aws

Panduan Developer

Pengontrol Pemulihan Aplikasi Amazon (ARC)

Pengontrol Pemulihan Aplikasi Amazon (ARC): Panduan Developer

Table of Contents

Apa itu ARC?

Amazon Application Recovery Controller (ARC) membantu Anda mempersiapkan dan menyelesaikan pemulihan yang lebih cepat untuk aplikasi yang berjalan di Infrastruktur Cloud AWS Global.

ARC menyediakan kemampuan berikut:

- Pemulihan Multi-Availability Zone (AZ), termasuk pergeseran zona dan pergeseran otomatis zona, yang memungkinkan Anda pulih dari gangguan AZ tunggal dengan mengalihkan lalu lintas sementara dari AZ yang terganggu ke AZ yang sehat.
- Pemulihan Multi-Wilayah, yang mencakup kontrol perutean dan sakelar Wilayah untuk pemulihan aplikasi Regional, dan pemeriksaan kesiapan untuk pemantauan aplikasi.

Pemulihan Zona Ketersediaan Multi-Ketersediaan

Pergeseran zona

Anda dapat menggunakan ARC zonal shift untuk mengisolasi dan memulihkan dengan cepat dari gangguan Availability Zone (AZ) tunggal. Pergeseran zona untuk sementara menggeser lalu lintas untuk sumber daya yang didukung dari AZ yang terganggu menjadi sehat AZs di Wilayah yang sama AWS . Memulai pergeseran zona membantu aplikasi Anda pulih dengan cepat, misalnya, dari penerapan kode pengembang yang buruk atau dari AWS gangguan pada satu AZ. Mengalihkan lalu lintas dari AZ yang terganggu mengurangi dampak bagi klien yang menggunakan aplikasi Anda di AZ yang mengalami gangguan.

Anda dapat memulai pergeseran zona untuk sumber daya apa pun yang didukung di akun Anda di AWS Wilayah. Pergeseran zona bersifat manual dan sementara. Saat Anda memulai pergeseran zona, Anda harus menentukan kedaluwarsa (dapat diperpanjang) hingga tiga hari. Untuk mengaktifkan pergeseran zona untuk sumber daya yang didukung, lihat. Sumber daya yang didukung

Pergeseran otomatis zona

Autoshift zona ARC mengizinkan AWS untuk mengalihkan lalu lintas dari AZ yang terganggu untuk sumber daya yang didukung, atas nama Anda, menjadi sehat AZs di Wilayah yang sama. AWS AWS memulai pergeseran otomatis zona ketika telemetri internal menunjukkan bahwa ada gangguan pada satu AZ di AWS Wilayah yang berpotensi berdampak pada pelanggan. Telemetri internal menggabungkan metrik dari berbagai sumber, termasuk AWS jaringan, dan layanan Amazon dan Elastic EC2 Load Balancing.

Autoshift zona bersifat sementara. AWS mengakhiri pergeseran otomatis zona ketika indikator telemetri internal menunjukkan bahwa tidak ada lagi masalah atau masalah potensial.

Untuk mempelajari lebih lanjut tentang kemampuan ini, lihat bab-babnya berikut:

- Pergeseran zona di ARC
- Autoshift zona di ARC

Pemulihan Multi-Wilayah

Sakelar wilayah

Sakelar wilayah di ARC menyediakan solusi terpusat, otomatis, dan dapat diamati untuk pemulihan aplikasi Multi-wilayah. Peralihan wilayah membantu Anda merencanakan dan mengoordinasikan pemulihan untuk aplikasi Anda di seluruh Wilayah AWS, untuk membantu memastikan kelangsungan bisnis dan mengurangi biaya operasional.

Anda dapat menggunakan Region switch untuk mengatur tugas pemulihan skala besar dan kompleks untuk sumber daya aplikasi Anda, di beberapa akun. AWS Jika Wilayah AWS menjadi terganggu, paket yang Anda buat dengan menggunakan sakelar Wilayah dapat gagal atau mengalihkan sumber daya Anda ke Wilayah lain, sehingga aplikasi Anda dapat terus beroperasi, dalam keadaan sehat Wilayah AWS.

Kontrol perutean

Kontrol perutean ARC yang sangat andal memungkinkan pemulihan Multi-wilayah sehingga aplikasi Anda dapat melakukan failover lalu lintas DNS Sistem Nama Domain di seluruh Wilayah. AWS

Jika aplikasi Anda dirancang untuk beroperasi dari beberapa AWS Wilayah, Anda dapat menggunakan kontrol perutean ARC untuk failover antar Wilayah. Kontrol perutean memungkinkan Anda untuk melakukan failover lalu lintas dari AWS Wilayah yang terganggu ke AWS Wilayah yang sehat, sehingga Anda dapat memastikan bahwa aplikasi Anda mempertahankan ketersediaan. Kontrol perutean mencakup aturan keselamatan, yang membantu melindungi Anda dari hasil yang tidak diinginkan dengan memaksakan pagar pembatas yang Anda tentukan. Misalnya, Anda dapat memberlakukan aturan keamanan bahwa hanya satu replika aplikasi Anda, aktif atau siaga, yang diaktifkan dan digunakan.

Pemeriksaan kesiapan

Pemeriksaan kesiapan ARC terus memantau kuota AWS sumber daya, kapasitas, dan kebijakan perutean jaringan, dan dapat memberi tahu Anda tentang perubahan yang dapat memengaruhi kemampuan Anda untuk melakukan failover ke aplikasi replika dan pulih dari gangguan Wilayah. Pemeriksaan kesiapan terus-menerus memastikan bahwa Anda dapat mempertahankan aplikasi Multi-wilayah Anda dalam keadaan yang diskalakan dan dikonfigurasi untuk menangani lalu lintas failover. Pemeriksaan kesiapan berguna ketika Anda pertama kali mengkonfigurasi ARC, dan selama operasi aplikasi normal. Pemeriksaan kesiapan tidak dimaksudkan untuk digunakan di jalur kritis untuk failover selama acara.

Untuk mempelajari lebih lanjut tentang kemampuan ini, lihat bab-babnya berikut:

- Sakelar wilayah di ARC
- Kontrol perutean di ARC
- Pemeriksaan kesiapan di ARC

Bandingkan kemampuan pemulihan Multi-AZ dan Multi-wilayah di ARC

Pergeseran zona, pergeseran otomatis zona, kontrol perutean, dan sakelar Wilayah di Amazon Application Recovery Controller (ARC) semuanya dapat mencapai pemulihan yang cepat dan membantu Anda memastikan ketahanan untuk aplikasi Anda. AWS Fitur-fitur ini sangat tersedia, dan membantu mendukung pemulihan dalam skenario saat aplikasi Anda mengalami peningkatan latensi atau ketersediaan berkurang. Fitur-fitur ini juga membantu memulihkan aplikasi dengan cepat dengan mengalihkan lalu lintas dari gangguan yang terisolasi, yang membatasi dampak dan waktu yang hilang dari gangguan.

