게 됩니다.

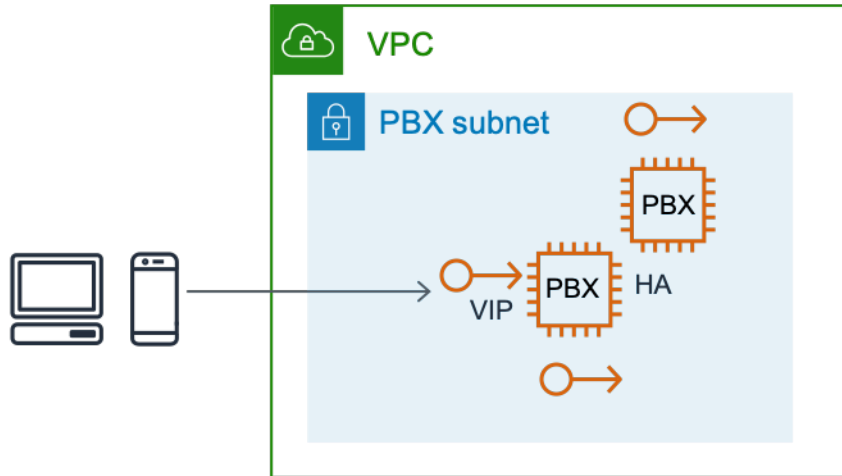


그림 4: 탄력적 IP 주소를 사용한 상태 유지 EC2 인스턴스 간 장애 조치

장점

이 접근 방식은 EC2 인스턴스, 인프라 또는 애플리케이션 수준에서 장애를 방지하는 안정적인 저예산 솔루션입니다.

제한 사항 및 확장성

이 설계 패턴은 일반적으로 단일 가용 영역 내로 제한됩니다. 두 가용 영역에 걸쳐 구현할 수 있지만 변형이 필요합니다. 이 경우 부동 탄력적 IP 주소는 사용 가능한 탄력적 IP 주소 재연결 API를 통해 서로 다른 가용 영역의 활성 노드와 대기 노드 간에 다시 연결됩니다. 그림 4에 표시된 장애 조치 구현에서는 진행 중인 통화가 종료되고 엔드포인트가 다시 연결되어야 합니다. 기본 세션 데이터의 복제를 통해 이 구현을 확장하여 원활한 세션 장애 조치 또는 미디어 연속성을 제공할 수도 있습니다.

WebRTC 및 SIP를 통한 확장성 및 HA를 위한 로드 밸런싱

라운드 로빈, 선호도, 대기 시간과 같은 사전 정의된 규칙을 기반으로 하는 활성 인스턴스 클러스터의 로드 밸런싱은 HTTP 요청의 무상태 특성 때문에 널리 사용되는 설계 패턴입니다. 실제로 많은 RTC 애플리케이션 구성 요소의 경우 로드 밸런싱은 실행 가능한 옵션입니다.

로드 밸런서는 원하는 애플리케이션에 대한 요청의 역방향 프록시 또는 진입점 역할을 하며, 이 애플리케이션은 여러 활성 노드에서 동시에 실행되도록 구성됩니다. 특정 시점에서 로드 밸런서는 정의된 클러스터의 활성 노드 중 하나로 사용자 요청을 보냅니다. 로드 밸런서는 대상 클러스터의 노드에 대해 상태 확인을 수행하여 상태 확인이 실패한 노드에는 수신된 요청을 보내지 않습니다. 따라서 로드 밸런싱을 통해 기본적인 수준의 고가용성을 확보할 수 있습니다. 또한 로드 밸런서는 1초 미만의 간격으로 모든 클러스터 노드에 대해 활성 및 수동 상태 확인을 수행하기 때문에 장애 조치는 거의 즉각적입니다.

요청을 전달할 노드에 대한 결정은 다음을 포함하여 로드 밸런서에 정의된 시스템 규칙에 따라 결정됩니다.

- 라운드 로빈
- 세션 또는 IP 선호도 - 동일한 세션 또는 동일한 IP의 여러 요청이 클러스터의 동일한 노드로 전송되도록 합니다.
- 대기 시간 기반
- 부하 기반

주제

- [RTC 아키텍처에서의 적용 가능성](#)
- [Application Load Balancer 및 Auto Scaling을 사용한 AWS 기반 WebRTC의 로드 밸런싱](#)
- [Network Load Balancer 또는 AWS Marketplace 제품을 사용한 SIP 구현](#)

RTC 아키텍처에서의 적용 가능성

WebRTC 프로토콜을 사용하면 Elastic Load Balancing, Application Load Balancer, Network Load Balancer 같은 HTTP 기반 로드 밸런서를 통해 WebRTC 게이트웨이를 쉽게 로드 밸런싱할 수 있습니다. 대부분의 SIP 구현은 TCP 및 UDP를 통한 전송에 의존하므로 TCP 및 UDP 기반 트래픽을 모두 지원하는 네트워크 또는 연결 수준 로드 밸런싱이 필요합니다.

Application Load Balancer 및 Auto Scaling을 사용한 AWS 기반 WebRTC의 로드 밸런싱

WebRTC 기반 통신의 경우 Elastic Load Balancing은 가용성 및 확장성이 뛰어난 완전관리형 로드 밸런서를 제공하여 요청의 진입점 역할을 하며, 그런 다음 요청은 Elastic Load Balancing과 연결된 EC2 인스턴스의 대상 클러스터로 전달됩니다. 또한 WebRTC 요청은 무상태이므로 Amazon EC2 Auto

Scaling을 사용하여 완전히 자동화되고 제어 가능한 확장성, 탄력성 및고가용성을 제공할 수 있습니다.

