AWS Whitepaper

Komunikasi Waktu Nyata di AWS

Komunikasi Waktu Nyata di AWS: AWS Whitepaper

Table of Contents

Komunikasi Real-Time di AWS

Praktik Terbaik untuk Merancang Beban Kerja Komunikasi Real-Time (RTC) yang Sangat Tersedia dan Dapat Diskalakan pada AWS

Tanggal publikasi: 5 Mei 2022 (Revisi dokumen)

Abstrak

Saat ini, banyak organisasi mencari untuk mengurangi biaya dan mencapai skalabilitas untuk suara real-time, pesan, dan beban kerja multimedia. Paper ini menguraikan praktik terbaik untuk mengelola beban kerja komunikasi real-time (RTC) di Amazon Web Services (AWS), dan menyertakan arsitektur referensi untuk memenuhi persyaratan ini. Paper ini berfungsi sebagai panduan bagi individu yang akrab dengan komunikasi real-time tentang cara mencapai ketersediaan dan skalabilitas yang tinggi untuk beban kerja ini.

Paper ini mencakup arsitektur referensi yang menunjukkan cara mengatur beban kerja RTC AWS, dan praktik terbaik untuk mengoptimalkan solusi guna memenuhi kebutuhan pengguna akhir sambil mengoptimalkan cloud. Evolved Packet Core (EPC) berada di luar cakupan untuk whitepaper ini, tetapi praktik terbaik yang dirinci di sini dapat diterapkan ke Virtual Network Functions (). VNFs

Apakah Anda sudah Well-Architected?

Kerangka Kerja AWS Well-Architected membantu Anda memahami pro dan kontra dari keputusan yang Anda buat saat membangun sistem di cloud. Enam pilar dari Kerangka Kerja ini memungkinkan Anda mempelajari praktik terbaik arsitektural untuk merancang dan mengoperasikan sistem yang andal, aman, efisien, hemat biaya, dan berkelanjutan. Menggunakan AWS Well-Architected Tool, tersedia tanpa biaya di AWS Management Console(login diperlukan), Anda dapat meninjau beban kerja Anda terhadap praktik terbaik ini dengan menjawab serangkaian pertanyaan untuk setiap pilar.

Untuk panduan lebih lanjut dari para ahli dan praktik terbaik untuk arsitektur cloud Anda—referensi penerapan arsitektur, diagram, dan laporan resmi—lihat Pusat Arsitektur AWS.

Pengantar

Aplikasi telekomunikasi yang menggunakan suara, video, dan pesan sebagai saluran merupakan persyaratan utama bagi banyak organisasi dan pengguna akhir mereka. Beban kerja komunikasi real-time (RTC) ini memiliki persyaratan latensi dan ketersediaan khusus yang dapat dipenuhi dengan mengikuti praktik terbaik desain yang relevan. Di masa lalu, beban kerja RTC telah digunakan di pusat data lokal tradisional dengan sumber daya khusus.

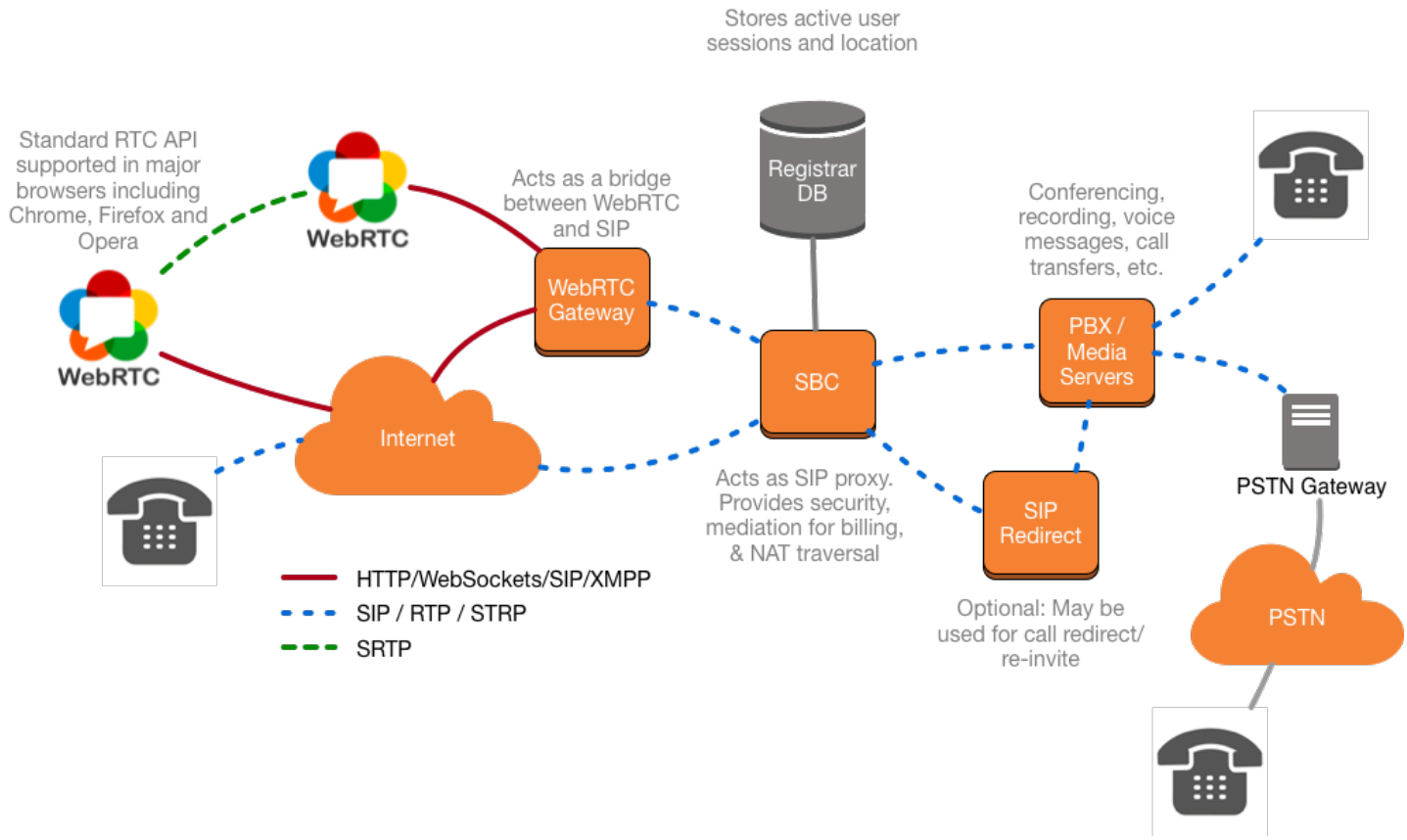
Beban kerja RTC membutuhkan lingkungan yang sangat skalabel, tangguh, dan tersedia. Saat ini, pelanggan menggunakan AWS untuk menjalankan beban kerja RTC dengan pengurangan biaya, peningkatan kelincahan, elastisitas, dan waktu ke pasar.

Komponen dasar arsitektur RTC

Dalam industri telekomunikasi, RTC umumnya mengacu pada sesi media langsung antara dua titik akhir dengan latensi minimum. Sesi ini dapat dikaitkan dengan:

- Sesi suara antara dua pihak (seperti sistem telepon, seluler, atau Voice over IP (VoIP))
- Pesan instan (seperti chatting dan Instant Relay Chat (IRC))
- Sesi video langsung (seperti konferensi video dan telepresence)

Masing-masing solusi sebelumnya memiliki beberapa komponen yang sama (seperti komponen yang menyediakan otentikasi, otorisasi dan kontrol akses, transcoding, buffering dan relay, dan sebagainya) dan beberapa komponen unik untuk jenis media yang ditransmisikan (seperti layanan siaran, server pesan dan antrian, dan sebagainya). Bagian ini berfokus pada mendefinisikan sistem RTC berbasis suara dan video dan semua komponen terkait, seperti yang diilustrasikan pada gambar berikut.



Komponen arsitektur penting untuk RTC

SoftSwitch/PBX

Softswitch atau PBX adalah otak dari sistem telepon suara dan memberikan kecerdasan untuk membangun, memelihara, dan merutekan panggilan suara di dalam atau di luar perusahaan dengan menggunakan komponen yang berbeda. Semua pelanggan perusahaan diharuskan mendaftar dengan softswitch untuk menerima atau melakukan panggilan. Fungsionalitas penting dari softswitch adalah untuk melacak setiap pelanggan dan bagaimana menjangkau mereka dengan menggunakan komponen lain dalam jaringan suara.

Pengontrol batas sesi (SBC)

Pengontrol batas sesi (SBC) berada di tepi jaringan suara dan melacak semua lalu lintas masuk dan keluar (baik pesawat kontrol maupun data). Salah satu tanggung jawab utama SBC adalah melindungi sistem suara dari penggunaan berbahaya. SBC dapat digunakan untuk interkoneksi dengan batang protokol inisiasi sesi (SIP) untuk konektivitas eksternal. Beberapa SBCs juga menyediakan kemampuan transcoding untuk mengkonversi CODECsdari satu format ke format lainnya. Sebagian besar SBCs juga menyediakan kemampuan traversal terjemahan alamat jaringan (NAT), yang membantu memastikan panggilan dibuat, bahkan di seluruh jaringan firewall.

Konektivitas PSTN

Solusi Voice over IP (VoIP) menggunakan gateway Public Switched Telephone Network (PSTN) dan batang SIP untuk terhubung dengan jaringan PSTN lama.

Gerbang PSTN

Gateway PSTN mengubah sinyal antara SIP dan media antara Real Time Transport Protocol (RTP) SS7 dan time division multiplexing (TDM) menggunakan transcoding CODEC. Gateway PSTN selalu berada di tepi dekat jaringan PSTN.

Batang SIP

Dalam trunk SIP, perusahaan tidak mengakhiri panggilannya ke jaringan TDM (SS7 berbasis), melainkan arus antara perusahaan dan telekomunikasi tetap melalui IP. Sebagian besar Batang SIP dibuat dengan menggunakan. SBCs Perusahaan harus menyetujui aturan keamanan yang telah ditentukan dari telekomunikasi, seperti mengizinkan rentang alamat IP tertentu, port, dan sebagainya.