Kontrol perutean dan sakelar Wilayah difokuskan pada AWS aplikasi yang ada di beberapa Wilayah AWS (Multi-wilayah), sedangkan pergeseran zona dan pergeseran otomatis zona hanya mendukung perpindahan lalu lintas untuk sumber daya yang didukung dengan aplikasi Multi-AZ.

Informasi dalam tabel berikut mencakup beberapa fitur utama dari kemampuan ketahanan ARC. Deskripsi ini dapat membantu Anda lebih memahami bagaimana opsi tertentu mungkin menjadi pilihan terbaik untuk kebutuhan aplikasi Anda.

Kontrol perutean	Sakelar wilayah	Pergeseran zona	Pergeseran otomatis zona
Regional Mengalihkan lalu lintas dari satu AWS Wilayah ke Wilayah lain (terutama)	Regional Mengalihkan lalu lintas dari satu AWS Wilayah ke Wilayah lain (terutama)	Zonal Mengalihkan lalu lintas dari Availability Zone Lalu lintas menuju ke Availability Zone lain di Wilayah, bukan ke target tertentu	Zonal Mengalihkan lalu lintas dari Availability Zone Lalu lintas menuju ke Availability Zone lain di Wilayah, bukan ke target tertentu
Membutuhkan pengaturan	Membutuhkan pengaturan	Mungkin memerlukan pengaturan Memerlukan opt-in untuk beberapa	Membutuhkan pengaturan

Kontrol perutean	Sakelar wilayah	Pergeseran zona	Pergeseran otomatis zona
Membutuhkan konfigurasi dan pengaturan	Membutuhkan konfigurasi dan pengaturan	sumber daya yang didukung Untuk informasi lebih lanjut, lihat Sumber daya yang didukung	Harus diaktifkan untuk sumber daya yang didukung Untuk informasi lebih lanjut, lihat Sumber daya yang didukung
Diprakarsai pelanggan Pelanggan menentukan kapan harus merutekan ulang lalu lintas	Diprakarsai pelanggan Pelanggan menentukan kapan harus merutekan ulang lalu lintas	Diprakarsai pelanggan Pelanggan menentukan kapan harus memulai pergeseran zona	AWS-diprakarsai AWS mengalihkan lalu lintas aplikasi dari AZ atas nama Anda
Berbasis biaya Membutuhkan biaya terpisah untuk kontrol routing	Berbasis biaya Memerlukan biaya terpisah untuk paket peralihan Wilayah	Termasuk dengan layanan (tanpa biaya tambahan) Membuat pergesera n zona untuk memindahkan lalu lintas dari disertaka n untuk sumber AZs daya yang didukung	Termasuk dengan layanan (tanpa biaya tambahan) Memulai pergesera n otomatis untuk memindahkan lalu lintas dari AZs atas nama Anda disertaka n untuk sumber daya yang didukung
Tidak kedaluwarsa Lalu lintas dapat dialihkan ke replika tanpa batas waktu	Tidak kedaluwarsa Aplikasi dapat digeser ke replika tanpa batas	Sementara Semua pergesera n zona harus diatur untuk kedaluwarsa	Sementara AWS memulai dan mengakhiri autoshifts

Untuk mempelajari lebih lanjut tentang masing-masing fitur ini, lihat bab-babnya:

- Pergeseran zona di ARC
- Autoshift zona di ARC
- Kontrol perutean di ARC
- Sakelar wilayah di ARC

Gunakan zonal shift dan zonal autoshift untuk memulihkan aplikasi di ARC

Bagian ini menjelaskan cara menggunakan kemampuan di Amazon Application Recovery Controller (ARC) untuk memulihkan AWS sumber daya Anda dari masalah di Availability Zone (AZ) yang terganggu. Pergeseran zona dan pergeseran otomatis zona untuk sementara mengalihkan lalu lintas untuk sumber daya yang didukung dari AZ yang rusak, yang mengurangi waktu pemulihan untuk aplikasi Anda.

Perbedaan utama antara zonal shift dan zonal autoshift adalah bahwa salah satunya adalah pergeseran lalu lintas manual yang Anda kontrol, dan yang lainnya menggeser lalu lintas dari gangguan secara otomatis atas nama Anda.

- Dengan pergeseran zona, Anda secara manual mengalihkan lalu lintas untuk sumber daya yang didukung di Wilayah AWS jauh dari Availability Zone.
- Dengan pergeseran otomatis zona, lalu lintas untuk sumber daya yang didukung secara otomatis bergeser dari AZ yang terganggu dan dialihkan ke sehat AZs di Wilayah yang sama. AWS

Topik berikut menjelaskan kemampuan pergeseran zona dan pergeseran otomatis zona, dan cara menggunakannya.

Topik

Pergeseran zona di ARC

Pergeseran zona Amazon Application Recovery Controller (ARC) memungkinkan Anda mengalihkan lalu lintas untuk sumber daya yang didukung dari Availability Zone (AZ) yang terganggu Wilayah AWS ke sehat AZs di Wilayah yang sama. Mengalihkan lalu lintas sumber daya Anda dari AZ yang terganggu mengurangi durasi dan tingkat keparahan dampak yang disebabkan oleh pemadaman listrik, atau masalah perangkat keras atau perangkat lunak di AZ, dan membantu mengurangi masalah dan memulihkan aplikasi Anda dengan cepat. Anda mungkin memilih untuk mengalihkan lalu lintas, misalnya, karena penerapan yang buruk menyebabkan masalah latensi, atau karena Availability Zone terganggu.

Anda harus memilih sumber daya untuk menggunakan pergeseran zona. Untuk informasi lebih lanjut, lihatSumber daya yang didukung.

Sebelum Anda memulai pergeseran zona, Anda harus menskalakan aplikasi Anda dan memastikan bahwa Anda memiliki kapasitas yang cukup untuk mengalihkan lalu lintas dari Availability Zone. Setelah prescaling, Anda dapat memilih Availability Zone untuk beralih dari dan sumber daya untuk mengalihkan lalu lintas, dan kemudian memulai pergeseran zona. Anda dapat membatalkan shift kapan saja agar lalu lintas mulai kembali ke Availability Zone asli. Untuk informasi selengkapnya, lihat Praktik terbaik untuk pergeseran zona di ARC

Semua pergeseran zona adalah mitigasi sementara. Anda menetapkan kedaluwarsa awal ketika Anda memulai shift zona, dari satu menit hingga tiga hari (72 jam), yang dapat Anda perpanjang, jika Anda perlu melanjutkan shift lalu lintas.

Dalam skenario tertentu, pergeseran zona tidak menggeser lalu lintas dari AZ. Untuk informasi selengkapnya, lihat Sumber daya yang didukung.