Application Load Balancer는 여러 가용 영역을 사용하여고가용성이고 확장 가능한 완전관리형 로드 밸런싱 서비스를 제공합니다. 이 로드 밸런서는 WebRTC 애플리케이션을 위한 신호 전송 및 장기 실행 TCP 연결을 사용하는 클라이언트와 서버 간의 양방향 통신을 처리하는 WebSocket 요청의 로드 밸런싱을 지원합니다. 또한 Application Load Balancer는 콘텐츠 기반 라우팅 및 스티키 세션을 지원하여, 로드 밸런서 생성 쿠키를 사용해 동일한 클라이언트에서 동일한 대상으로 요청을 라우팅합니다. 스티키 세션을 활성화할 경우 동일한 대상이 요청을 수신하고 쿠키를 사용하여 세션 컨텍스트를 복구할 수 있습니다.

그림 5에서는 대상 토폴로지를 보여줍니다.

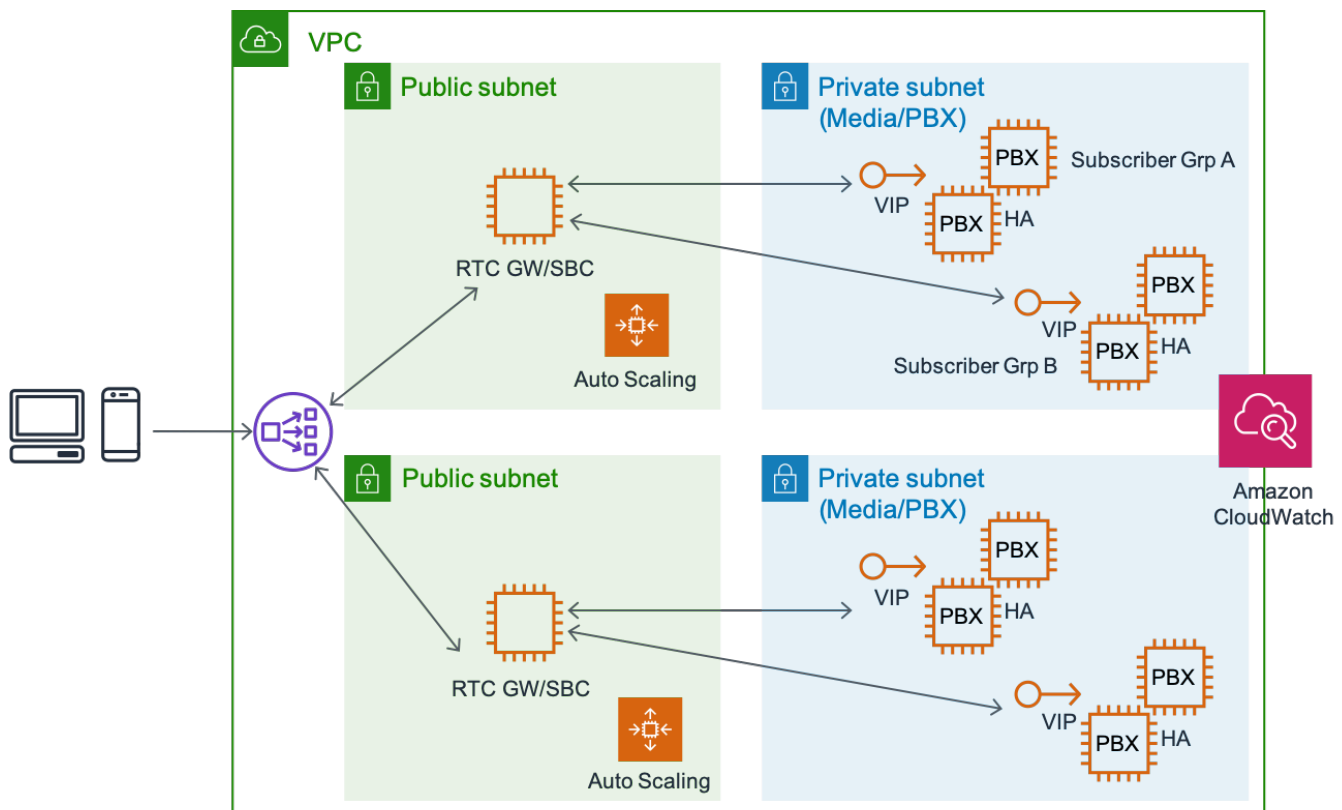


그림 5: WebRTC 확장성 및고가용성 아키텍처

Network Load Balancer 또는 AWS Marketplace 제품을 사용한 SIP 구현

SIP 기반 통신의 경우 TCP 또는 UDP를 통해 연결되는데, 대부분의 RTC 애플리케이션은 UDP를 사용합니다. SIP/TCP가 신호 프로토콜인 경우 Network Load Balancer를 사용하여 가용성, 확장성 및 성능이 뛰어난 완전관리형 로드 밸런싱을 수행할 수 있습니다.

Network Load Balancer는 연결 수준(계층 4)에서 작동하며 IP 프로토콜 데이터를 기반으로 Amazon EC2 인스턴스, 컨테이너, IP 주소와 같은 대상으로 연결을 라우팅합니다. TCP 또는 UDP 트래픽 로드 밸런싱에 적합한 네트워크 로드 밸런싱은 매우 짧은 대기 시간을 유지하면서 초당 수백만 개의 요청을 처리할 수 있습니다. AWS Auto Scaling, Amazon Elastic Container Service(Amazon ECS), Amazon Elastic Kubernetes Service(Amazon EKS), AWS CloudFormation 등 다른 인기 있는 AWS 서비스와 통합됩니다.

SIP 연결이 시작된 경우 다른 옵션은 AWS Marketplace 상용 소프트웨어(COTS)를 사용하는 것입니다. AWS Marketplace에서 UDP 및 기타 유형의 계층 4 연결 로드 밸런싱을 처리할 수 있는 여러 제품을 사용할 수 있습니다. 이러한 COTS는 일반적으로고가용성을 지원하며 흔히 AWS Auto Scaling 같은 기능과 통합되어 가용성 및 확장성을 더욱 향상시킬 수 있습니다. 그림 6에서는 대상 토폴로지를 보여줍니다.