Gerbang media (transcoder)

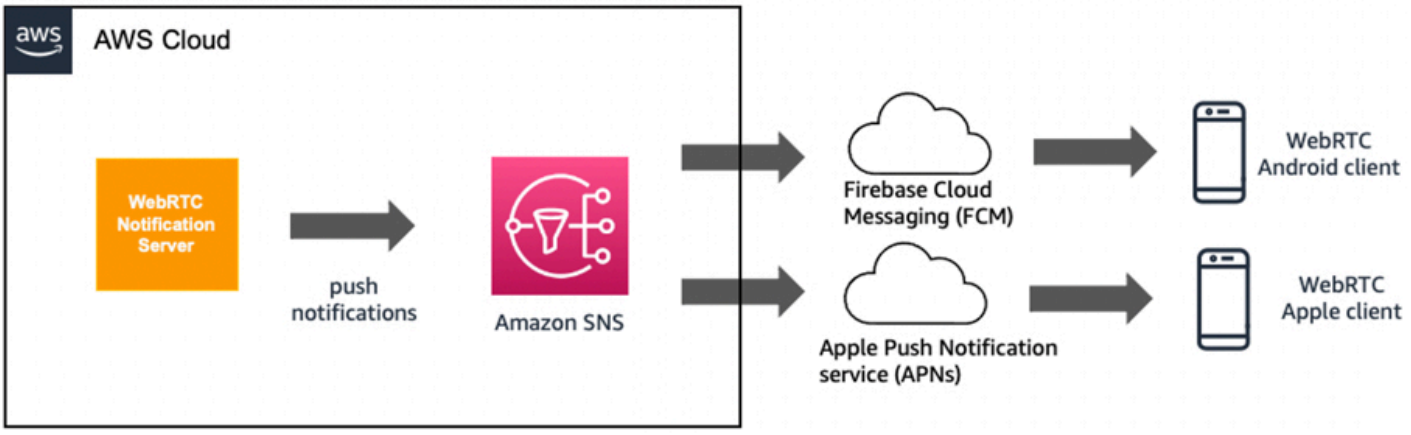
Pengguna berkomunikasi secara real-time menggunakan audio dan/atau video, serta data opsional dan informasi lainnya. Untuk berkomunikasi, kedua perangkat harus dapat menyetujui codec yang saling dipahami untuk setiap jalur media, sehingga mereka dapat berhasil berkomunikasi dan menyajikan media bersama. Semua browser yang kompatibel dengan WebRTC harus mendukung dukungan pengguna pemosisian online (OPUS) dan G711 untuk audio, VP8, dan profil Garis Dasar Terbatas H.264 untuk video.

Solusi suara khas di luar ekosistem WebRTC memungkinkan berbagai jenis. CODECs Beberapa yang umum CODECs adalah G.711 μ-law untuk Amerika Utara, G.711 A-law, G.729, dan G.722. Ketika dua perangkat yang menggunakan dua berbeda CODECs berkomunikasi satu sama lain, gateway media menerjemahkan aliran CODEC antara perangkat. Dengan kata lain, gateway media memproses media, dan memastikan bahwa perangkat akhir dapat berkomunikasi satu sama lain.

Pemberitahuan push di WebRTC

Implementasi WebRTC sangat umum di perangkat seluler. Tidak seperti browser web, perangkat seluler tidak dapat menjaga konektivitas websocket terbuka untuk waktu yang lama. Oleh karena itu, perlu mengandalkan notifikasi push dari server WebRTC untuk semua permintaan akhir, seperti panggilan dan pesan.

Amazon Simple Notification Service (Amazon SNS) memungkinkan Anda mengirim pemberitahuan push ke aplikasi di perangkat seluler. Aplikasi ini dapat berjalan di berbagai sistem operasi seperti Apple iOS atau Android. Gambar berikut menunjukkan ikhtisar tingkat tinggi aliran push-notifikasi, dari server notifikasi WebRTC ke endpoint seluler WebRTC.

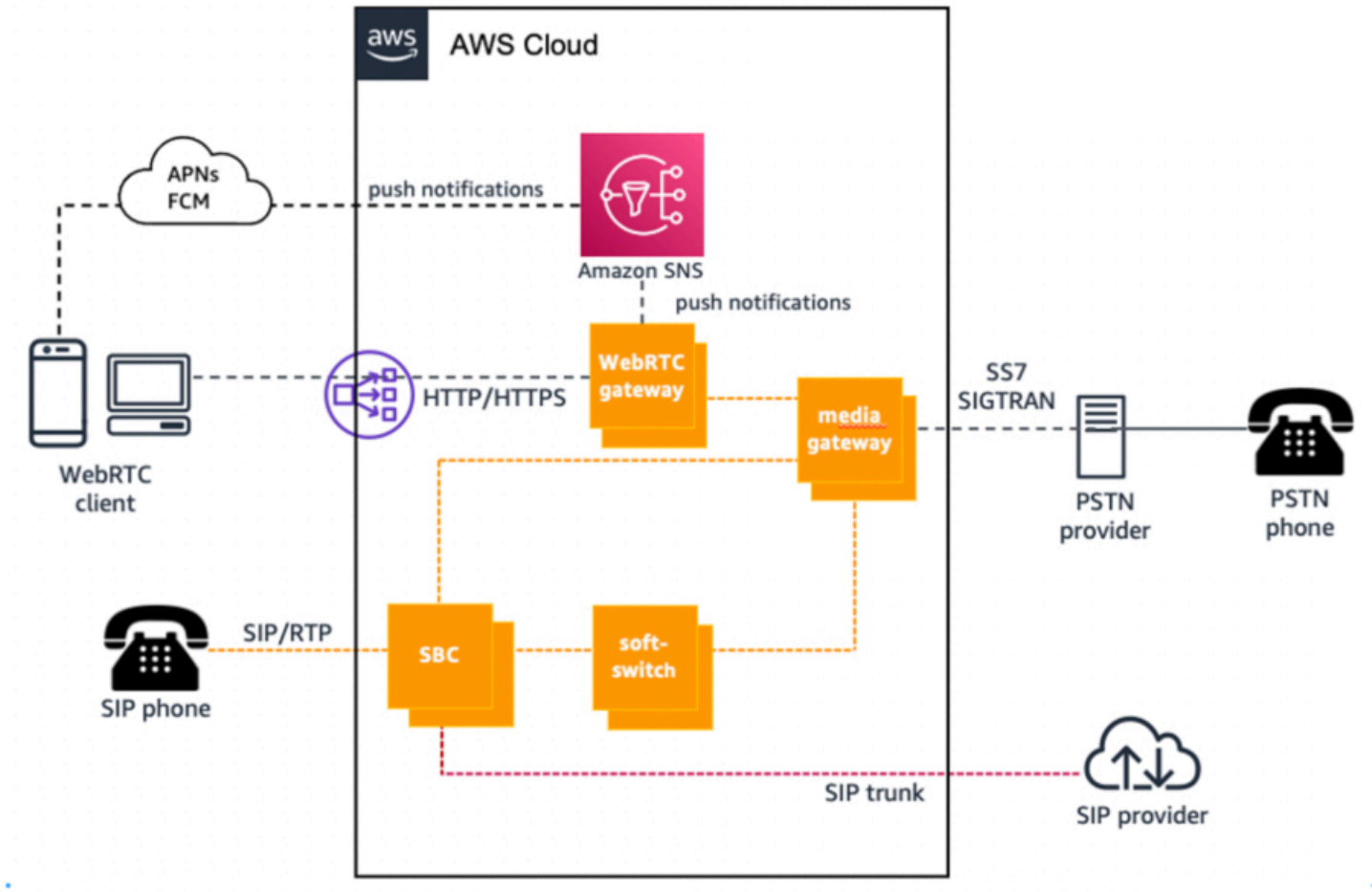


Amazon SNS untuk pemberitahuan push

WebRTC dan WebRTC gateway

Komunikasi real-time web (WebRTC) memungkinkan Anda untuk membuat panggilan dari browser web atau meminta sumber daya dari server backend dengan menggunakan API. Teknologi ini dirancang dengan teknologi cloud dalam pikiran dan karena itu menyediakan berbagai APIs yang dapat digunakan untuk membuat panggilan. Karena tidak semua solusi suara (termasuk SIP) mendukung ini APIs, gateway WebRTC diperlukan untuk menerjemahkan panggilan API ke pesan SIP dan sebaliknya.

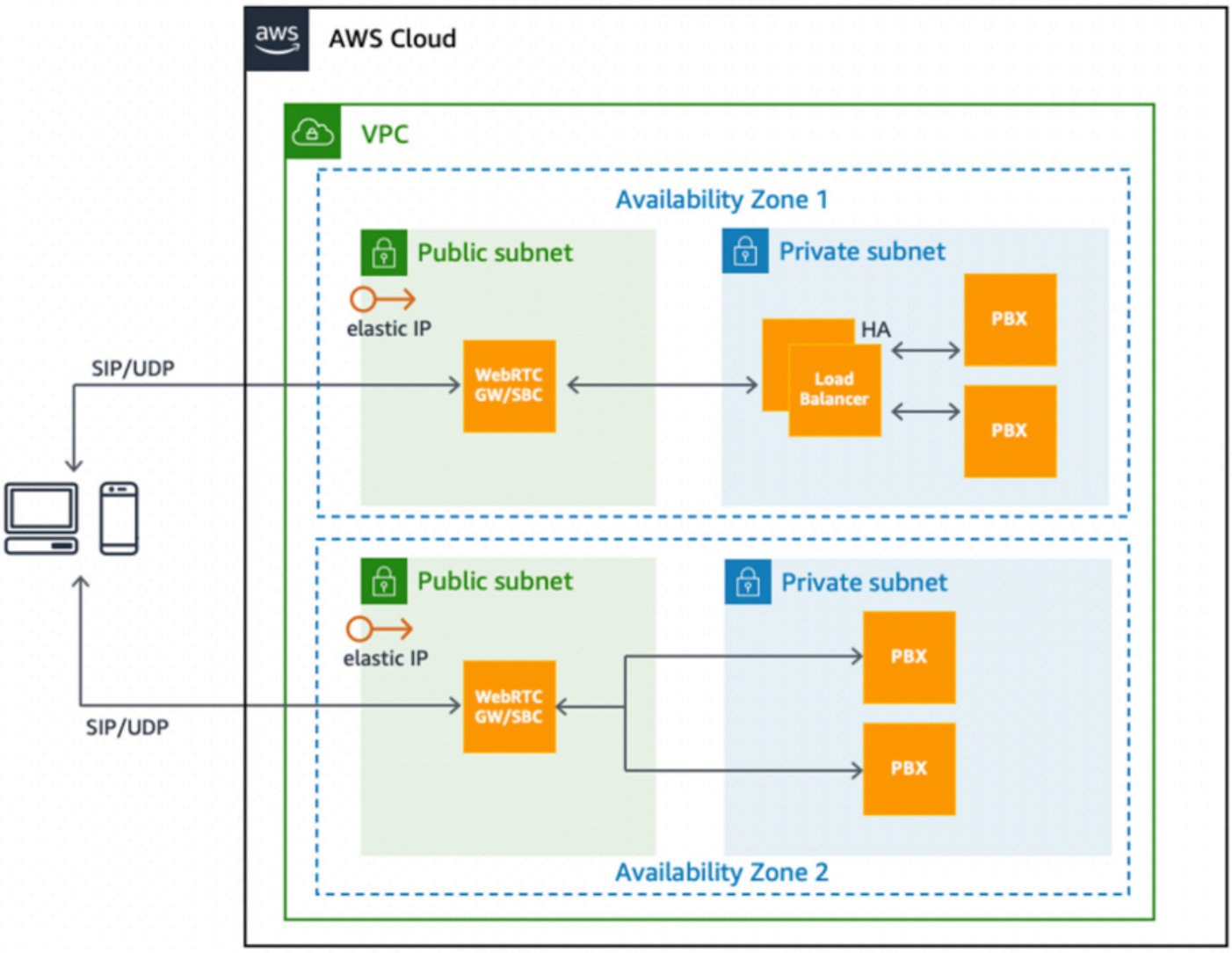
Gambar berikut menunjukkan pola desain untuk arsitektur WebRTC yang sangat tersedia. Lalu lintas masuk dari klien WebRTC diseimbangkan oleh Application Load Balancer (ALB) dengan WebRTC yang berjalan di instans Amazon Elastic Compute Cloud (Amazon) yang merupakan bagian dari grup Amazon Auto Scaling. EC2 EC2