Cara kerja pergeseran zona

Saat Anda memulai pergeseran zona untuk sumber daya yang didukung, lalu lintas untuk sumber daya dipindahkan dari Availability Zone (AZ) yang telah Anda tentukan. Sumber daya yang didukung ARC menyediakan integrasi yang menandai AZ yang ditentukan sebagai tidak sehat, yang mengakibatkan lalu lintas bergeser dari AZ yang rusak.

Lalu lintas mulai bergeser - Saat Anda memulai pergeseran zona di ARC, Anda mungkin tidak segera melihat lalu lintas keluar dari Availability Zone. Diperlukan waktu singkat untuk menyelesaikan koneksi yang ada dan sedang berlangsung di Availability Zone, tergantung pada perilaku klien dan penggunaan kembali koneksi. Pengaturan DNS dan faktor lain termasuk koneksi yang ada dapat selesai hanya dalam beberapa menit, tetapi mungkin memakan waktu lebih lama. Untuk informasi selengkapnya, lihat Memastikan pergeseran lalu lintas selesai dengan cepat.

Pergeseran lalu lintas berakhir - Ketika pergeseran zona berakhir atau Anda membatalkannya, ARC mengambil langkah-langkah untuk menghentikan pergeseran lalu lintas dan membalikkan proses untuk memulai pergeseran lalu lintas. Sekarang, AZ yang dipulihkan diakui tersedia untuk sumber daya dan resume lalu lintas yang mengalir ke AZ.

Anda harus mengatur semua pergeseran zona untuk kedaluwarsa saat Anda memulai shift. Anda awalnya dapat mengatur pergeseran zona untuk kedaluwarsa dalam maksimal tiga hari (72 jam). Namun, Anda dapat memperbarui pergeseran zona untuk mengatur kedaluwarsa baru

kapan saja. Anda juga dapat membatalkan pergeseran zona sebelum kedaluwarsa, jika Anda siap mengembalikan lalu lintas ke Availability Zone.

Ketika lalu lintas tidak bergeser - Dalam skenario tertentu, pergeseran zona tidak menggeser lalu lintas dari Availability Zone. Misalnya, Anda memulai pergeseran zona untuk penyeimbang beban ketika kelompok target penyeimbang beban di AZs tidak memiliki instance apa pun, atau jika semua instance tidak sehat. Dalam skenario ini, penyeimbang beban berada dalam keadaan terbuka gagal dan memulai pergeseran zona tidak mengalihkan lalu lintas.

Sebelum Anda memulai pergeseran zona untuk sumber daya, pastikan bahwa semua kondisi untuk pergeseran zona yang berhasil terpenuhi. AWS sumber daya menangani pergeseran zona secara berbeda. Untuk informasi selengkapnya tentang dukungan pergeseran zona, lihatSumber daya yang didukung.

Wilayah AWS ketersediaan untuk pergeseran zona

Untuk informasi terperinci tentang dukungan Regional dan titik akhir layanan untuk Amazon Application Recovery Controller (ARC), lihat titik akhir dan kuota Amazon Application Recovery Controller (ARC) di Referensi Umum Amazon Web Services.

Zonal shift dan zonal autoshift saat ini tersedia di daftar di sini. Wilayah AWS Pergeseran zona dan pergeseran otomatis zona juga tersedia di Wilayah Tiongkok, yaitu Wilayah Tiongkok (Beijing) dan Wilayah Tiongkok (Ningxia). Sumber daya yang menggunakan Amazon Application Recovery Controller (ARC) mungkin memiliki pertimbangan tambahan. Untuk informasi lebih lanjut, lihatSumber daya yang didukung.

Nama Wilayah	Wilayah	Titik Akhir	Protokol	
AS Timur (Ohio)	us–east-2	arc-zonal-shift.us-east-2.amazonaws.com	HTTPS	
		arc-zonal-shift-fips.us-east-2.api.aws	HTTPS	
		arc-zonal-shift.us-east-2.api.aws	HTTPS	
AS Timur (Virginia Utara)	us-east-1	arc-zonal-shift.us-east-1.amazonaws.com	HTTPS	
		arc-zonal-shift-fips.us-east-1.api.aws	HTTPS	
		arc-zonal-shift.us-east-1.api.aws	HTTPS	

Nama Wilayah	Wilayah	Titik Akhir	Protokol
AS Barat (California Utara)	us-west-1	arc-zonal-shift.us-west-1.amazonaws.com	HTTPS
		arc-zonal-shift-fips.us-west-1.api.aws	HTTPS
		arc-zonal-shift.us-west-1.api.aws	HTTPS
AS Barat (Oregon)	us-west-2	arc-zonal-shift.us-west-2.amazonaws.com	HTTPS
		arc-zonal-shift-fips.us-west-2.api.aws	HTTPS
		arc-zonal-shift.us-west-2.api.aws	HTTPS
Afrika (Cape Town)	af-south-1	arc-zonal-shift.af-south-1.amazonaws.com	HTTPS
		arc-zonal-shift.af-south-1.api.aws	HTTPS
Asia Pasifik (Hong Kong)	ap-east-1	arc-zonal-shift.ap-east-1.amazonaws.com	HTTPS
		arc-zonal-shift.ap-east-1.api.aws	HTTPS
Asia Pasifik (Hyderabad)	ap-south-2	arc-zonal-shift.ap-south-2.amazonaws.com	HTTPS
		arc-zonal-shift.ap-south-2.api.aws	HTTPS
Asia Pasifik (Jakarta)	ap-southeast-3	arc-zonal-shift.ap-southeast-3.amazonaws.com	HTTPS
		arc-zonal-shift.ap-southeast-3.api.aws	HTTPS
Asia Pasifik (Malaysia)	ap-southeast-5	arc-zonal-shift.ap-southeast-5.amazonaws.com	HTTPS
		arc-zonal-shift.ap-southeast-5.api.aws	HTTPS