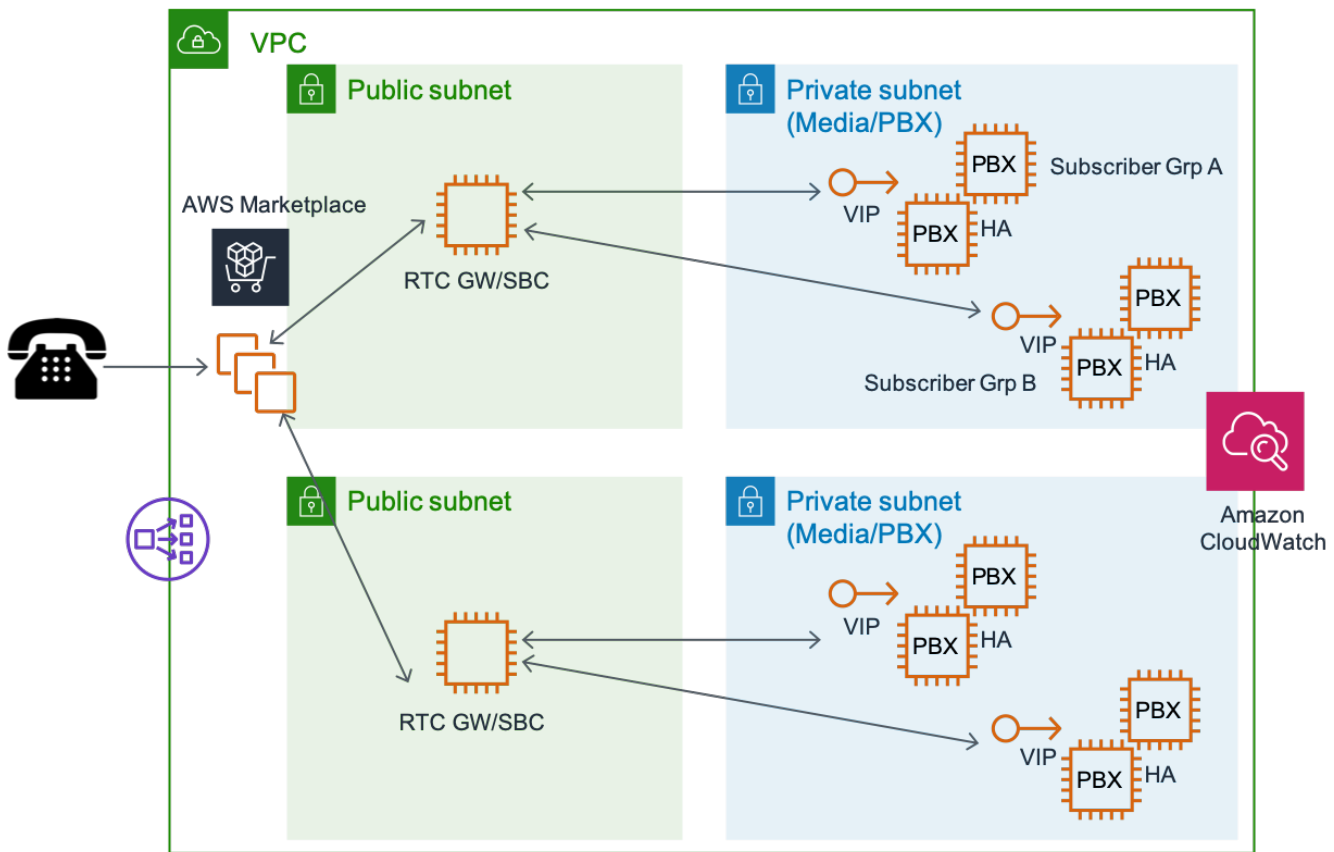


그림 6: AWS Marketplace 제품을 사용한 SIP 기반 RTC 확장성

교차 리전 DNS 기반 로드 밸런싱 및 장애 조치

Amazon Route 53는 RTC 클라이언트가 미디어 애플리케이션을 등록하고 연결할 수 있도록 퍼블릭 또는 프라이빗 엔드포인트로 사용할 수 있는 글로벌 DNS 서비스를 제공합니다. Amazon Route 53에

서는 트래픽을 정상 엔드포인트로 라우팅하거나 애플리케이션의 상태를 개별적으로 모니터링하도록 DNS 상태 확인을 구성할 수 있습니다. Amazon Route 53 트래픽 흐름 기능을 사용하면 지연 시간 기반 라우팅, 지역 DNS, 지역 근접성, 가중치 기반 라운드 로빈을 비롯하여 다양한 라우팅 유형을 통해 전역적으로 트래픽을 관리할 수 있습니다. 대기 시간이 짧고 내결함성이 있는 다양한 아키텍처를 제공하기 위해 이러한 모든 라우팅 유형을 DNS 장애 조치와 결합할 수 있습니다. Amazon Route 53 트래픽 흐름의 단순한 시각적 편집기를 사용하면 단일 AWS 리전이든 전 세계적으로 분산되어 있든 최종 사용자가 애플리케이션의 엔드포인트로 라우팅되는 방식을 쉽게 관리할 수 있습니다.

글로벌 배포의 경우 Route 53의 지연 시간 기반 라우팅 정책은 실시간 미디어 교환과 관련된 서비스 품질을 개선하기 위해 고객을 미디어 서버에 가장 가까운 지점으로 디렉션하는 데 특히 유용합니다.

새 DNS 주소로 장애 조치를 적용하려면 클라이언트 캐시를 플러시해야 합니다. 또한 DNS 변경이 글로벌 DNS 서버에 전파될 때 지연이 발생할 수 있습니다. 유지 시간(TTL) 속성을 사용하여 DNS 조회 새로 고침 간격을 관리할 수 있습니다. 이 속성은 DNS 정책을 설정할 때 구성할 수 있습니다.