Topologi dasar sistem RTC untuk suara

Pola desain lain untuk lalu lintas SIP dan RTP adalah menggunakan pasangan SBCs di Amazon EC2 dalam mode aktif-pasif di seluruh Availability Zones, seperti yang terlihat pada gambar berikut. Di

sini, alamat IP Elastis dapat dipindahkan secara dinamis antar instance setelah kegagalan, di mana Domain Name Service (DNS) tidak dapat digunakan.



Arsitektur RTC menggunakan Amazon EC2 di cloud pribadi virtual (VPC)

Ketersediaan dan skalabilitas tinggi pada AWS

Sebagian besar penyedia komunikasi real-time selaras dengan tingkat layanan yang menyediakan ketersediaan dari 99,9% hingga 99,999%. Bergantung pada tingkat ketersediaan tinggi (HA) yang Anda inginkan, Anda harus mengambil langkah-langkah yang semakin canggih di sepanjang siklus hidup penuh aplikasi. AWS merekomendasikan mengikuti pedoman ini untuk mencapai tingkat ketersediaan tinggi yang kuat:

- Rancang sistem agar tidak memiliki titik kegagalan tunggal. Gunakan pemantauan otomatis, deteksi kegagalan, dan mekanisme failover untuk komponen stateless dan stateful
  - Titik kegagalan tunggal (SPOF) biasanya dihilangkan dengan konfigurasi redundansi N+1 atau 2N, di mana N+1 dicapai melalui penyeimbangan beban di antara node aktif-aktif, dan 2N dicapai oleh sepasang node dalam konfigurasi siaga aktif.
  - AWS memiliki beberapa metode untuk mencapai HA melalui kedua pendekatan, seperti melalui cluster load balanced yang dapat diskalakan atau mengasumsikan pasangan siaga aktif.
- Instrumen dan ketersediaan sistem uji dengan benar.
- Mempersiapkan prosedur operasi untuk mekanisme manual untuk merespons, mengurangi, dan memulihkan dari kegagalan.

Bagian ini berfokus pada bagaimana mencapai tidak ada satu titik kegagalan menggunakan kemampuan yang tersedia di AWS. Secara khusus, bagian ini menjelaskan subset dari AWS kemampuan inti dan pola desain yang memungkinkan Anda membangun aplikasi komunikasi real-time yang sangat tersedia.

Pola IP mengambang untuk HA antara server stateful aktif — siaga

Pola desain IP mengambang adalah mekanisme yang terkenal untuk mencapai failover otomatis antara sepasang node perangkat keras aktif dan siaga (server media). Alamat IP virtual sekunder statis ditetapkan ke node aktif. Pemantauan berkelanjutan antara node aktif dan siaga mendeteksi kegagalan. Jika node aktif gagal, skrip pemantauan menetapkan IP virtual ke node siaga siap dan node siaga mengambil alih fungsi aktif utama. Dengan cara ini, IP virtual mengapung di antara node aktif dan siaga.

Penerapan dalam solusi RTC

Hal ini tidak selalu mungkin untuk memiliki beberapa instance aktif dari komponen yang sama dalam layanan, seperti cluster aktif-aktif dari N node. Konfigurasi siaga aktif menyediakan mekanisme terbaik untuk HA. Misalnya, komponen stateful dalam solusi RTC, seperti server media atau server konferensi, atau bahkan SBC atau server database, sangat cocok untuk pengaturan siaga aktif. Server SBC atau media memiliki beberapa sesi atau saluran yang berjalan lama yang aktif pada waktu tertentu, dan dalam kasus instans aktif SBC gagal, titik akhir dapat terhubung kembali ke node siaga tanpa konfigurasi sisi klien karena IP mengambang.

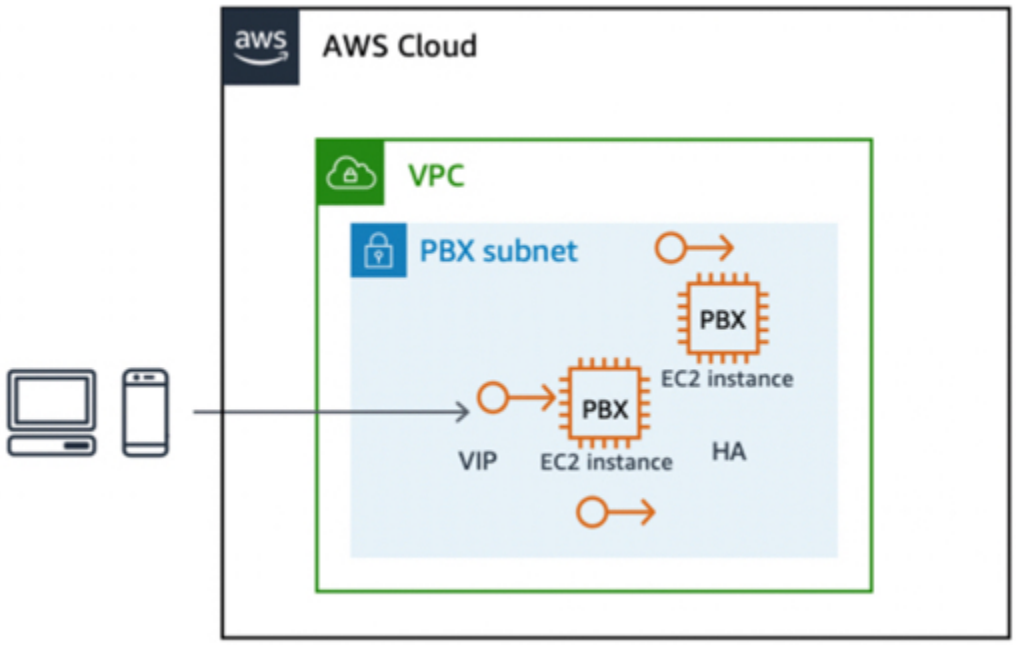
Implementasi pada AWS

Anda dapat menerapkan pola ini di AWS menggunakan kemampuan inti di Amazon Elastic Compute Cloud (Amazon EC2), Amazon EC2 API, alamat IP Elastis, dan dukungan di Amazon EC2 untuk alamat IP pribadi sekunder.

Untuk mengimplementasikan pola IP mengambang pada AWS:

1. Luncurkan dua EC2 instance untuk mengambil peran node primer dan sekunder, di mana primer diasumsikan dalam keadaan aktif secara default.
2. Tetapkan alamat IP pribadi sekunder tambahan ke EC2 instance utama.
3. Alamat IP elastis, yang mirip dengan IP virtual (VIP), dikaitkan dengan alamat pribadi sekunder. Alamat pribadi sekunder ini adalah alamat yang digunakan oleh endpoint eksternal untuk mengakses aplikasi.
4. Beberapa konfigurasi sistem operasi (OS) diperlukan untuk membuat alamat IP sekunder ditambahkan sebagai alias ke antarmuka jaringan utama.
5. Aplikasi harus mengikat alamat IP elastis ini. Dalam kasus perangkat lunak Asterisk, Anda dapat mengonfigurasi pengikatan melalui pengaturan SIP Asterisk lanjutan.
6. Jalankan skrip pemantauan—kustom, KeepAlive di Linux, Corosync, dan sebagainya—pada setiap node untuk memantau status peer node. Jika node aktif saat ini gagal, peer mendeteksi kegagalan ini, dan memanggil Amazon EC2 API untuk menetapkan kembali alamat IP pribadi sekunder ke dirinya sendiri.

   Oleh karena itu, aplikasi yang mendengarkan pada VIP yang terkait dengan alamat IP pribadi sekunder menjadi tersedia untuk titik akhir melalui node siaga.



Failover antara EC2 instance stateful menggunakan alamat IP elastis

Manfaat

Pendekatan ini adalah solusi anggaran rendah yang andal yang melindungi terhadap kegagalan di tingkat EC2 instans, infrastruktur, atau aplikasi.

Keterbatasan dan ekstensibilitas

Pola desain ini biasanya terbatas dalam satu Availability Zone. Ini dapat diimplementasikan di dua Availability Zone, tetapi dengan variasi. Dalam hal ini, alamat IP Floating Elastic diasosiasikan kembali antara node aktif dan siaga di Availability Zone yang berbeda melalui API alamat IP elastis reasosiasi ulang yang tersedia. Dalam implementasi failover yang ditunjukkan pada gambar sebelumnya, panggilan yang sedang berlangsung dihentikan dan titik akhir harus terhubung kembali. Dimungkinkan untuk memperluas implementasi ini dengan replikasi data sesi yang mendasarinya untuk memberikan failover sesi atau kontinuitas media yang mulus juga.

Load balancing untuk skalabilitas dan HA dengan WebRTC dan SIP

Load balancing sekelompok instance aktif berdasarkan aturan yang telah ditentukan, seperti round robin, afinitas atau latensi, dan sebagainya, adalah pola desain yang dipopulerkan secara luas oleh sifat stateless dari permintaan HTTP. Faktanya, load balancing adalah opsi yang layak jika banyak komponen aplikasi RTC.

Load balancer bertindak sebagai proxy terbalik atau titik masuk untuk permintaan ke aplikasi yang diinginkan, yang dengan sendirinya dikonfigurasi untuk berjalan di beberapa node aktif secara bersamaan. Pada titik waktu tertentu, penyeimbang beban mengarahkan permintaan pengguna ke salah satu node aktif di cluster yang ditentukan. Load balancer melakukan pemeriksaan kesehatan terhadap node di cluster target mereka dan tidak mengirim permintaan masuk ke node yang gagal dalam pemeriksaan kesehatan. Oleh karena itu, tingkat dasar ketersediaan tinggi dicapai dengan penyeimbangan beban. Juga, karena penyeimbang beban melakukan pemeriksaan kesehatan aktif dan pasif terhadap semua node cluster dalam interval sub-detik, waktu untuk failover hampir seketika.