Nama Wilayah	Wilayah	Titik Akhir	Protokol	
Asia Pasifik (Melbourne)	ap-southeast-4	arc-zonal-shift.ap-southeast-4.amazonaws.com arc-zonal-shift.ap-southeast-4.api.aws	HTTPS HTTPS	
Asia Pasifik (Mumbai)	ap-south-1	arc-zonal-shift.ap-south-1.amazonaws.com arc-zonal-shift.ap-south-1.api.aws	HTTPS HTTPS	
Asia Pasifik (Osaka)	ap-northeast-3	arc-zonal-shift.ap-northeast-3.amazonaws.com arc-zonal-shift.ap-northeast-3.api.aws	HTTPS HTTPS	
Asia Pasifik (Seoul)	ap-northeast-2	arc-zonal-shift.ap-northeast-2.amazonaws.com arc-zonal-shift.ap-northeast-2.api.aws	HTTPS HTTPS	
Asia Pasifik (Singapura)	ap-southeast-1	arc-zonal-shift.ap-southeast-1.amazonaws.com arc-zonal-shift.ap-southeast-1.api.aws	HTTPS HTTPS	
Asia Pasifik (Sydney)	ap-southeast-2	arc-zonal-shift.ap-southeast-2.amazonaws.com arc-zonal-shift.ap-southeast-2.api.aws	HTTPS HTTPS	
Asia Pasifik (Taipei)	ap-timur-2	arc-zonal-shift.ap-east-2.amazonaws.com arc-zonal-shift.ap-east-2.api.aws	HTTPS HTTPS	
Asia Pasifik (Thailand)	ap-tenggara 7	arc-zonal-shift.ap-southeast-7.amazonaws.com arc-zonal-shift.ap-southeast-7.api.aws	HTTPS HTTPS	

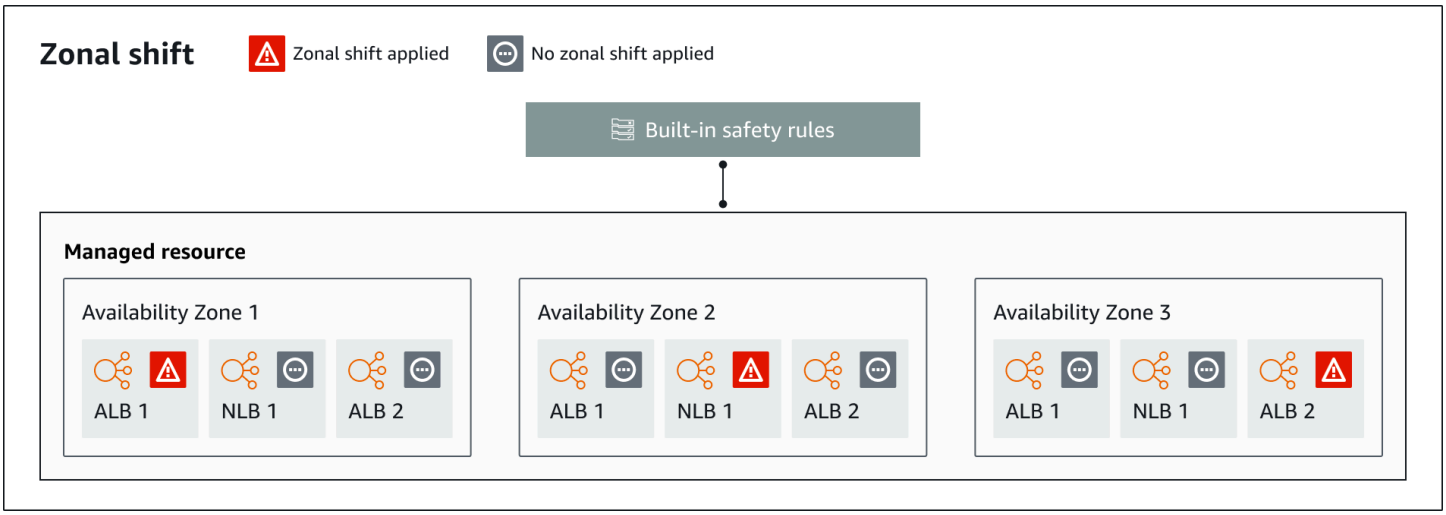
Nama Wilayah	Wilayah	Titik Akhir	Protokol
Asia Pacific (Tokyo)	ap-northeast-1	arc-zonal-shift.ap-northeast-1.amazonaws.com	HTTPS
		arc-zonal-shift.ap-northeast-1.api.aws	HTTPS
Kanada (Pusat)	ca-central-1	arc-zonal-shift.ca-central-1.amazonaws.com	HTTPS
		arc-zonal-shift-fips.ca-central-1.api.aws	HTTPS
		arc-zonal-shift.ca-central-1.api.aws	HTTPS
Kanada Barat (Calgary)	ca-west-1	arc-zonal-shift.ca-west-1.amazonaws.com	HTTPS
		arc-zonal-shift-fips.ca-west-1.api.aws	HTTPS
		arc-zonal-shift.ca-west-1.api.aws	HTTPS
Eropa (Frankfurt)	eu-central-1	arc-zonal-shift.eu-central-1.amazonaws.com	HTTPS
		arc-zonal-shift.eu-central-1.api.aws	HTTPS
Eropa (Irlandia)	eu-west-1	arc-zonal-shift.eu-west-1.amazonaws.com	HTTPS
		arc-zonal-shift.eu-west-1.api.aws	HTTPS
Eropa (London)	eu-west-2	arc-zonal-shift.eu-west-2.amazonaws.com	HTTPS
		arc-zonal-shift.eu-west-2.api.aws	HTTPS
Eropa (Milan)	eu-south-1	arc-zonal-shift.eu-south-1.amazonaws.com	HTTPS
		arc-zonal-shift.eu-south-1.api.aws	HTTPS
Eropa (Paris)	eu-west-3	arc-zonal-shift.eu-west-3.amazonaws.com	HTTPS
		arc-zonal-shift.eu-west-3.api.aws	HTTPS
Eropa (Spanyol)	eu-south-2	arc-zonal-shift.eu-south-2.amazonaws.com	HTTPS
		arc-zonal-shift.eu-south-2.api.aws	HTTPS

Nama Wilayah	Wilayah	Titik Akhir	Protokol
Eropa (Stockholm)	eu-north-1	arc-zonal-shift.eu-north-1.amazonaws.com	HTTPS
		arc-zonal-shift.eu-north-1.api.aws	HTTPS
Eropa (Zürich)	eu-central-2	arc-zonal-shift.eu-central-2.amazonaws.com	HTTPS
		arc-zonal-shift.eu-central-2.api.aws	HTTPS
Israel (Tel Aviv)	il-central-1	arc-zonal-shift.il-central-1.amazonaws.com	HTTPS
		arc-zonal-shift.il-central-1.api.aws	HTTPS
Meksiko (Tengah)	mx-pusat-1	arc-zonal-shift.mx-central-1.amazonaws.com	HTTPS
		arc-zonal-shift.mx-central-1.api.aws	HTTPS
Timur Tengah (Bahrain)	me-south-1	arc-zonal-shift.me-south-1.amazonaws.com	HTTPS
		arc-zonal-shift.me-south-1.api.aws	HTTPS
Timur Tengah (UAE)	me-central-1	arc-zonal-shift.me-central-1.amazonaws.com	HTTPS
		arc-zonal-shift.me-central-1.api.aws	HTTPS
Amerika Selatan (Sao Paulo)	sa-east-1	arc-zonal-shift.sa-east-1.amazonaws.com	HTTPS
		arc-zonal-shift.sa-east-1.api.aws	HTTPS
AWS GovCloud (AS-Timur)	us-gov-east-1	arc-zonal-shift.us-gov-east-1.amazonaws.com	HTTPS
		arc-zonal-shift-fips.us-gov-east-1.api.aws	HTTPS
		arc-zonal-shift.us-gov-east-1.api.aws	HTTPS