글로벌 사용자에게 빠르게 도달하거나 단일 퍼블릭 IP 사용 요구 사항을 충족하기 위해 AWS Global Accelerator를 교차 리전 장애 조치에 사용할 수도 있습니다. AWS Global Accelerator는 로컬 및 글로벌 모두에서 애플리케이션의 가용성 및 성능을 향상시키는 네트워킹 서비스입니다. AWS Global Accelerator는 단일 또는 다중 AWS 리전의 Application Load Balancer, Network Load Balancer, Amazon EC2 인스턴스 같은 애플리케이션 엔드포인트에 대한 고정 진입점 역할을 하는 고정 IP 주소를 제공합니다. 또한 AWS 글로벌 네트워크를 사용하여 사용자부터 애플리케이션까지의 경로를 최적화함으로써 TCP 및 UDP 트래픽의 대기 시간과 같은 성능을 개선합니다. AWS Global Accelerator는 애플리케이션 엔드포인트의 상태를 지속적으로 모니터링하고, 현재 엔드포인트가 비정상적으로 전환되는 경우 트래픽을 가장 가까운 정상 엔드포인트로 자동으로 리디렉션합니다. 추가 보안 요구 사항의 경우 Accelerated Site-to-Site VPN은 AWS Global Accelerator를 사용하여 AWS 글로벌 네트워크 및 AWS 엣지 로케이션을 통해 트래픽을 지능적으로 라우팅함으로써 VPN 연결 성능을 개선합니다.

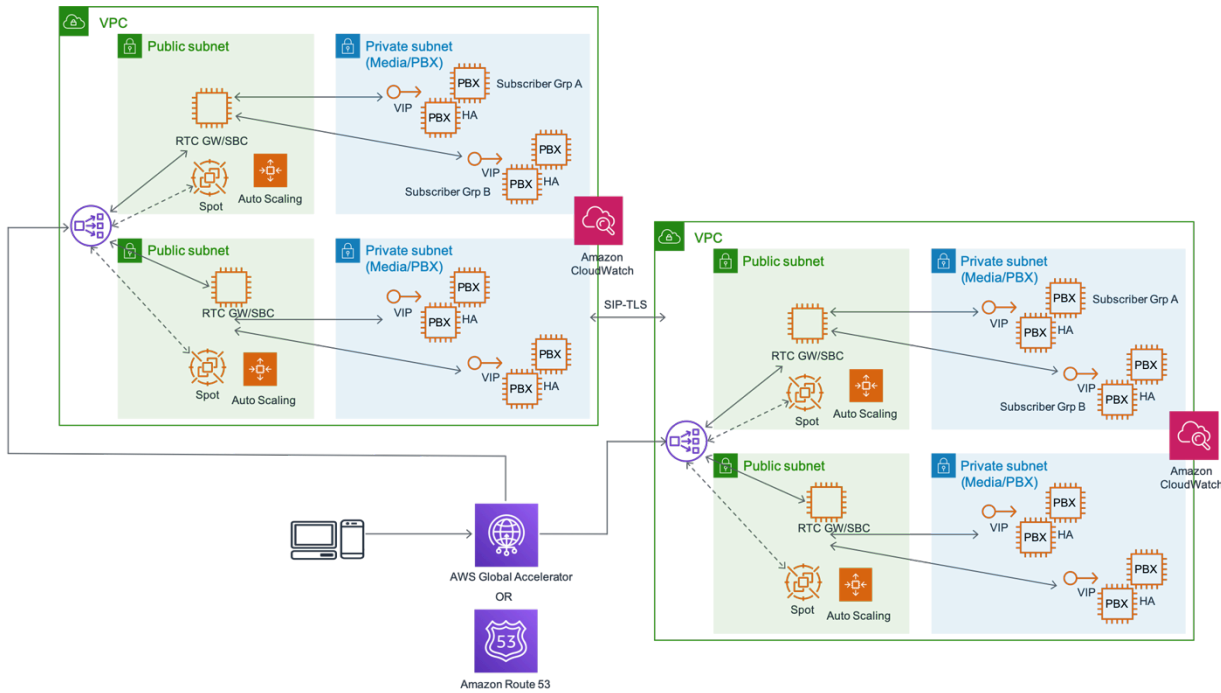


그림 7: AWS Global Accelerator 또는 Amazon Route 53를 사용한 리전 간 고가용성 설계

영구 스토리지를 사용한 데이터 내구성 및 HA

대부분의 RTC 애플리케이션은 인증, 권한 부여, 회계(세션 데이터, 통화 상세 기록 등), 운영 모니터링 및 로깅을 위해 데이터를 저장하고 액세스하기 위해 영구 스토리지를 사용합니다. 기존 데이터 센터에서 영구 스토리지 구성 요소(데이터베이스, 파일 시스템 등)의 고가용성 및 내구성을 보장하려면 일반적으로 SAN 설정, RAID 설계, 백업, 복원 및 장애 조치 처리를 위한 프로세스 등 상당한 업무 부담이 요구됩니다. AWS 클라우드는 데이터 내구성 및 가용성과 관련된 기존 데이터 센터 관행을 크게 간소화하고 개선합니다.

객체 스토리지 및 파일 스토리지의 경우 Amazon Simple Storage Service(Amazon S3), Amazon Elastic File System(Amazon EFS)과 같은 AWS 서비스가 관리형 고가용성 및 확장성을 제공합니다. Amazon S3의 데이터 내구성은 99.999999999%입니다.

트랜잭션 데이터 스토리지의 경우 고객은 고가용성 배포를 통해 Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle, Microsoft SQL Server를 지원하는 완전관리형 Amazon Relational Database Service(Amazon RDS)를 활용할 수 있습니다. 등록 기관 기능, 구독자 프로필 또는 회계 레코드 스토리지(예: CDR)의 경우 Amazon RDS가 내결함성, 고가용성 및 확장 가능한 옵션을 제공합니다.

AWS Lambda, Amazon Route 53, 및 AWS Auto Scaling를 사용한 동적 조정

AWS를 사용하면 여러 기능을 연결하고 인프라 이벤트를 기반으로 서비스형 사용자 지정 서버리스 함수를 통합할 수 있습니다. RTC 애플리케이션에서 다양한 용도로 사용할 수 있는 이러한 설계 패턴 중 하나는 자동 크기 조정 수명 주기 후크와 Amazon CloudWatch Events, Amazon Route 53 및 AWS Lambda 함수를 조합하는 것입니다. AWS Lambda 함수는 모든 작업 또는 로직을 포함할 수 있습니다. 그림 8에서는 이러한 기능을 함께 연결하여 자동화를 통해 시스템 안정성 및 확장성을 향상시키는 방법을 보여줍니다.