Keputusan node mana yang akan diarahkan didasarkan pada aturan sistem yang ditentukan dalam penyeimbang beban, termasuk:

- Round robin
- Sesi atau afinitas IP, yang memastikan bahwa beberapa permintaan dalam sesi atau dari IP yang sama dikirim ke node yang sama di cluster
- Berbasis latensi
- Berbasis beban

Penerapan dalam arsitektur RTC

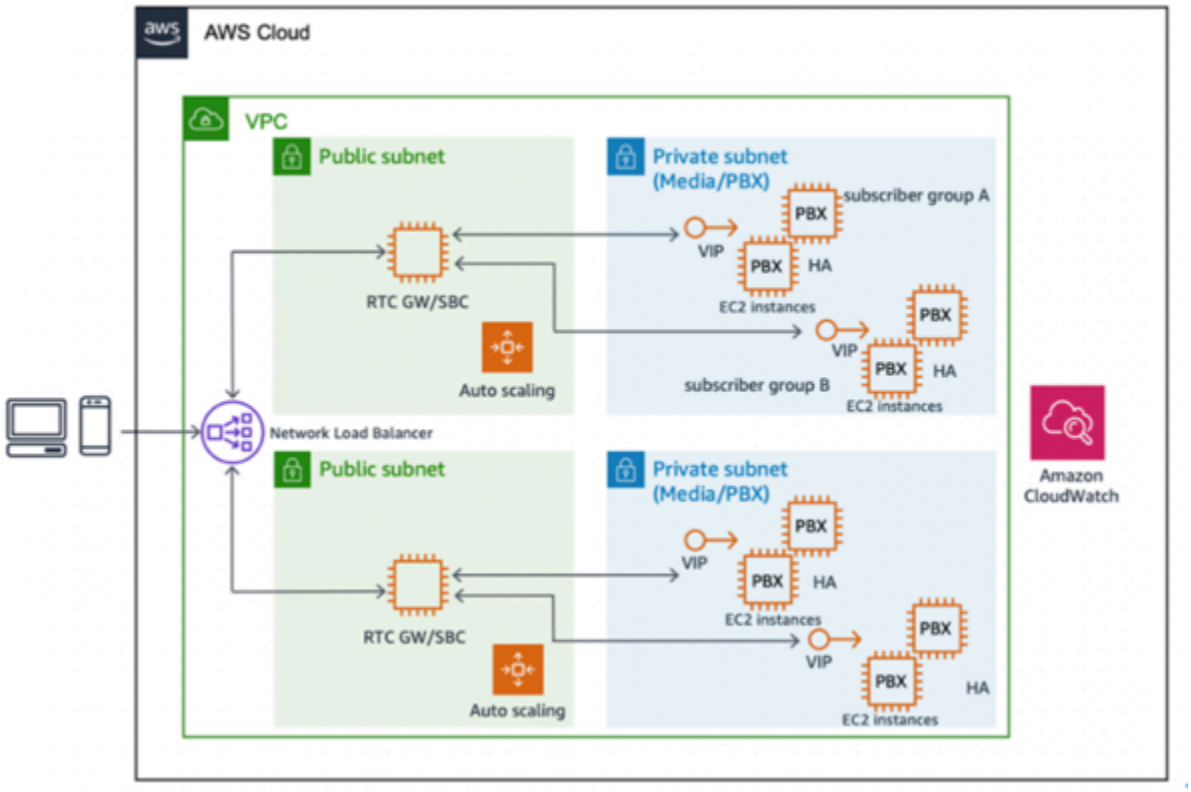
Protokol WebRTC memungkinkan WebRTC Gateways untuk dengan mudah memuat seimbang melalui penyeimbang beban berbasis HTTP, seperti Elastic Load Balancing (ELB), Application Load Balancer (ALB), atau Network Load Balancer (NLB). Dengan sebagian besar implementasi SIP yang mengandalkan transportasi melalui Transmission Control Protocol (TCP) dan User Datagram Protocol (UDP), Anda memerlukan penyeimbangan beban tingkat jaringan atau koneksi dengan dukungan untuk lalu lintas berbasis TCP dan UDP diperlukan.

Load balancing pada AWS WebRTC menggunakan Application Load Balancer dan Auto Scaling

Dalam kasus komunikasi berbasis WebRTC, Elastic Load Balancing menyediakan penyeimbang beban yang dikelola sepenuhnya, sangat tersedia, dan dapat diskalakan untuk berfungsi sebagai titik masuk permintaan, yang kemudian diarahkan ke kelompok target instance yang terkait dengan Elastic Load Balancing. EC2 Karena permintaan WebRTC bersifat stateless, Anda dapat menggunakan Amazon EC2 Auto Scaling, untuk menyediakan skalabilitas, elastisitas, dan ketersediaan tinggi yang sepenuhnya otomatis dan dapat dikontrol.

Application Load Balancer menyediakan layanan load balancing yang dikelola sepenuhnya yang sangat tersedia menggunakan beberapa Availability Zone, dan dapat diskalakan. Ini mendukung penyeimbangan beban WebSocket permintaan yang menangani pensinyalan untuk aplikasi WebRTC dan komunikasi dua arah antara klien dan server menggunakan koneksi TCP yang berjalan lama. Application Load Balancer juga mendukung perutean berbasis konten dan sesi lengket, merutekan permintaan dari klien yang sama ke target yang sama menggunakan cookie yang dihasilkan penyeimbang beban. Jika Anda mengaktifkan sesi lengket, target yang sama menerima permintaan dan dapat menggunakan cookie untuk memulihkan konteks sesi.

Gambar berikut menunjukkan topologi target.



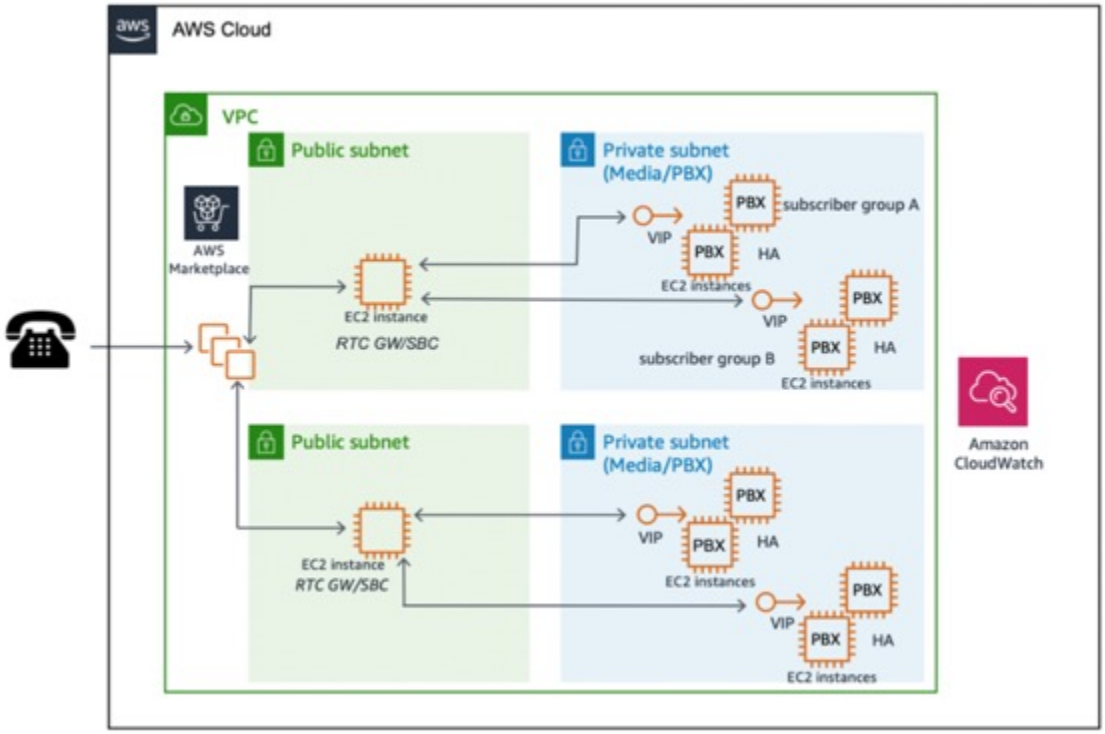
WebRTC skalabilitas dan arsitektur ketersediaan tinggi

Implementasi untuk SIP menggunakan Network Load Balancer atau produk AWS Marketplace

Dalam kasus komunikasi berbasis SIP, koneksi dibuat melalui TCP atau UDP, dengan sebagian besar aplikasi RTC menggunakan UDP. Jika SIP/TCP adalah protokol sinyal pilihan, maka layak untuk menggunakan Network Load Balancer untuk penyeimbangan beban yang dikelola sepenuhnya, sangat tersedia, skalabel, dan kinerja.

Network Load Balancer beroperasi pada tingkat koneksi (Lapisan empat), merutekan koneksi ke target seperti EC2 instans Amazon, kontainer, dan alamat IP berdasarkan data protokol IP. Ideal untuk penyeimbangan beban lalu lintas TCP atau UDP, load balancing jaringan mampu menangani jutaan permintaan per detik sambil mempertahankan latensi ultra-rendah. Ini terintegrasi dengan layanan AWS populer lainnya, seperti Amazon EC2 Auto Scaling, Amazon Elastic Container Service (Amazon ECS), Amazon Elastic Kubernetes Service (Amazon EKS) dan. AWS CloudFormation

Jika koneksi SIP dimulai, opsi lain adalah menggunakan off-the-shelf perangkat lunak AWS Marketplacekomersial (COTS). AWS Marketplace Menawarkan banyak produk yang dapat menangani UDP dan jenis penyeimbangan beban koneksi lapisan empat lainnya. COTS biasanya mencakup dukungan untuk ketersediaan tinggi dan umumnya terintegrasi dengan fitur, seperti Amazon EC2 Auto Scaling, untuk lebih meningkatkan ketersediaan dan skalabilitas. Gambar berikut menunjukkan topologi target:



Skalabilitas RTC berbasis SIP dengan produk AWS Marketplace

Load balancing dan failover berbasis DNS Lintas Wilayah

Amazon Route 53 menyediakan layanan DNS global yang dapat digunakan sebagai titik akhir publik atau pribadi bagi klien RTC untuk mendaftar dan terhubung dengan aplikasi media. Dengan Amazon Route 53, pemeriksaan kesehatan DNS dapat dikonfigurasi untuk mengarahkan lalu lintas ke titik akhir yang sehat atau untuk memantau kesehatan aplikasi Anda secara independen.