Nama Wilayah	Wilayah	Titik Akhir	Protokol	
AWS GovCloud (AS-Barat)	us-gov-west-1	arc-zonal-shift.us-gov-west-1.amazonaws.com	HTTPS	
		arc-zonal-shift-fips.us-gov-west-1.api.aws	HTTPS	
		arc-zonal-shift.us-gov-west-1.api.aws	HTTPS	

Komponen pergeseran zona

Diagram berikut mengilustrasikan contoh pergeseran zona yang menggeser lalu lintas dari Availability Zone di. Wilayah AWS Pemeriksaan yang dibangun ke dalam pergeseran zona mencegah Anda memulai pergeseran zona lain untuk sumber daya ketika sudah memiliki pergeseran aktif.



Berikut ini adalah komponen kemampuan pergeseran zona di ARC.

Pergeseran zona

Anda memulai pergeseran zona untuk sumber daya terkelola di AWS akun Anda untuk sementara memindahkan lalu lintas dari Availability Zone di Wilayah AWS, ke sehat AZs di Wilayah, untuk memulihkan dengan cepat dari masalah di satu AZ. Untuk informasi lebih lanjut tentang sumber daya yang didukung untuk pergeseran zona, lihat. Sumber daya yang didukung

Pemeriksaan keamanan bawaan

Pemeriksaan yang dibangun ke dalam ARC mencegah lebih dari satu pergeseran lalu lintas untuk sumber daya berlaku pada suatu waktu. Artinya, hanya satu pergeseran zona yang diprakarsai

pelanggan, praktik lari, atau pergeseran otomatis untuk sumber daya yang dapat secara aktif mengalihkan lalu lintas dari Availability Zone. Misalnya, jika Anda memulai pergeseran zona untuk sumber daya ketika saat ini digeser dengan pergeseran otomatis, pergeseran zona Anda diutamakan. Untuk informasi lebih lanjut, lihat Autoshift zona di ARC dan Hasil untuk latihan berjalan.

Pengidentifikasi sumber daya

Pengidentifikasi sumber daya untuk disertakan dalam pergeseran zona. Pengenal adalah Amazon Resource Name (ARN) untuk sumber daya.

Untuk pergeseran zona, Anda hanya dapat memilih sumber daya di akun Anda untuk AWS layanan yang didukung oleh ARC. Untuk informasi lebih lanjut tentang sumber daya yang didukung untuk pergeseran zona, lihat. Sumber daya yang didukung

Sumber daya terkelola

Beberapa AWS sumber daya harus secara manual ikut serta ke pergeseran zona, dan lainnya diaktifkan secara otomatis. Untuk informasi lebih lanjut tentang sumber daya yang didukung untuk pergeseran zona, lihat. Sumber daya yang didukung

Nama sumber daya

Nama sumber daya di ARC yang dapat Anda tentukan untuk pergeseran zona.

Status (status pergeseran zona)

Status untuk pergeseran zona. `Status`Untuk pergeseran zona dapat memiliki salah satu nilai berikut:

- AKTIF: Pergeseran zona dimulai dan aktif.
- KEDALUWARSA: Pergeseran zona telah kedaluwarsa (waktu kedaluwarsa terlampaui).
- DIBATALKAN: Pergeseran zona dibatalkan.

Status terapan

Status yang diterapkan menunjukkan apakah pergeseran berlaku untuk sumber daya. Pergeseran yang memiliki status `APPLIED` menentukan Availability Zone tempat lalu lintas aplikasi telah digeser untuk sumber daya, dan kapan shift itu berakhir.

Jenis shift

Mendefinisikan jenis pergeseran zona. `shiftType`Dapat memiliki nilai-nilai berikut:

- ZONAL_SHIFT

- ZONAL_AUTOSHIFT
- PRACTICE_RUN
- EKSPERIMEN FIS_

Waktu kedaluwarsa (waktu kedaluwarsa)

Waktu kedaluwarsa (waktu kedaluwarsa) untuk pergeseran zona. Pergeseran zona bersifat sementara. Untuk pergeseran zona, Anda awalnya dapat mengatur pergeseran zona agar aktif hingga tiga hari (72 jam).

Saat Anda memulai pergeseran zona, Anda menentukan berapa lama Anda ingin itu aktif, ARC mana yang mengonversi menjadi waktu kedaluwarsa (waktu kedaluwarsa). Anda dapat membatalkan pergeseran zona, misalnya, jika Anda siap mengembalikan lalu lintas ke Availability Zone. Atau Anda dapat memperpanjang pergeseran zona yang diprakarsai pelanggan dengan memperbaruinya untuk menentukan jangka waktu lain untuk kedaluwarsa.

Anda dapat membatalkan latihan pergeseran zona yang merupakan bagian dari pergeseran otomatis zona.

Bidang data dan kontrol untuk pergeseran zona

Saat Anda merencanakan kegagalan dan pemulihan bencana, pertimbangkan seberapa tangguh mekanisme failover Anda. Kami menyarankan Anda memastikan bahwa mekanisme yang Anda andalkan selama failover sangat tersedia, sehingga Anda dapat menggunakannya saat Anda membutuhkannya dalam skenario bencana. Biasanya, Anda harus menggunakan fungsi bidang data untuk mekanisme Anda kapan pun Anda bisa, untuk keandalan dan toleransi kesalahan terbesar. Dengan mengingat hal itu, penting untuk memahami bagaimana fungsionalitas layanan dibagi antara bidang kontrol dan pesawat data, dan kapan Anda dapat mengandalkan harapan keandalan ekstrim dengan bidang data layanan.

Seperti kebanyakan AWS layanan, fungsionalitas untuk kemampuan pergeseran zona didukung oleh pesawat kontrol dan pesawat data. Meskipun keduanya dibangun agar dapat diandalkan, bidang kontrol dioptimalkan untuk konsistensi data, sementara bidang data dioptimalkan untuk ketersediaan. Pesawat data dirancang untuk ketahanan sehingga dapat mempertahankan ketersediaan bahkan selama peristiwa yang mengganggu, ketika pesawat kontrol mungkin menjadi tidak tersedia.

Secara umum, bidang kontrol memungkinkan Anda melakukan fungsi manajemen dasar, seperti membuat, memperbarui, dan menghapus sumber daya dalam layanan. Pesawat data menyediakan fungsionalitas inti layanan.

Untuk informasi selengkapnya tentang bidang data, pesawat kontrol, dan cara AWS membangun layanan untuk memenuhi target ketersediaan tinggi, lihat paper Stabilitas statis menggunakan Availability Zones di Amazon Builders' Library.