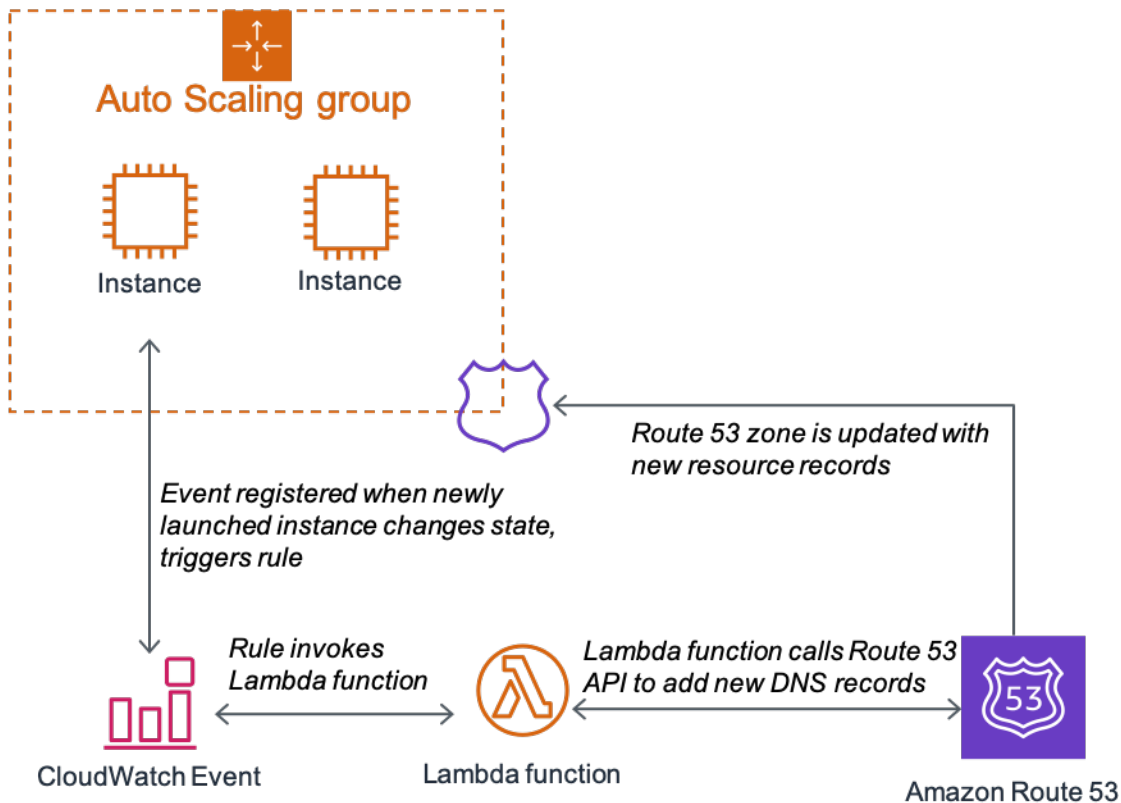


그림 8: Amazon Route 53에 대한 동적 업데이트를 통한 자동 크기 조정

Kinesis Video Streams를 사용한 고가용성 WebRTC

Amazon Kinesis Video Streams는 WebRTC를 통한 실시간 미디어 스트리밍을 제공하므로 사용자는 재생, 분석 및 기계 학습을 위해 미디어 스트림을 캡처, 처리, 저장할 수 있습니다. 이러한 스트림은 고가용성이고 확장 가능하며 WebRTC 표준을 준수합니다. Amazon Kinesis Video Streams에는 빠른 피어 검색 및 보안 연결 설정을 위한 WebRTC 신호 엔드포인트가 포함되어 있습니다. 여기에는 피어 간의 실시간 미디어 교환을 위해 관리형 STUN(Session Traversal Utilities for NAT) 및 TURN(Traversal

Using Rays around NAT) 엔드포인트가 포함됩니다. 또한 피어 검색 및 미디어 스트리밍이 가능하도록 카메라 펌웨어와 직접 통합되어 Kinesis Video Streams 엔드포인트와의 보안 통신을 가능하게 하는 무료 오픈 소스 SDK도 포함되어 있습니다. 마지막으로, WebRTC 호환 모바일 및 웹 플레이어가 미디어 스트리밍 및 양방향 통신을 위해 카메라 디바이스를 안전하게 검색하고 연결할 수 있도록 Android, iOS 및 JavaScript용 클라이언트 라이브러리를 제공합니다.

Amazon Chime 음성 커넥터를 사용한 고가용성 SIP 트렁킹

Amazon Chime 음성 커넥터는 회사에서 회사의 전화 시스템을 사용하여 저렴한 비용으로 안전하게 전화를 걸고 받을 수 있는 종량제 SIP 트렁킹 서비스를 제공합니다. Amazon Chime 음성 커넥터는 서비스 공급자 SIP 트렁크 또는 ISDN(Integrated Services Digital Network) PRI(Primary Rate Interface)를 대체하는 저렴한 비용의 서비스입니다. 고객은 수신 전화, 발신 전화 또는 둘 다를 활성화할 수 있습니다. 이 서비스는 AWS 네트워크를 활용하여 여러 AWS 리전 전체에서 고가용성 전화 경험을 제공합니다. SIP 트렁킹 전화 통화 또는 전달된 SIP 기반 미디어 레코딩(SIPREC) 피드의 오디오를 Amazon Kinesis Video Streams로 스트리밍하여 비즈니스 통화에서 실시간으로 인사이트를 얻을 수 있습니다. Amazon Transcribe 및 기타 일반적인 기계 학습 라이브러리와 통합을 통해 오디오 분석용 애플리케이션을 빠르게 구축할 수 있습니다.