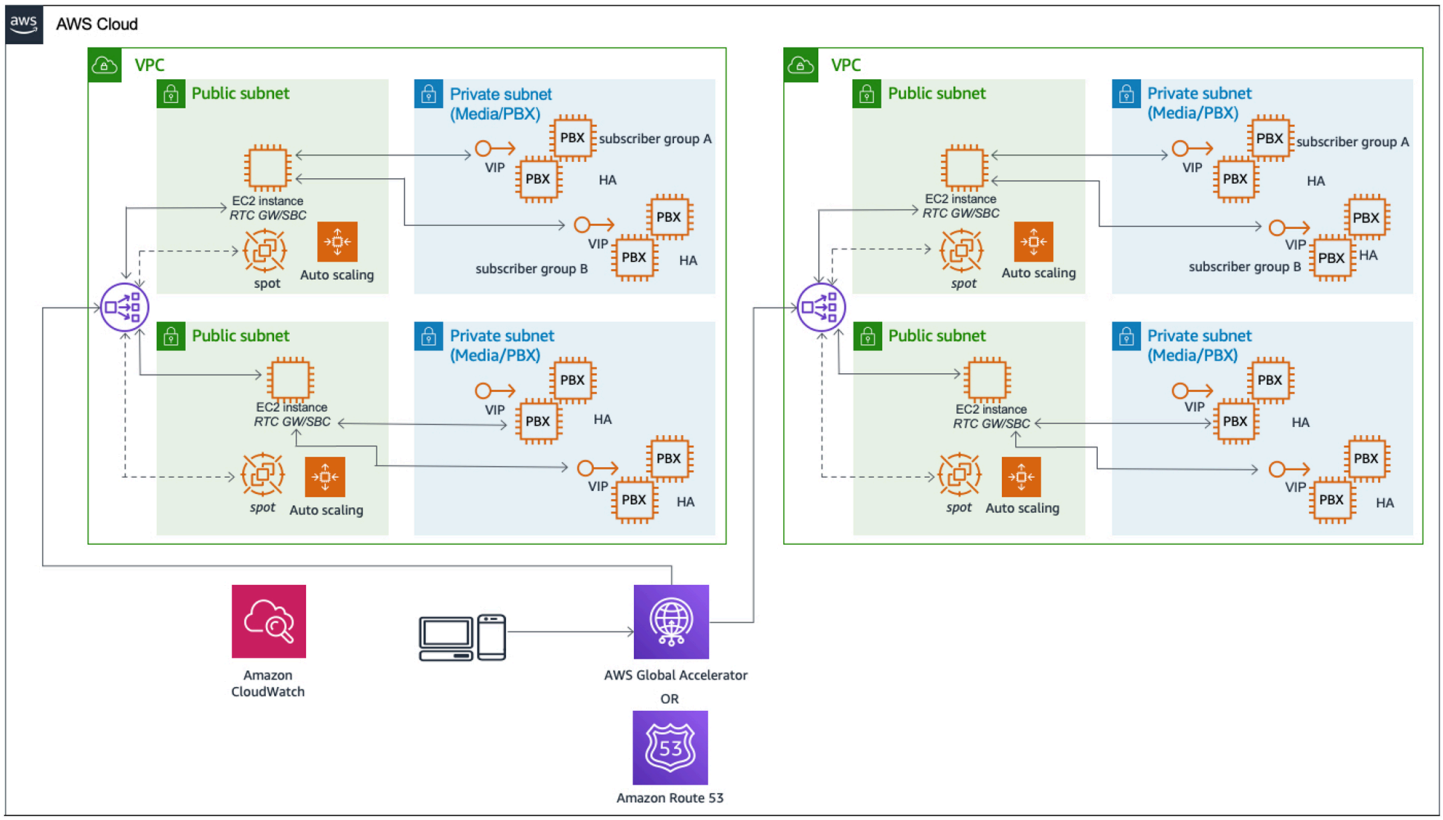
Fitur Amazon Route 53 Traffic Flow memudahkan Anda mengelola lalu lintas secara global melalui berbagai jenis perutean, termasuk perutean berbasis latensi, DNS geo, geoproximity, dan round robin berbobot—yang semuanya dapat digabungkan dengan DNS Failover untuk mengaktifkan berbagai arsitektur latensi rendah dan toleran kesalahan. Editor visual sederhana Amazon Route 53 Traffic Flow memungkinkan Anda mengelola cara pengguna akhir diarahkan ke titik akhir aplikasi Anda—baik dalam satu Wilayah AWS atau didistribusikan di seluruh dunia.

Dalam kasus penyebaran global, kebijakan perutean berbasis latensi di Route 53 sangat berguna untuk mengarahkan pelanggan ke titik kehadiran terdekat untuk server media untuk meningkatkan kualitas layanan yang terkait dengan pertukaran media real-time.

Perhatikan bahwa untuk menerapkan failover ke alamat DNS baru, cache klien harus di-flush. Selain itu, perubahan DNS mungkin memiliki kelambatan karena disebarkan di seluruh server DNS global. Anda dapat mengelola interval penyegaran untuk pencarian DNS dengan atribut Time to Live. Atribut ini dapat dikonfigurasi pada saat menyiapkan kebijakan DNS.

Untuk menjangkau pengguna global dengan cepat atau untuk memenuhi persyaratan menggunakan IP publik tunggal, juga AWS Global Accelerator dapat digunakan untuk failover lintas wilayah. AWS Global Acceleratoradalah layanan jaringan yang meningkatkan ketersediaan dan kinerja untuk aplikasi dengan jangkauan lokal dan global. AWS Global Accelerator menyediakan alamat IP statis yang bertindak sebagai titik masuk tetap ke titik akhir aplikasi Anda, seperti Application Load Balancers, Network Load Balancers, atau instans EC2 Amazon dalam satu atau beberapa Wilayah AWS. Ini menggunakan jaringan global AWS untuk mengoptimalkan jalur dari pengguna Anda ke aplikasi Anda, meningkatkan kinerja, seperti latensi lalu lintas TCP dan UDP Anda.

AWS Global Accelerator terus memantau kesehatan titik akhir aplikasi Anda, dan secara otomatis mengalihkan lalu lintas ke titik akhir sehat terdekat jika titik akhir saat ini berubah menjadi tidak sehat. Untuk persyaratan keamanan tambahan, Accelerated Site-to-Site VPN digunakan AWS Global Accelerator untuk meningkatkan kinerja koneksi VPN dengan merutekan lalu lintas secara cerdas melalui AWS Global Network dan lokasi AWS edge.



Desain ketersediaan tinggi Antar Wilayah menggunakan AWS Global Accelerator atau Amazon Route 53

Daya tahan data dan HA dengan penyimpanan persisten

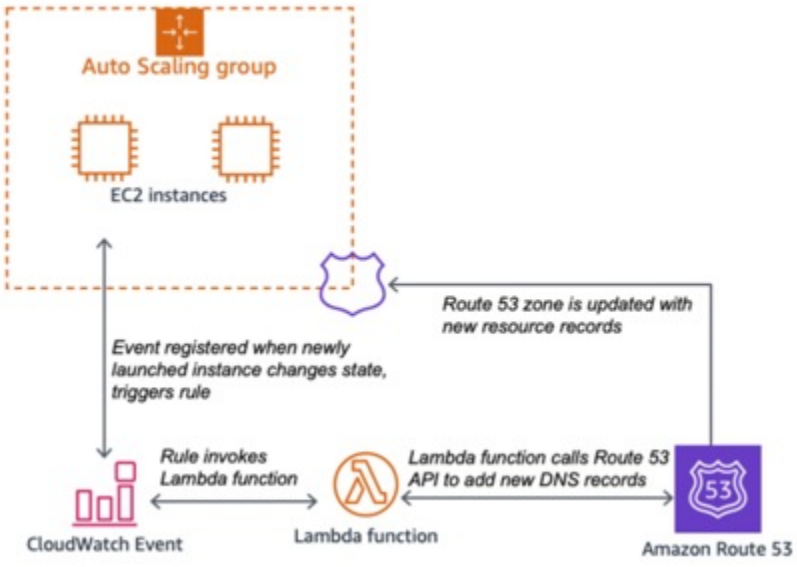
Sebagian besar aplikasi RTC mengandalkan penyimpanan persisten untuk menyimpan dan mengakses data untuk otentikasi, otorisasi, akuntansi (data sesi, catatan detail panggilan, dll.), pemantauan operasional, dan pencatatan. Di pusat data tradisional, memastikan ketersediaan dan daya tahan tinggi untuk komponen penyimpanan persisten (database, sistem file, dan sebagainya) biasanya memerlukan pengangkatan berat melalui pengaturan jaringan area penyimpanan (SAN), Redundant Array of Independent Disks (RAID) desain, dan proses untuk pencadangan, pemulihan, dan pemrosesan failover. Ini AWS Cloud sangat menyederhanakan dan meningkatkan praktik pusat data tradisional seputar daya tahan dan ketersediaan data.

Untuk penyimpanan objek dan penyimpanan file, AWS layanan seperti Amazon Simple Storage Service (Amazon S3) dan Amazon Elastic File System (Amazon EFS) menyediakan ketersediaan dan skalabilitas tinggi yang dikelola. Amazon S3 memiliki daya tahan data 99,999999999% (11 sembilan).

Untuk penyimpanan data transaksional, pelanggan memiliki opsi untuk memanfaatkan Amazon Relational Database Service (Amazon RDS) yang dikelola sepenuhnya yang mendukung Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle, dan Microsoft SQL Server dengan penerapan ketersediaan tinggi. Untuk fungsi registrar, profil pelanggan, atau penyimpanan catatan akuntansi (seperti CDRs), Amazon RDS menyediakan opsi yang toleran terhadap kesalahan, sangat tersedia, dan dapat diskalakan.

Penskalaan dinamis dengan AWS Lambda, Amazon Route 53, dan Amazon EC2 Auto Scaling

AWS memungkinkan rangkaian fitur dan kemampuan untuk menggabungkan fungsi tanpa server khusus sebagai layanan berdasarkan peristiwa infrastruktur. Salah satu pola desain yang memiliki banyak kegunaan serbaguna dalam aplikasi RTC adalah kombinasi kait siklus hidup penskalaan otomatis dengan Amazon Events CloudWatch , Amazon Route 53, dan fungsi. AWS Lambda AWS Lambda fungsi dapat menanamkan tindakan atau logika apa pun. Gambar berikut menunjukkan bagaimana fitur-fitur ini dirantai bersama dapat meningkatkan keandalan dan skalabilitas sistem dengan otomatisasi.