Harga untuk pergeseran zona di ARC

Untuk pergeseran zona, Anda dapat memulai pergeseran zona untuk sumber daya yang didukung, untuk memulihkan aplikasi Anda dari masalah di Availability Zone. Tidak ada biaya tambahan untuk menggunakan zonal shift.

Untuk informasi harga terperinci untuk ARC dan contoh harga, lihat Harga ARC.

Praktik terbaik untuk pergeseran zona di ARC

Kami merekomendasikan praktik terbaik berikut untuk menggunakan pergeseran zona untuk pemulihan Multi-AZ di ARC.

Topik

- Perencanaan kapasitas dan pra-penskalaan
- Batasi waktu klien tetap terhubung ke titik akhir Anda
- Uji mulai pergeseran zona, terlebih dahulu
- Pastikan bahwa semua Availability Zone sehat dan mengambil lalu lintas
- Menggunakan operasi API pesawat data untuk pemulihan bencana
- Memindahkan lalu lintas dengan pergeseran zona hanya sementara

Perencanaan kapasitas dan pra-penskalaan

Pastikan bahwa Anda telah merencanakan, dan baik pra-skala atau dapat skala otomatis, kapasitas yang cukup untuk mengakomodasi beban ekstra yang dikenakan pada Availability Zone saat Anda memulai pergeseran zona. Dengan arsitektur berorientasi pemulihan, rekomendasi tipikal adalah untuk pra-skala kapasitas komputasi untuk memasukkan ruang kepala yang cukup untuk melayani lalu lintas puncak Anda ketika salah satu dari (biasanya) tiga replika Anda offline.

Saat Anda memulai pergeseran zona untuk sumber daya yang didukung dan lalu lintas digeser dari AZ, kapasitas yang digunakan aplikasi Anda untuk permintaan layanan akan dihapus. Anda harus memastikan bahwa Anda telah merencanakan pergeseran lalu lintas dari AZ dan dapat melanjutkan permintaan layanan di sisanya AZs.

Batasi waktu klien tetap terhubung ke titik akhir Anda

Ketika Amazon Application Recovery Controller (ARC) mengalihkan lalu lintas dari gangguan, misalnya, dengan menggunakan zonal shift atau zonal autoshift, mekanisme yang digunakan ARC untuk memindahkan lalu lintas aplikasi Anda adalah pembaruan DNS. Pembaruan DNS menyebabkan semua koneksi baru diarahkan menjauh dari lokasi yang rusak.

Namun, klien dengan koneksi terbuka yang sudah ada sebelumnya mungkin terus membuat permintaan terhadap lokasi yang rusak sampai klien terhubung kembali. Untuk memastikan pemulihan yang cepat, kami sarankan Anda membatasi jumlah waktu klien tetap terhubung ke titik akhir Anda.

Uji mulai pergeseran zona, terlebih dahulu

Uji secara teratur memindahkan lalu lintas dari Availability Zones untuk aplikasi Anda dengan memulai pergeseran zona. Rencanakan dan jalankan pergeseran zona awal, sebaiknya di lingkungan pengujian dan produksi, sebagai bagian dari pengujian failover reguler untuk memulihkan aplikasi Anda jika terjadi bencana. Pengujian rutin adalah bagian penting untuk memastikan bahwa Anda siap dan memiliki kepercayaan diri untuk mengurangi masalah ketika peristiwa operasional terjadi.

Pastikan bahwa semua Availability Zone sehat dan mengambil lalu lintas

Pergeseran zona bekerja dengan menandai sumber daya, yaitu replika aplikasi, sebagai tidak sehat di Availability Zone. Ini berarti bahwa sangat penting untuk memastikan bahwa sumber daya dalam aplikasi Anda umumnya sehat dan secara aktif mengambil lalu lintas di Availability Zone di suatu Wilayah. Kami menyarankan Anda memiliki dasbor untuk melacak ini, termasuk, misalnya, metrik Elastic Load Balancing untuk target yang tidak sehat dan BytesProcessed per Availability Zone.

Pertimbangkan untuk memantau kesehatan sumber daya Anda dari Wilayah kedua yang berdekatan. Keuntungan dari pendekatan ini adalah dapat lebih mewakili pengalaman pengguna akhir Anda, dan juga mengurangi risiko aplikasi dan pemantauan Anda terkena dampak bencana yang sama pada saat yang bersamaan.

Menggunakan operasi API pesawat data untuk pemulihan bencana

Untuk memulai pergeseran zona saat Anda perlu memulihkan aplikasi dengan cepat, dengan sedikit dependensi, sebaiknya gunakan AWS Command Line Interface atau API dengan tindakan pergeseran zona, dengan kredensil yang disimpan sebelumnya, jika memungkinkan. Anda juga dapat memulai pergeseran zona di AWS Management Console, untuk kemudahan penggunaan.

Tetapi ketika pemulihan yang cepat dan andal sangat penting, operasi pesawat data adalah pilihan yang lebih baik. Untuk informasi selengkapnya, lihat Panduan Referensi API Zonal Shift.

Memindahkan lalu lintas dengan pergeseran zona hanya sementara

Pergeseran zona memindahkan lalu lintas dari Availability Zone secara sementara, untuk mengurangi penurunan nilai. Anda harus mengembalikan sumber daya untuk aplikasi ke layanan segera setelah Anda mengambil tindakan untuk memperbaiki masalah. Ini memastikan bahwa keseluruhan aplikasi Anda dikembalikan ke keadaan semula yang sepenuhnya berlebihan dan tangguh.

Operasi API pergeseran zona

Tabel berikut mencantumkan operasi ARC API yang dapat Anda gunakan menggunakan zonal shift, yang memindahkan lalu lintas dari Availability Zone untuk aplikasi Multi-AZ. Tabel ini juga mencakup tautan ke dokumentasi yang relevan.

Untuk contoh cara menggunakan operasi API pergeseran zona umum dengan AWS Command Line Interface, lihatContoh menggunakan AWS CLI with zonal shift.

Tindakan	Menggunakan konsol ARC	Menggunakan ARC API
Mulai pergeseran zona	Lihat Memulai pergeseran zona	Lihat StartZonalShift
Perbarui pergeseran zona	Lihat Memperbarui atau membatalkan pergeseran zona	Lihat UpdateZonalShift
Daftar pergeseran zona	Lihat Pergeseran zona di ARC	Lihat ListZonalShifts
Daftar sumber daya terkelola	Lihat Sumber daya yang didukung	Lihat ListManagedResources
Dapatkan sumber daya terkelola	Lihat Sumber daya yang didukung	Lihat GetManagedResource

Tindakan	Menggunakan konsol ARC	Menggunakan ARC API
Batalkan pergeseran zona	Lihat Memperbarui atau membatalkan pergeseran zona	Lihat CancelZonalShift

Contoh menggunakan AWS CLI with zonal shift

Bagian ini memberikan contoh aplikasi menggunakan zonal shift, menggunakan AWS Command Line Interface untuk bekerja dengan kemampuan zonal shift di Amazon Application Recovery Controller (ARC) menggunakan operasi API. Contoh-contoh tersebut dimaksudkan untuk membantu Anda mengembangkan pemahaman dasar tentang cara bekerja dengan pergeseran zona menggunakan CLI.