현장 모범 사례

이 단원에서는 대규모 실시간 SIP(Session Initiation Protocol) 워크로드를 실행하는 가장 규모가 크고 가장 성공적인 AWS 고객 일부가 구현한 모범 사례를 요약합니다. 퍼블릭 클라우드에서 자체 SIP 인프라를 실행하려는 AWS 고객은 다양한 종류의 장애 발생 시 시스템 안정성 및 복원력을 높이는 데 도움이 된다는 점에서 이러한 모범 사례가 유용할 것입니다. 이러한 모범 사례 중 일부는 SIP와 관련되지만 대부분은 AWS에서 실행되는 모든 실시간 통신 애플리케이션에 적용됩니다.

주제

- [SIP 오버레이 생성](#)
- [세부 모니터링 수행](#)
- [로드 밸런싱에 DNS를 사용하고 장애 조치에 부동 IP를 사용](#)
- [다중 가용 영역 사용](#)
- [트래픽을 단일 가용 영역 내에 유지하고 EC2 배치 그룹을 사용](#)
- [향상된 네트워킹 EC2 인스턴스 유형을 사용](#)

SIP 오버레이 생성

AWS에는 리전 간 연결을 제공하는 강력하고 확장 가능한 이중화 네트워크 백본이 있습니다. 광섬유 절단과 같은 네트워크 이벤트로 인해 AWS 백본 링크에 문제가 발생하면 트래픽이 BGP와 같은 네트워크 수준 라우팅 프로토콜을 사용하여 중복 경로로 빠르게 장애 조치됩니다. 이 네트워크 수준 트래픽 엔지니어링은 AWS 고객에게는 블랙 박스이며 대부분 이러한 장애 조치 이벤트를 인지하지 못합니다. 그러나 음성, 고화질 비디오 및 낮은 대기 시간 메시징과 같은 실시간 워크로드를 실행하는 고객의 경우에는 이러한 이벤트가 인지될 수 있습니다. 그렇다면 AWS 고객이 네트워크 수준에서 AWS가 제공하는 것 외에 자체 트래픽 엔지니어링을 구현하려면 어떻게 해야 할까요? 솔루션은 다양한 AWS 리전에 SIP 인프라를 배포하는 것입니다. 통화 제어 기능의 일부로 SIP는 특정 SIP 프록시를 통해 통화를 라우팅하는 기능도 제공합니다.

그림 9: SIP 라우팅을 사용하여 네트워크 라우팅 재정의

그림 9에서 SIP 인프라(녹색 점으로 표시)가 미국 4개 리전 모두에서 실행되고 있습니다. 파란색 실선은 AWS 백본을 가상으로 묘사한 것입니다. SIP 라우팅이 구현되지 않은 경우 미국 서부 해안에서 시작하여 미국 동부 해안으로 향하는 통화는 오레곤 리전과 버지니아 리전을 직접 연결하는 백본 링크를 통과합니다. 이 다이어그램은 고객이 네트워크 수준 라우팅을 재정의하고 SIP 라우팅을 사용하여 동

일한 오레곤과 버지니아 간 통화가 캘리포니아를 경유하도록 하는 방법을 보여줍니다. 이러한 유형의 SIP 트래픽 엔지니어링은 SIP 재전송, 고객별 비즈니스 선호와 같은 네트워크 지표를 기반으로 SIP 프록시 및 미디어 게이트웨이를 사용하여 구현할 수 있습니다.

세부 모니터링 수행

실시간 음성 및 비디오 애플리케이션의 최종 사용자는 기존의 전화 통신 서비스와 동일한 수준의 성능을 기대합니다. 따라서 애플리케이션에 문제가 발생하면 결국 공급자의 평판이 손상됩니다. 사후 대응이 아닌 사전 예방적으로 행동하려면 최종 사용자에게 서비스를 제공하는 시스템의 모든 부분에 세부 모니터링을 배포해야 합니다.

그림 10: SIPp를 사용하여 VoIP 인프라 모니터링

SIP/RTP 트래픽을 모니터링하는 데 사용할 수 있는 [iPerf](#) 또는 [SIPp](#) 및 [VOIPMonitor](#) 같은 많은 오픈 소스 도구를 사용할 수 있습니다. 위의 예에서 클라이언트 및 서버 모드에 SIPp를 실행하는 노드는 4개의 미국 AWS 리전 모두에서 통화 성공 및 SIP 재전송과 같은 SIP 지표를 측정합니다. 그런 다음 사용자 지정 스크립트를 사용하여 이러한 지표를 Amazon CloudWatch로 내보낼 수 있습니다. CloudWatch를 사용하면 특정 임계값을 기반으로 이러한 사용자 지정 지표에 대한 경보를 생성할 수 있습니다. 그러면 이러한 CloudWatch 경보의 상태에 따라 자동 또는 수동 수정 조치를 취할 수 있습니다.

맞춤형 모니터링 시스템을 개발하고 유지 관리하는 데 필요한 엔지니어링 리소스를 할당하고 싶지 않은 고객을 위해 [ThousandEyes](#)와 같은 여러 우수한 VoIP 모니터링 솔루션을 시장에서 사용할 수 있습니다. 수정 작업의 예로는 SIP 재전송 증가에 따라 SIP 라우팅을 변경하는 것입니다.

로드 밸런싱에 DNS를 사용하고 장애 조치에 부동 IP를 사용

DNS SRV 기능을 지원하는 IP 텔레포니 클라이언트는 클라이언트를 서로 다른 SBC/PBX로 로드 밸런싱하여 인프라에 내장된 이중화를 효율적으로 사용할 수 있습니다.