Penskalaan otomatis dengan pembaruan dinamis ke Amazon Route 53

WebRTC yang sangat tersedia dengan Amazon Kinesis Video Streams

Amazon Kinesis Video Streams menawarkan streaming media real-time melalui WebRTC, memungkinkan pengguna untuk menangkap, memproses, dan menyimpan aliran media untuk pemutaran, analitik, dan pembelajaran mesin. Aliran ini sangat tersedia, dapat diskalakan, dan sesuai dengan standar WebRTC. Amazon Kinesis Video Streams menyertakan titik akhir pensinyalan WebRTC untuk penemuan rekan yang cepat dan pembuatan koneksi yang aman. Ini termasuk Utilitas Penjelajahan Sesi yang dikelola untuk NAT (STUN) dan Traversal Menggunakan Relay di sekitar titik akhir NAT (TURN) untuk pertukaran media secara real-time antara rekan-rekan. Ini juga mencakup SDK open-source gratis yang secara langsung terintegrasi dengan firmware kamera untuk memungkinkan komunikasi yang aman dengan titik akhir Amazon Kinesis Video Streams, memungkinkan penemuan rekan dan streaming media. Terakhir, ia menyediakan pustaka klien untuk Android, iOS, dan JavaScript yang memungkinkan pemutar seluler dan web yang sesuai dengan WebRTC untuk menemukan dan terhubung dengan aman dengan perangkat kamera untuk streaming media dan komunikasi dua arah.

Trunking SIP yang sangat tersedia dengan Konektor Suara Amazon Chime

Amazon Chime Voice Connector memberikan layanan trunking pay-as-you-go SIP yang memungkinkan perusahaan untuk membuat dan/atau menerima panggilan telepon yang aman dan murah dengan sistem telepon mereka. Amazon Chime Voice Connector adalah alternatif berbiaya rendah untuk penyedia layanan SIP trunks atau Integrated Services Digital Network (ISDN) Primary Rate Interfaces (). PRIs Pelanggan memiliki opsi untuk mengaktifkan panggilan masuk, panggilan keluar, atau keduanya.

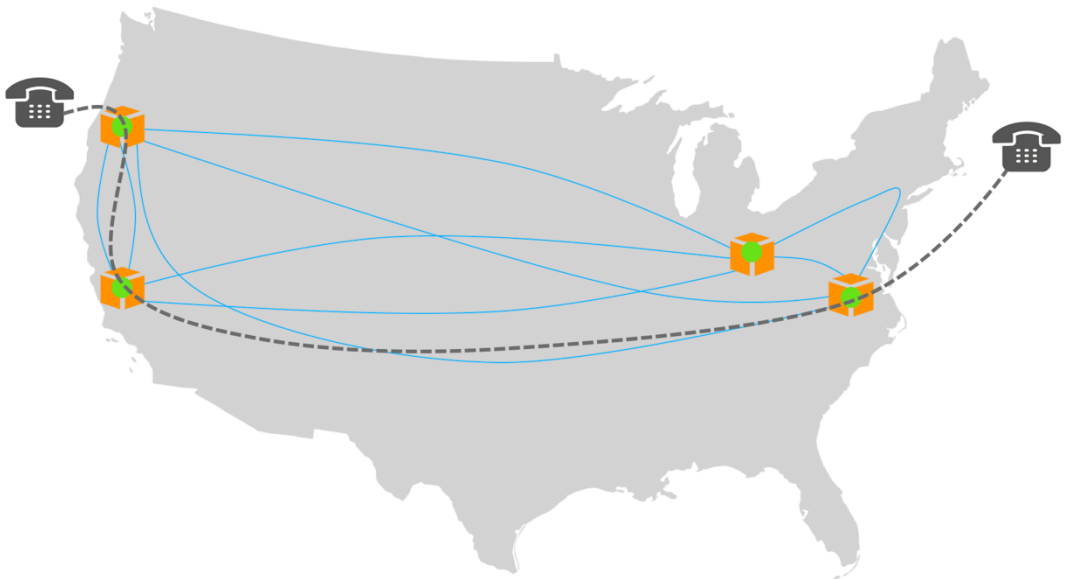
Layanan ini menggunakan AWS jaringan untuk memberikan pengalaman panggilan yang sangat tersedia di beberapa Wilayah AWS. Anda dapat melakukan streaming audio dari panggilan telepon trunking SIP, atau umpan rekaman media berbasis SIP (SIPREC) yang diteruskan ke Amazon Kinesis Video Streams untuk mendapatkan wawasan dari panggilan bisnis secara real time. Anda dapat dengan cepat membangun aplikasi untuk analisis audio melalui integrasi dengan Amazon Transcribe dan pustaka pembelajaran mesin umum lainnya.

Praktik terbaik dari lapangan

Bagian ini merangkum praktik terbaik yang telah diterapkan oleh beberapa AWS pelanggan terbesar dan paling sukses yang menjalankan beban kerja Protokol Inisiasi Sesi (SIP) real-time yang besar. AWS pelanggan yang ingin menjalankan infrastruktur SIP mereka sendiri di cloud publik akan menemukan praktik terbaik ini berharga karena dapat membantu meningkatkan keandalan dan ketahanan sistem jika terjadi berbagai jenis kegagalan. Meskipun beberapa praktik terbaik ini spesifik SIP, kebanyakan dari mereka berlaku untuk aplikasi komunikasi real-time yang berjalan AWS.

Buat overlay SIP

AWS memiliki tulang punggung jaringan yang kuat, terukur, dan redundan yang menyediakan konektivitas antara yang berbeda. Wilayah AWS Ketika peristiwa jaringan, seperti pemotongan serat, menurunkan tautan AWS tulang punggung, lalu lintas dengan cepat gagal ke jalur redundan menggunakan protokol perutean tingkat jaringan, seperti Border Gateway Protocol (BGP). Rekayasa lalu lintas tingkat jaringan ini adalah kotak hitam bagi AWS pelanggan dan sebagian besar bahkan tidak memperhatikan peristiwa failover ini. Namun, pelanggan yang menjalankan beban kerja real-time, seperti suara, video berkualitas tinggi, dan pesan latensi rendah, terkadang memperhatikan peristiwa ini. Jadi, bagaimana AWS pelanggan dapat menerapkan teknik lalu lintas mereka sendiri di atas apa yang disediakan oleh AWS di tingkat jaringan? Solusinya adalah menyebarkan infrastruktur SIP di banyak hal yang berbeda Wilayah AWS. Sebagai bagian dari fitur kontrol panggilan, SIP juga menyediakan kemampuan untuk merutekan panggilan melalui proxy SIP tertentu.

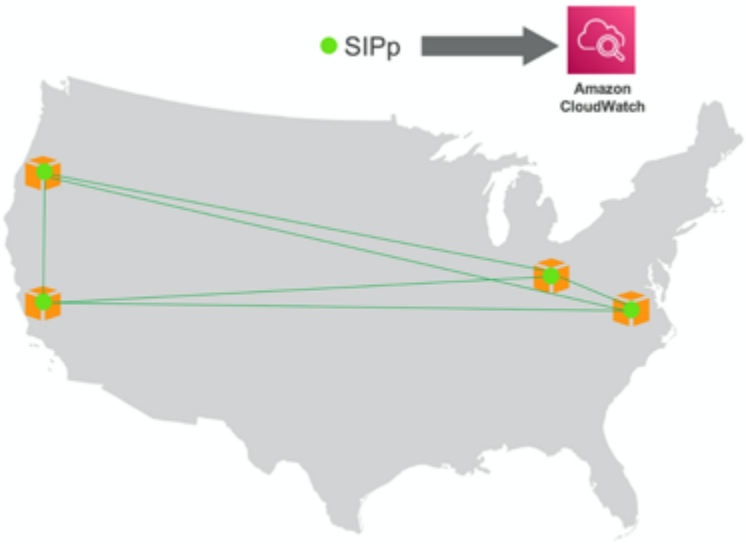


Menggunakan perutean SIP untuk mengganti perutean jaringan

Pada gambar sebelumnya, infrastruktur SIP (diwakili oleh titik-titik hijau di dalam kubus) berjalan di keempat Wilayah AS. Garis biru solid mewakili penggambaran fiksi tulang punggung. AWS Jika tidak ada perutean SIP yang diterapkan, panggilan yang berasal dari pantai barat AS dan ditujukan untuk pantai timur AS melewati tautan tulang punggung yang secara langsung menghubungkan wilayah Oregon dan Virginia. Diagram menunjukkan bagaimana pelanggan dapat mengganti perutean tingkat jaringan dan membuat panggilan yang sama antara Oregon dan Virginia yang dirutekan melalui California menggunakan perutean SIP. Jenis teknik lalu lintas SIP ini dapat diimplementasikan menggunakan proxy SIP dan gateway media berdasarkan metrik jaringan seperti transmisi ulang SIP dan preferensi bisnis khusus pelanggan.

Lakukan pemantauan terperinci

Pengguna akhir aplikasi suara dan video real-time mengharapkan tingkat kinerja yang sama seperti yang mereka capai dengan layanan telepon tradisional. Jadi, ketika mereka mengalami masalah dengan aplikasi, itu akhirnya merusak reputasi penyedia. Untuk menjadi proaktif daripada reaktif, sangat penting bahwa pemantauan rinci diterapkan di setiap bagian dari sistem yang melayani pengguna akhir.