Pergeseran zona di ARC memungkinkan Anda memindahkan lalu lintas sementara untuk sumber daya yang didukung dari Availability Zone sehingga aplikasi Anda dapat terus beroperasi secara normal dengan Zona Availabilty lainnya di file. Wilayah AWS

Semua pergeseran zona bersifat sementara dan harus ditetapkan pada awalnya untuk kedaluwarsa dalam waktu tiga hari. Namun, Anda dapat memperbarui pergeseran zona nanti untuk menetapkan kedaluwarsa baru.

Untuk informasi selengkapnya tentang penggunaan AWS CLI, lihat Referensi AWS CLI Perintah. Untuk daftar tindakan API pergeseran zona dan tautan ke informasi selengkapnya, lihatOperasi API pergeseran zona.

Mulai pergeseran zona

Anda dapat memulai pergeseran zona dengan CLI dengan menggunakan `start-zonal-shift` perintah.

```
aws arc-zonal-shift start-zonal-shift \
       --resource-identifier arn:aws:elasticloadbalancing:us-
east-1:111122223333:loadbalancer/app/Testing/5a19403ecd42dc05 \
       --away-from use1-az1 \
       --expires-in 10m \
       --comment "Shifting traffic away from use1-az1"
```

```
{
```

```
    "awayFrom": "use1-az1",
    "comment": "Shifting traffic away from use1-az1",
    "expiryTime": "2024-12-17T21:37:26-08:00",
    "resourceIdentifier": "arn:aws:elasticloadbalancing:us-
east-1:111122223333:loadbalancer/app/Testing/5a19403ecd42dc05",
    "startTime": "2024-12-17T21:27:26-08:00",
    "status": "ACTIVE",
    "zonalShiftId": "9ac9ec1e-1df1-0755-3dc5-8cf573cd9c38"
}
```

Dapatkan sumber daya terkelola

Anda bisa mendapatkan informasi tentang sumber daya yang dikelola dengan CLI dengan menggunakan perintah. `get-managed-resource`

```
aws arc-zonal-shift get-managed-resource \
       --resource-identifier arn:aws:elasticloadbalancing:us-
east-1:111122223333:loadbalancer/app/Testing/5a19403ecd42dc05
```

```
{
    "appliedWeights": {
        "use1-az1": 0.0,
        "use1-az2": 1.0,
        "use1-az6": 1.0
    },
    "arn": "arn:aws:elasticloadbalancing:us-east-1:111122223333:loadbalancer/app/
Testing/5a19403ecd42dc05",
    "autoshifts": [],
    "name": "Testing",
    "zonalAutoshiftStatus": "DISABLED",
    "zonalShifts": [
        {
            "appliedStatus": "APPLIED",
            "awayFrom": "use1-az1",
            "comment": "Shifting traffic away from use1-az1",
            "expiryTime": "2024-12-17T21:37:26-08:00",
            "resourceIdentifier": "arn:aws:elasticloadbalancing:us-
east-1:111122223333:loadbalancer/app/Testing/5a19403ecd42dc05",
            "startTime": "2024-12-17T21:27:26-08:00",
            "zonalShiftId": "9ac9ec1e-1df1-0755-3dc5-8cf573cd9c38"
            "shiftType": "MANUAL"
        }
    ]
```

```
}
```

Daftar sumber daya terkelola

Anda dapat membuat daftar sumber daya yang dikelola di akun Anda dengan CLI dengan menggunakan perintah. `list-managed-resources`

```
aws arc-zonal-shift list-managed-resources
```

```
{
    "items": [
        {
            "appliedWeights": {
                "use1-az1": 0.0,
                "use1-az2": 1.0,
                "use1-az6": 1.0
            },
            "arn": "arn:aws:elasticloadbalancing:us-east-1:111122223333:loadbalancer/
app/Testing/5a19403ecd42dc05",
            "autoshifts": [],
            "availabilityZones": [
                "use1-az1",
                "use1-az2",
                "use1-az6"
            ],
            "name": "Testing",
            "practiceRunStatus": "DISABLED",
            "zonalAutoshiftStatus": "DISABLED",
            "zonalShifts": [
                {
                    "appliedStatus": "APPLIED",
                    "awayFrom": "use1-az1",
                    "comment": "Shifting traffic away from use1-az1",
                    "expiryTime": "2024-12-17T21:37:26-08:00",
                    "resourceIdentifier": "arn:aws:elasticloadbalancing:us-
east-1:111122223333:loadbalancer/app/Testing/5a19403ecd42dc05",
                    "startTime": "2024-12-17T21:27:26-08:00",
                    "zonalShiftId": "9ac9ec1e-1df1-0755-3dc5-8cf573cd9c38"
                }
            ]
        }
    ]
```

```
}
```

Daftar pergeseran zona

Anda dapat membuat daftar pergeseran zona di akun Anda dengan CLI dengan menggunakan perintah. `list-zonal-shifts`

```
aws arc-zonal-shift list-zonal-shifts
```

```
{
    "items": [
        {
            "awayFrom": "use1-az1",
            "comment": "Shifting traffic away from use1-az1",
            "expiryTime": "2024-12-17T21:37:26-08:00",
            "resourceIdentifier": "arn:aws:elasticloadbalancing:us-
east-1:111122223333:loadbalancer/app/Testing/5a19403ecd42dc05",
            "startTime": "2024-12-17T21:27:26-08:00",
            "status": "ACTIVE",
            "zonalShiftId": "9ac9ec1e-1df1-0755-3dc5-8cf573cd9c38"
        }
    ]
}
```

Perbarui pergeseran zona

Anda dapat memperbarui pergeseran zona dengan CLI dengan menggunakan `update-zonal-shift` perintah.

```
aws arc-zonal-shift update-zonal-shift \
       --zonal-shift-id 9ac9ec1e-1df1-0755-3dc5-8cf573cd9c38 \
       --expires-in 1h \
       --comment "Still shifting traffic away from use1-az1"
```

```
{
    "awayFrom": "use1-az1",
    "comment": "Still shifting traffic away from use1-az1",
    "expiryTime": "2024-12-17T22:29:38-08:00",
    "resourceIdentifier": "arn:aws:elasticloadbalancing:us-
east-1:111122223333:loadbalancer/app/Testing/5a19403ecd42dc05",
    "startTime": "2024-12-17T21:27:26-08:00",
```

```
    "status": "ACTIVE",
    "zonalShiftId": "9ac9ec1e-1df1-0755-3dc5-8cf573cd9c38"
}
```