그림 11: DNS SRV 레코드를 사용하여 SIP 클라이언트 로드 밸런싱

그림 11에서는 고객이 SRV 레코드를 사용하여 SIP 트래픽을 로드 밸런싱하는 방법을 보여줍니다. SRV 표준을 지원하는 모든 IP 텔레포니 클라이언트는 SRV 유형 DNS 레코드에서 sip._<transport protocol> 접두사를 찾습니다. 이 예에서 DNS의 응답 섹션에는 서로 다른 AWS 가용 영역에서 실행 중인 PBX가 모두 포함되어 있습니다. 그러나 엔드포인트 URI 외에도 SRV 레코드에는 다음과 같은 세 가지 추가 정보가 포함되어 있습니다.

- 첫 번째 숫자는 우선 순위(위 예에서는 1)입니다. 우선 순위는 숫자가 낮을수록 선호됩니다.
- 두 번째 숫자는 가중치(위 예에서는 10)입니다.
- 세 번째 숫자는 사용할 포트(5060)입니다.

두 PBX 서버의 우선 순위가 동일(1)하므로 클라이언트는 가중치를 사용하여 두 PBX 간의 로드 밸런싱을 수행합니다. 이 예에서는 가중치가 동일하므로 SIP 트래픽은 두 PBX 간에 균등하게 로드 밸런싱되어야 합니다.

DNS는 클라이언트 로드 밸런싱을 위한 훌륭한 솔루션이 될 수 있지만 DNS 'A' 레코드를 변경/업데이트하여 장애 조치를 구현하는 것은 어떨까요? 클라이언트 및 중간 노드 내의 DNS 캐싱 동작에서 불일치가 발견되므로 이 방법은 사용하지 않는 것이 좋습니다. SIP 노드 클러스터 간에 AZ 내부 장애 조치를 수행하는 더 좋은 방법은 EC2 API를 사용하여 손상된 호스트의 IP 주소가 정상 호스트에 즉시 재할당되는 EC2 IP 재할당을 사용하는 것입니다. 장애 노드 IP의 재할당은 세부 모니터링 및 상태 확인 솔루션과 결합되어 트래픽이 적시에 정상 호스트로 이동되도록 하여 최종 사용자 중단을 최소화합니다.

다중 가용 영역 사용

각 AWS 리전은 개별 가용 영역으로 세분화됩니다. 각 가용 영역은 자체적으로 전원, 냉각 및 네트워크 연결이 있으므로 격리된 장애 도메인을 형성합니다. AWS의 구조 내에서 고객은 항상 둘 이상의 가용 영역에서 워크로드를 실행하는 것이 좋습니다. 이를 통해 고객 애플리케이션은 전체 가용 영역 장애(실제로는 매우 드문 경우)도 견딜 수 있습니다. 이 권장 사항은 실시간 SIP 인프라에도 적용됩니다.

그림 12: 가용 영역 장애 처리

재해 이벤트(예: 5등급 허리케인)로 인해 us-east-1 리전에서 가용 영역이 완전히 중단되었다고 가정해 보겠습니다. 다이어그램과 같이 인프라가 실행 중인 경우 장애가 발생한 가용 영역의 노드에 등록되어 있던 모든 SIP 클라이언트는 가용 영역 #2에서 실행 중인 SIP 노드에 다시 등록되어야 합니다. (SIP 클라이언트/휴대폰으로 이 동작을 테스트하여 지원되는지 확인합니다.) 가용 영역 중단 시점의 활성 SIP 통화는 손실되지만 새 통화는 모두 가용 영역 #2를 통해 라우팅됩니다.

요약하면 DNS SRV 레코드는 클라이언트가 각 가용 영역에 하나씩 여러 'A' 레코드를 가리키도록 해야 합니다. 각 'A' 레코드는 해당 가용 영역에 있는 SBC/PBX의 여러 IP 주소를 가리켜 AZ 내부 및 AZ 간 복원력을 모두 제공해야 합니다. IP가 퍼블릭인 경우 IP 재할당을 사용하여 AZ 내부 및 AZ 간 장애 조치를 모두 구현할 수 있습니다. 하지만 프라이빗 IP는 가용 영역 간에 재할당할 수 없습니다. 프라이빗 IP 주소 지정을 사용하는 고객은 AZ 간 장애 조치를 위해 SIP 클라이언트가 백업 SBC/PBX에 다시 등록되어야 합니다.

트래픽을 단일 가용 영역 내에 유지하고 EC2 배치 그룹을 사용

가용 영역 선호도라고도 하는 이 모범 사례는 전체 가용 영역에 장애가 발생하는 드문 경우에도 적용됩니다. 하나의 가용 영역으로 들어오는 모든 SIP 또는 RTP 트래픽이 리전을 벗어날 때까지 해당 가용 영역에 남아 있도록 AZ 간 트래픽을 배제하는 것이 좋습니다.

그림 13: 가용 영역 선호도(활성 통화가 최대 50% 손실됨)

그림 13에서는 가용 영역 선호도를 사용하는 아키텍처를 개략적으로 보여줍니다. 전체 가용 영역 중단의 영향을 감안한다면 이 접근 방식의 비교 우위가 분명해집니다. 다이어그램에서 볼 수 있듯이 가용 영역 #2가 손실되면 활성 통화의 최대 50%가 영향을 받습니다(가용 영역 간의 로드 밸런싱이 동일하다고 가정). 가용 영역 선호도가 구현되지 않은 경우 일부 통화는 한 리전의 가용 영역 간에 흐르므로 장애가 발생하면 활성 통화의 50% 이상이 영향을 받을 가능성이 큼니다.

또한 트래픽 대기 시간을 최소화하려면 각 가용 영역 내에서 [EC2 배치 그룹](#)을 사용하는 것이 좋습니다. 동일한 EC2 배치 그룹에서 시작된 인스턴스는 EC2가 이러한 인스턴스의 네트워크 근접성을 보장하므로 대역폭이 더 높고 대기 시간이 줄어듭니다.