Menggunakan SIPp untuk memantau infrastruktur VoIP

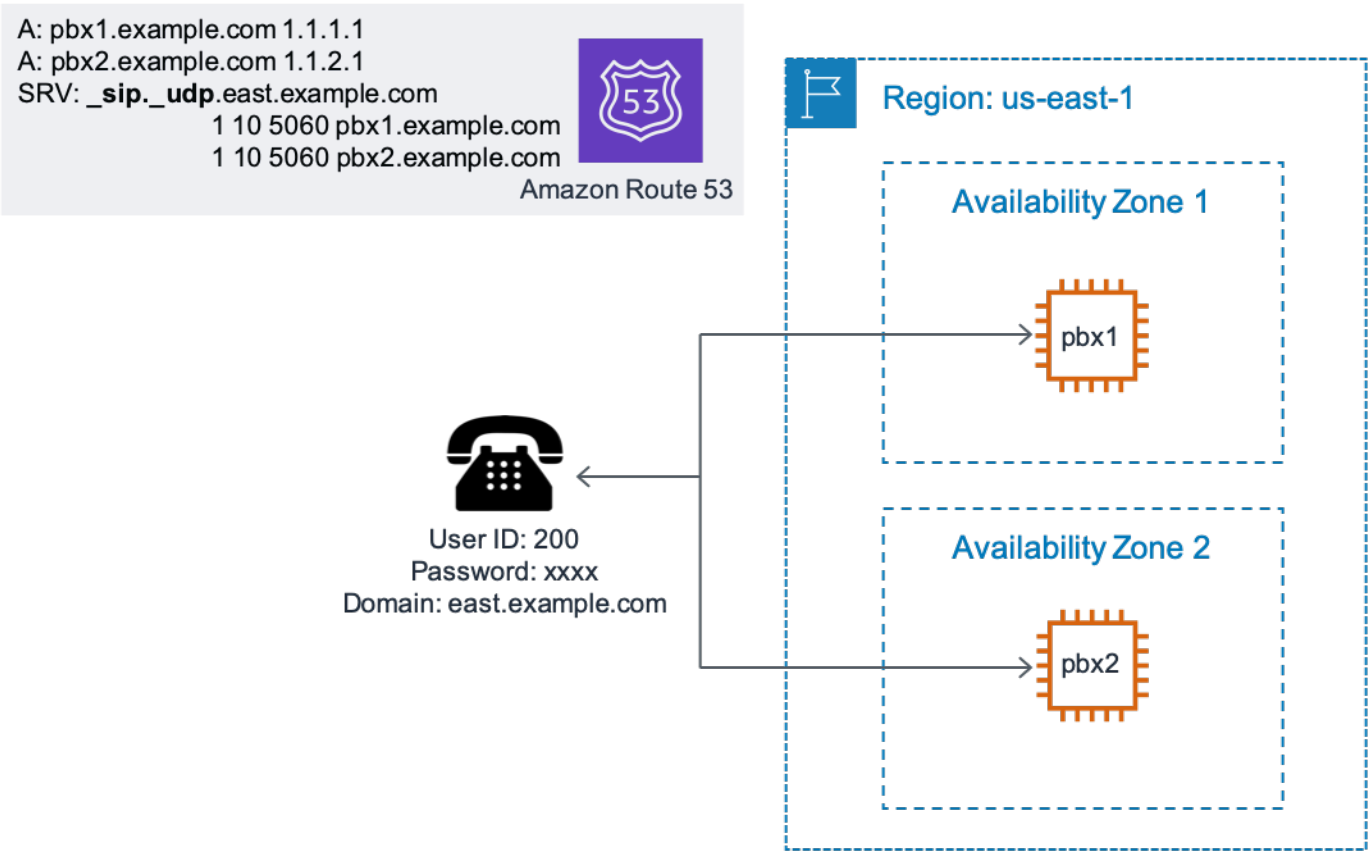
Banyak alat open source, seperti iPerf atau SIPp, dan VOIPMonitor, tersedia untuk digunakan dalam memantau lalu lintas SIP/RTP. Dalam contoh sebelumnya, node yang menjalankan SIP dalam mode klien dan server mengukur metrik SIP seperti Panggilan Sukses dan Transmisi Ulang SIP antara keempat AS. Wilayah AWS Metrik ini kemudian dapat diekspor ke Amazon CloudWatch

menggunakan skrip khusus. Dengan menggunakan CloudWatch, pelanggan dapat membuat alarm pada metrik khusus ini berdasarkan nilai ambang tertentu. Tindakan remediasi otomatis atau manual kemudian dapat diambil berdasarkan keadaan CloudWatch alarm ini.

Bagi pelanggan yang tidak ingin mengalokasikan sumber daya teknik yang diperlukan untuk mengembangkan dan memelihara sistem pemantauan khusus, banyak solusi pemantauan VoIP yang baik tersedia di pasar, seperti. ThousandEyes Contoh tindakan remediasi adalah mengubah perutean SIP berdasarkan peningkatan transmisi ulang SIP.

Gunakan DNS untuk load balancing dan floating IPs untuk failover

Klien IP telephony yang mendukung kemampuan DNS SRV dapat secara efisien menggunakan redundansi yang dibangun ke dalam infrastruktur dengan load balancing klien ke berbagai/. SBCs PBXs



Menggunakan catatan DNS SRV untuk memuat keseimbangan klien SIP

Angka sebelumnya menunjukkan bagaimana pelanggan dapat menggunakan catatan SRV untuk memuat keseimbangan lalu lintas SIP. Setiap klien telepon IP yang mendukung standar SRV akan

mencari SIP. `<transport protocol>`awalan dalam catatan DNS tipe SRV. Dalam contoh, bagian jawaban dari DNS berisi kedua yang PBXs berjalan di AWS Availability Zone yang berbeda. Namun, selain titik akhir URIs, catatan SRV berisi tiga informasi tambahan:

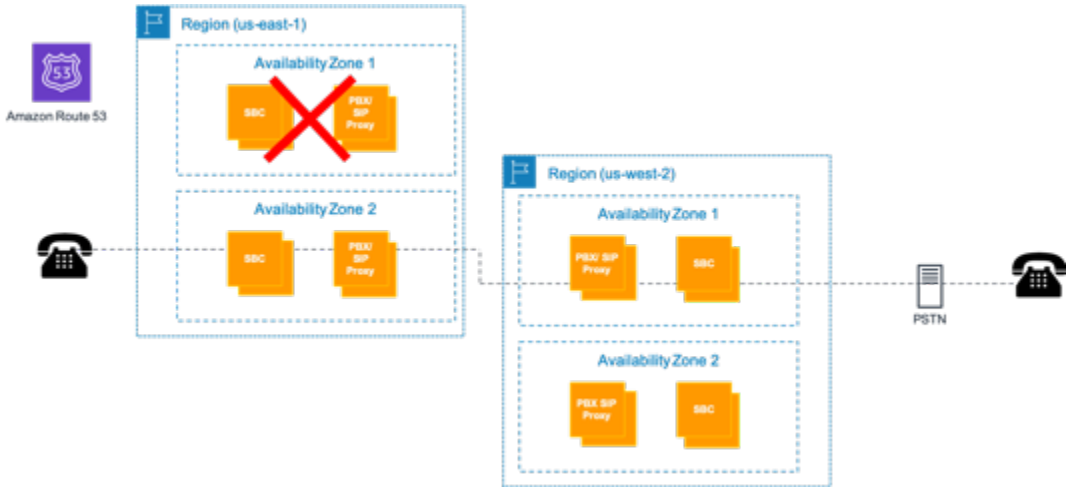
- Angka pertama adalah Prioritas (1 pada contoh di atas). Prioritas yang lebih rendah lebih disukai daripada yang lebih tinggi.
- Angka kedua adalah Berat (10 pada contoh di atas).
- Dan nomor ketiga adalah Port yang akan digunakan (5060).

Karena prioritasnya sama (1) untuk kedua PBXs server, klien menggunakan bobot untuk memuat keseimbangan antara keduanya PBXs. Dalam hal ini, karena bobotnya sama, lalu lintas SIP harus memuat seimbang antara keduanya PBXs.

DNS dapat menjadi solusi yang baik untuk penyeimbangan beban klien, tetapi bagaimana dengan menerapkan failover dengan mengubah/memperbarui catatan DNS 'A'? Metode ini tidak disarankan karena ketidakkonsistenan yang ditemukan dalam perilaku caching DNS dalam klien dan node perantara. Pendekatan yang lebih baik untuk failover intra-AZ antara sekelompok node SIP adalah dengan menggunakan penggantian EC2 IP di mana alamat IP host yang terganggu langsung dipindahkan ke host yang sehat dengan menggunakan API. EC2 Dipasangkan dengan pemantauan terperinci dan solusi pemeriksaan kesehatan, penugasan kembali IP dari node yang gagal memastikan bahwa lalu lintas dipindahkan ke host yang sehat pada waktu yang tepat yang meminimalkan gangguan pengguna akhir.

Gunakan beberapa Availability Zone

Masing-masing Wilayah AWS dibagi lagi menjadi Availability Zone yang terpisah. Setiap Availability Zone memiliki daya, pendinginan, dan konektivitas jaringannya sendiri dan dengan demikian membentuk domain kegagalan yang terisolasi. Dalam konstruksi AWS, pelanggan didorong untuk menjalankan beban kerja mereka di lebih dari satu Availability Zone. Ini memastikan bahwa aplikasi pelanggan dapat menahan bahkan kegagalan Availability Zone lengkap - peristiwa yang sangat langka dalam dirinya sendiri. Rekomendasi ini juga merupakan singkatan dari infrastruktur SIP real-time.



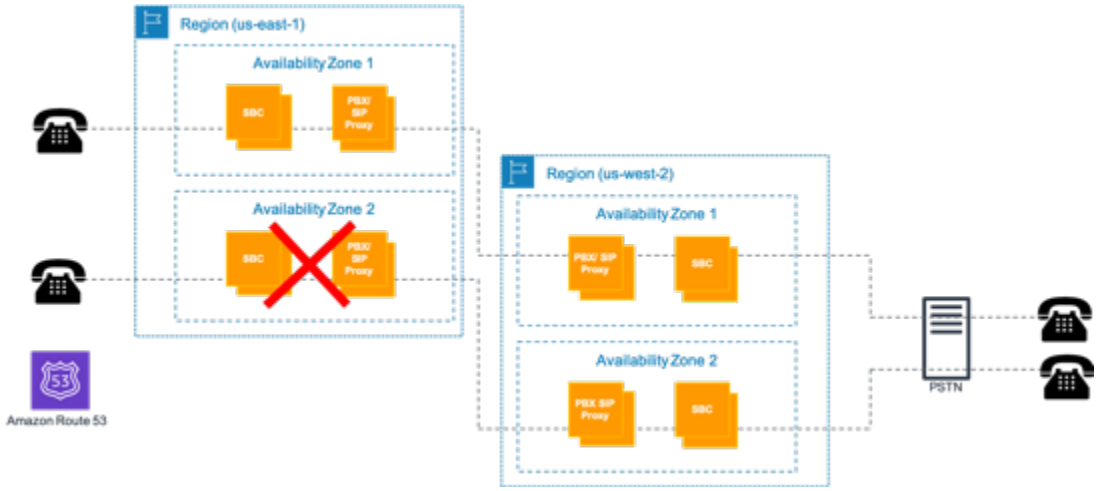
Menangani kegagalan Availability Zone

Misalkan peristiwa bencana (seperti badai kategori lima) menyebabkan pemadaman Zona Ketersediaan lengkap di Wilayah us-east-1. Dengan infrastruktur yang berjalan seperti yang ditunjukkan pada diagram, semua klien SIP yang awalnya terdaftar dengan node di Availability Zone yang gagal harus mendaftar ulang dengan node SIP yang berjalan di Availability Zone #2. (Uji perilaku ini dengan klien/ponsel SIP Anda untuk memastikannya didukung.) Meskipun panggilan SIP aktif pada saat pemadaman Availability Zone hilang, setiap panggilan baru dialihkan melalui Availability Zone 2.