향상된 네트워킹 EC2 인스턴스 유형을 사용

Amazon EC2에 올바른 인스턴스 유형을 선택하면 시스템 안정성과 효율적인 인프라 사용이 모두 보장됩니다. EC2는 각 사용 사례에 맞게 최적화된 다양한 인스턴스 유형을 제공합니다. 인스턴스 유형은 CPU, 메모리, 스토리지 및 네트워킹 용량의 다양한 조합으로 구성되며, 애플리케이션에 따라 적합한 리소스 조합을 선택할 수 있는 유연성을 제공합니다. 이러한 향상된 네트워킹 인스턴스 유형을 통해 실행 중인 SIP 워크로드가 일관된 대역폭에 액세스할 수 있고 총 대기 시간을 비교적 줄일 수 있습니다. Amazon EC2에 최근 추가된 기능은 최대 100Gbps의 대역폭을 제공하는 Elastic Network Adapter(ENA)입니다. EC2 인스턴스 유형 및 관련 기능의 최신 카탈로그는 [EC2 인스턴스 유형 페이지](#)에서 확인할 수 있습니다.

대부분의 고객에게 최신 세대의 [컴퓨팅 최적화 인스턴스](#)는 최고의 가성비를 제공할 것입니다. 예를 들어 C5N은 최대 100Gbps의 대역폭과 초당 수백만 패킷(PPS)의 새로운 Elastic Network Adapter를 지원합니다. 또한 대부분의 실시간 애플리케이션은 네트워크 패킷 처리를 크게 향상시킬 수 있는 [Intel DPDK\(데이터 영역 개발자 키트\)](#)를 사용하면 이점을 얻을 수 있습니다.

그러나 항상 요구 사항에 따라 다양한 EC2 인스턴스 유형을 벤치마킹하여 어떤 인스턴스 유형이 가장 적합한지 확인하는 것이 모범 사례입니다. 벤치마킹을 사용하면 특정 인스턴스 유형이 한 번에 처리할 수 있는 최대 통화 수와 같은 다른 구성 파라미터도 확인할 수 있습니다.

보안 고려 사항

RTC 애플리케이션 구성 요소는 일반적으로 인터넷과 연결되는 Amazon EC2 인스턴스에서 직접 실행됩니다. 흐름은 TCP 외에도 UDP 및 SIP 같은 프로토콜을 사용합니다. 이러한 경우 AWS Shield Standard는 UDP 반사 공격, DNS 반사, NTP 반사, SSDP 반사 등과 같은 공통 인프라 계층(계층 3 및 4) DDoS 공격으로부터 Amazon EC2 인스턴스를 보호합니다. AWS Shield Standard는 잘 정의된 DDoS 공격 시그니처가 탐지되면 자동으로 실행되는 우선 순위 기반 트래픽 셰이핑과 같은 다양한 기술을 사용합니다.

또한 AWS는 탄력적 IP 주소에 AWS Shield Advanced를 활성화하여 이러한 애플리케이션에 대한 대규모 정교한 DDoS 공격에 대해 고급 보호 기능을 제공합니다. AWS Shield Advanced는 AWS 리소스의 유형과 EC2 인스턴스의 크기를 자동으로 감지하고 SYN 또는 UDP flood에 대한 보호와 함께 사전 정의된 적절한 완화 기능을 적용하는 향상된 DDoS 탐지 기능을 제공합니다. AWS Shield Advanced를 통해 고객은 연중무휴 24시간 가동하는 AWS DDoS 대응 팀(DRT)과 함께 자체 사용자 지정 완화 프로필을 생성할 수도 있습니다. 또한 AWS Shield Advanced는 DDoS 공격 중에 대규모 DDoS 공격을 완화하기 위해 추가 대역폭 및 스크러빙 용량을 제공하여 모든 Amazon VPC 네트워크 액세스 제어 목록(ACL)이 AWS 네트워크 경계에 자동으로 적용되도록 합니다.

결론

실시간 통신(RTC) 워크로드를 Amazon Web Services(AWS)에 배포하여 주요 요구 사항을 충족하면서도 확장성, 탄력성 및고가용성을 확보할 수 있습니다. 오늘날 여러 고객이 AWS, 파트너 및 오픈 소스 솔루션을 사용하여 비용을 절감하고 민첩성을 제고하면서도 글로벌 풋프린트를 축소하는 방식으로 RTC 워크로드를 실행하고 있습니다.

이 백서에 제공된 참조 아키텍처 및 모범 사례는 고객이 AWS에서 RTC 워크로드를 성공적으로 설정하고 최종 사용자 요구 사항을 충족하도록 솔루션을 최적화하는 동시에 클라우드 최적화하는 데 도움이 될 수 있습니다.

기여자

다음은 본 문서를 작성하는 데 도움을 준 개인 및 조직입니다.

- Ahmad Khan, 선임 솔루션스 아키텍트, Amazon Web Services
- Tipu Qureshi, 수석 엔지니어, AWS Support, Amazon Web Services
- Hasan Khan, 선임 기술 지원 관리자, Amazon Web Services
- Shoma Chakravarty, WW 기술 리더, Telecom, Amazon Web Services

문서 수정

이 백서의 업데이트에 대한 알림을 받으려면 RSS 피드를 구독하십시오.

업데이트 기록-변경	update-history-description	update-history-date
백서 업데이트됨	최신 서비스 및 기능이 업데이트되었습니다.	2020년 2월 13일
최초 게시	백서가 처음 게시되었습니다.	2018년 10월 1일

고지 사항

고객은 본 문서에 포함된 정보를 독자적으로 평가할 책임이 있습니다. 본 문서는 (a) 정보 제공만을 위한 것이며, (b) 사전 고지 없이 변경될 수 있는 현재의 AWS 제품 제공 서비스 및 사례를 보여 주며, (c) AWS 및 자회사, 공급업체 또는 라이선스 제공자로부터 어떠한 약정 또는 보증도 하지 않습니다. AWS 제품 또는 서비스는 명시적이든 묵시적이든 어떠한 종류의 보증, 진술 또는 조건 없이 '있는 그대로' 제공됩니다. 고객에 대한 AWS의 책임과 법적 책임은 AWS 계약서에 준하며 본 문서는 AWS와 고객 간의 계약에 포함되지 않고 계약을 변경하지도 않습니다.

© 2020 Amazon Web Services, Inc. 또는 자회사. All rights reserved.