Untuk meringkas, catatan DNS SRV harus mengarahkan klien ke beberapa catatan 'A', satu di setiap Availability Zone. Masing-masing catatan 'A' tersebut harus, pada gilirannya, menunjuk ke beberapa alamat IP SBCs/PBXs di Availability Zone yang menyediakan ketahanan Zona intra dan Inter-availability. Failover Zona intra- dan Inter-availability dapat diimplementasikan dengan menggunakan penggantian IP jika bersifat publik. IPs Private IPs, bagaimanapun, tidak dapat dipindahkan di seluruh Availability Zone. Jika pelanggan menggunakan alamat IP pribadi, maka mereka harus bergantung pada klien SIP yang mendaftar ulang dengan cadangan SBC/PBX untuk failover Inter-availability Zone.

Simpan lalu lintas dalam satu Availability Zone dan gunakan grup EC2 penempatan

Juga dikenal sebagai Availability Zone Affinity, praktik terbaik ini juga berlaku untuk peristiwa langka kegagalan Availability Zone yang lengkap. Disarankan agar Anda menghilangkan lalu lintas lintas lintas AZ sehingga lalu lintas SIP atau RTP apa pun yang memasuki satu Availability Zone harus tetap berada di Availability Zone tersebut sampai keluar dari Wilayah.



Afinitas Zona Ketersediaan (paling banyak, 50% panggilan aktif hilang)

Gambar sebelumnya menunjukkan arsitektur yang disederhanakan yang menggunakan afinitas Availability Zone. Keuntungan komparatif dari pendekatan ini menjadi jelas jika seseorang memperhitungkan efek dari pemadaman Zona Ketersediaan yang lengkap. Seperti yang digambarkan dalam diagram, jika Availability Zone 2 hilang, 50% panggilan aktif paling banyak terpengaruh (dengan asumsi penyeimbangan beban yang sama antara Availability Zones). Seandainya Afinitas Zona Ketersediaan tidak diterapkan, beberapa panggilan akan mengalir antara Availability Zone di satu Wilayah dan kegagalan kemungkinan besar akan memengaruhi lebih dari 50% panggilan aktif.

Untuk meminimalkan latensi lalu lintas, AWS juga menyarankan Anda mempertimbangkan untuk menggunakan grup EC2 penempatan dalam setiap Availability Zone. Instans yang diluncurkan dalam grup EC2 penempatan yang sama memiliki bandwidth yang lebih tinggi dan latensi yang berkurang karena EC2 memastikan kedekatan jaringan dari instans ini relatif satu sama lain.

Gunakan jenis EC2 instance jaringan yang disempurnakan

Memilih jenis instans yang tepat di Amazon EC2 memastikan keandalan sistem serta penggunaan infrastruktur yang efisien. EC2 menyediakan berbagai pilihan jenis instance yang dioptimalkan agar sesuai dengan kasus penggunaan yang berbeda. Tipe instans terdiri dari berbagai kombinasi CPU, memori, penyimpanan, dan kapasitas jaringan, serta memberi Anda fleksibilitas untuk memilih campuran sumber daya yang sesuai untuk aplikasi Anda. Jenis instance jaringan yang ditingkatkan ini memastikan bahwa beban kerja SIP yang berjalan pada mereka memiliki akses ke bandwidth yang konsisten dan latensi agregat yang relatif lebih rendah. Tambahan baru-baru ini ke Amazon EC2 adalah ketersediaan Adaptor Jaringan Elastis (ENA) yang menyediakan bandwidth hingga 100

Gbps. Katalog terbaru dari jenis EC2 instans dan fitur terkait dapat ditemukan di halaman jenis EC2 instance.

Bagi sebagian besar pelanggan, instans Compute Optimized generasi terbaru harus memberikan nilai terbaik untuk biayanya. Misalnya, C5N mendukung Adaptor Jaringan Elastis baru dengan bandwidth hingga 100 Gbps dengan jutaan paket per detik (PPS). Sebagian besar aplikasi real-time juga akan mendapat manfaat dari penggunaan Intel Data Plane Developer Kit (DPDK) yang dapat sangat meningkatkan pemrosesan paket jaringan.

Namun, selalu merupakan praktik terbaik untuk membandingkan berbagai jenis EC2 instans sesuai dengan kebutuhan Anda untuk melihat jenis instance mana yang paling cocok untuk Anda. Benchmarking juga memungkinkan Anda menemukan parameter konfigurasi lainnya, seperti jumlah maksimum panggilan yang dapat diproses oleh jenis instans tertentu pada suatu waktu.

Pertimbangan keamanan

Komponen aplikasi RTC biasanya berjalan langsung di internet yang menghadap EC2 instans Amazon. Selain TCP, flow menggunakan protokol seperti UDP dan SIP. Dalam kasus ini, AWS Shield Standard melindungi EC2 instance Amazon dari serangan lapisan infrastruktur umum (Layer 3 dan 4) DDo S, seperti serangan refleksi UDP, refleksi DNS, refleksi NTP, refleksi SSDP, dan sebagainya. AWS Shield Standard menggunakan berbagai teknik seperti pembentukan lalu lintas berbasis prioritas yang secara otomatis terlibat ketika tanda tangan serangan DDo S yang terdefinisi dengan baik terdeteksi.

AWS juga memberikan perlindungan lanjutan terhadap serangan DDo S yang besar dan canggih untuk aplikasi ini dengan mengaktifkan AWS Shield Advanced alamat IP Elastic. AWS Shield Advanced menyediakan deteksi DDo S yang ditingkatkan yang secara otomatis mendeteksi jenis AWS sumber daya dan ukuran EC2 instans dan menerapkan mitigasi standar yang sesuai dengan perlindungan terhadap banjir SYN atau UDP. Dengan AWS Shield Advanced, pelanggan juga dapat membuat profil mitigasi kustom mereka sendiri dengan melibatkan 24x7 AWS DDo S Response Team (DRT). AWS Shield Advanced juga memastikan bahwa selama serangan DDo S, semua Daftar Kontrol Akses Jaringan VPC Amazon Anda (ACLs) secara otomatis diberlakukan di perbatasan AWS jaringan yang memberi Anda akses ke bandwidth tambahan dan kapasitas scrubbing untuk mengurangi serangan S volumetrik besar. DDo

Kesimpulan

Beban kerja komunikasi real-time (RTC) dapat digunakan AWS untuk mencapai skalabilitas, elastisitas, dan ketersediaan tinggi sambil memenuhi persyaratan utama. Saat ini, beberapa pelanggan menggunakan AWS, mitranya, dan solusi open source untuk menjalankan beban kerja RTC dengan biaya yang lebih rendah dan kelincahan yang lebih cepat serta pengurangan jejak global.

Arsitektur referensi dan praktik terbaik yang disediakan dalam white paper ini dapat membantu pelanggan berhasil mengatur beban kerja RTC AWS dan mengoptimalkan solusi untuk memenuhi kebutuhan pengguna akhir sambil mengoptimalkan cloud.

Akronim

Akronim yang digunakan dalam dokumen ini meliputi:

ACL - Daftar Kontrol Akses

ALB - Application Load Balancer

APNs — Layanan Pemberitahuan Push Apple

BGP — Protokol Gerbang Perbatasan

CDR - Catatan Detail Panggilan

COTS — perangkat lunak komersial off-the-shelf

DDoS — didistribusikan denial-of-service

DNS — Sistem Nama Domain

DPDK — Kit Pengembang Pesawat Data Intel

Tim Respons DRT - DDo S

ENA - Adaptor Jaringan Elastis

EPC - Inti Paket Berevolusi

FCM — Pesan Cloud Firebase

HA - Ketersediaan Tinggi

IRC — Obrolan Relay Internet

ISDN — Jaringan Digital Layanan Terpadu

NAT — terjemahan alamat jaringan

OPUS - dukungan pengguna pemosisian online

PBX — Pertukaran Cabang Swasta

PRI - Antarmuka Tingkat Primer

PSTN - Jaringan Telepon Beralih Publik

RAID - Array Redundan dari Disk Independen

RTC — komunikasi waktu nyata

RTP —Protokol Transportasi Waktu Nyata

SAN - Jaringan Area Penyimpanan

SBC - pengontrol batas sesi

SIP —Protokol Inisiasi Sesi

SPOF — titik kegagalan tunggal

SRV - Layanan

SS7 — Sistem Pensinyalan n.7

STUN - Utilitas Traversal Sesi untuk NAT

SYN —Sinkronisasi

TCP - Protokol Kontrol Transmisi

TDM - pembagian waktu multiplexing

TURN - Traversal Menggunakan Relay di sekitar NAT

UDP - Protokol Datagram Pengguna

URI - Pengidentifikasi Sumber Daya Seragam

VIP - IP virtual

VNF - Fungsi Jaringan Virtual

VoIP - Voice Over IP

VPC - Awan Pribadi Virtual

WebRTC — komunikasi real-time web

Kontributor

Individu dan organisasi berikut berkontribusi terhadap dokumen ini:

- Mounir Chennana, Arsitek Solusi Senior, Amazon Web Services
- Mohammed Al-Mehdar, Arsitek Solusi Senior, Amazon Web Services
- Ejaz Sial, Arsitek Solusi Senior, Amazon Web Services
- Ahmad Khan, Arsitek Solusi Senior, Amazon Web Services
- Tipu Qureshi, Insinyur Utama,, Amazon Web AWS Dukungan Services
- Hasan Khan, Manajer Akun Teknis Senior, Amazon Web Services
- Shoma Chakravarty, Pemimpin Teknis WW, Telecom, Amazon Web Services

Revisi dokumen

Untuk mengetahui jika ada perubahan pada laporan resmi ini, Anda dapat berlangganan umpan RSS.

Perubahan	Deskripsi	Tanggal
Laporan resmi diperbarui	Diperbarui untuk layanan dan fitur terbaru.	5 Mei 2022
Laporan resmi diperbarui	Diperbarui untuk layanan dan fitur terbaru.	13 Februari 2020
Publikasi awal	Whitepaper pertama kali diterbitkan.	1 Oktober 2018

Pemberitahuan

Pelanggan bertanggung jawab untuk membuat penilaian independen mereka sendiri atas informasi dalam dokumen ini. Dokumen ini: (a) hanya untuk tujuan informasi, (b) mewakili penawaran dan praktik produk AWS saat ini, yang dapat berubah tanpa pemberitahuan, dan (c) tidak membuat komitmen atau jaminan apa pun dari AWS dan afiliasinya, pemasok, atau pemberi lisensinya. Produk atau layanan AWS disediakan “sebagaimana adanya” tanpa jaminan, pernyataan, atau ketentuan dalam bentuk apa pun, baik tersurat maupun tersirat. Tanggung jawab dan kewajiban AWS kepada pelanggannya dikendalikan oleh perjanjian AWS, dan dokumen ini bukan bagian dari, juga tidak mengubah, perjanjian apa pun antara AWS dan pelanggannya.

AWS Glosarium

Untuk AWS terminologi terbaru, lihat AWS glosarium di Referensi.Glosarium AWS

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.