Batalkan pergeseran zona

Anda dapat membatalkan pergeseran zona dengan CLI dengan menggunakan `cancel-zonal-shift` perintah.

```
aws arc-zonal-shift cancel-zonal-shift \
       --zonal-shift-id 9ac9ec1e-1df1-0755-3dc5-8cf573cd9c38
```

```
{
    "awayFrom": "use1-az1",
    "comment": "Still shifting traffic away from use1-az1",
    "expiryTime": "2024-12-17T22:29:38-08:00",
    "resourceIdentifier": "arn:aws:elasticloadbalancing:us-
east-1:111122223333:loadbalancer/app/Testing/5a19403ecd42dc05",
    "startTime": "2024-12-17T21:27:26-08:00",
    "status": "CANCELED",
    "zonalShiftId": "9ac9ec1e-1df1-0755-3dc5-8cf573cd9c38"
}
```

Sumber daya yang didukung

Amazon Application Recovery Controller (ARC) saat ini mendukung pengaktifan sumber daya berikut untuk zonal shift dan zonal autoshift:

- Grup EC2 Auto Scaling Amazon
- Amazon Elastic Kubernetes Service
- Application Load Balancerdengan penyeimbangan beban lintas zona diaktifkan atau dinonaktifkan
- Penyeimbang Beban Jaringandengan penyeimbangan beban lintas zona diaktifkan atau dinonaktifkan

Untuk persyaratan khusus untuk Network Load Balancers dan Application Load Balancer, lihat topik tambahan di bagian ini.

Tinjau kondisi berikut untuk bekerja dengan pergeseran zona, pergeseran otomatis zona, dan sumber daya di ARC:

- Sumber daya harus aktif dan sepenuhnya disediakan untuk mengalihkan lalu lintas untuk itu. Sebelum Anda memulai pergeseran zona untuk sumber daya, periksa untuk memastikan bahwa itu adalah sumber daya terkelola di ARC. Misalnya, lihat daftar sumber daya terkelola di AWS Management Console, atau gunakan `get-managed-resource` operasi dengan pengenal sumber daya.
- Untuk memulai pergeseran zona dengan sumber daya, itu harus digunakan di Availability Zone dan Wilayah AWS di mana Anda memulai shift. Pastikan Anda memulai pergeseran zona di Wilayah yang sama dengan AZ yang ingin Anda geser, dan sumber daya tempat Anda mengalihkan lalu lintas berada di AZ dan Wilayah yang sama juga.
- Pastikan Anda memiliki izin IAM yang benar untuk menggunakan pergeseran zona dengan sumber daya. Untuk informasi selengkapnya, lihat IAM dan izin untuk pergeseran zona.
- Ketika Network Load Balancer atau Application Load Balancer dalam keadaan terbuka gagal, pergeseran zona tidak akan berpengaruh. Ini adalah perilaku yang diharapkan karena pergeseran zona tidak dapat memaksa AZ menjadi tidak sehat dan kemudian mengalihkan lalu lintas ke yang lain AZs di Wilayah ketika penyeimbang beban gagal terbuka. Untuk informasi selengkapnya, lihat failover DNS Menggunakan Route 53 untuk penyeimbang beban Anda di Panduan Pengguna Network Load Balancers dan Menggunakan failover DNS Route 53 untuk penyeimbang beban Anda di Panduan Pengguna Application Load Balancers.
- Jika beberapa penyeimbang beban meneruskan lalu lintas ke target yang sama, pergeseran zona pada penyeimbang beban yang diaktifkan lintas zona akan menurunkan kapasitas target untuk semua penyeimbang beban, bahkan jika mereka tidak digeser zona.

Grup EC2 Auto Scaling Amazon

Grup EC2 Auto Scaling Amazon berisi kumpulan EC2 instans Amazon yang diperlakukan sebagai pengelompokan logis untuk keperluan penskalaan dan pengelolaan otomatis. Grup Auto Scaling juga memungkinkan Anda menggunakan fitur Auto EC2 Scaling Amazon seperti penggantian pemeriksaan kesehatan dan kebijakan penskalaan. Baik mempertahankan jumlah instans dalam grup Auto Scaling dan penskalaan otomatis adalah fungsionalitas inti dari layanan Auto EC2 Scaling Amazon.

Menggunakan pergeseran zona untuk grup Auto Scaling

Untuk mengaktifkan pergeseran zona, gunakan salah satu metode berikut.

Console

Untuk mengaktifkan pergeseran zona pada grup baru (konsol)

1. Ikuti petunjuk di Buat grup Auto Scaling menggunakan template peluncuran dan selesaikan setiap langkah dalam prosedur, hingga langkah 10.
2. Pada halaman Integrasikan dengan layanan lain, untuk pergeseran zona ARC, pilih kotak centang untuk mengaktifkan pergeseran zona.
3. Untuk perilaku pemeriksaan Kesehatan, pilih Abaikan tidak sehat atau Ganti tidak sehat. Jika disetel `kereplace-unhealthy`, instance yang tidak sehat akan diganti di Availability Zone dengan pergeseran zona aktif. Jika disetel `keignore-unhealthy`, instance yang tidak sehat tidak akan diganti di Availability Zone dengan pergeseran zona aktif.
4. Lanjutkan dengan langkah-langkah di Buat grup Auto Scaling menggunakan template peluncuran.

AWS CLI

Untuk mengaktifkan pergeseran zona pada grup baru ()AWS CLI

Tambahkan parameter `--availability-zone-impairment-policy` ke perintah create-auto-scaling-group.

`--availability-zone-impairment-policy`Parameter memiliki dua opsi:

- ZonalShiftEnabled— Jika diatur `ketrue`, Auto Scaling mendaftarkan grup Auto Scaling dengan ARC zonal shift dan Anda dapat memulai, memperbarui, atau membatalkan pergeseran zona pada konsol ARC. Jika disetel `kefalse`, Auto Scaling membatalkan pendaftaran grup Auto Scaling dari pergeseran zona ARC. Anda harus sudah mengaktifkan zonal shift untuk disetel `kefalse`.
- ImpairedZoneHealthCheckBehavior— Jika disetel `kereplace-unhealthy`, instance yang tidak sehat akan diganti di Availability Zone dengan pergeseran zona aktif. Jika disetel `keignore-unhealthy`, instance yang tidak sehat tidak akan diganti di Availability Zone dengan pergeseran zona aktif.

Contoh berikut memungkinkan pergeseran zona pada grup Auto Scaling baru bernama. *my-asg*

```
aws autoscaling create-auto-scaling-group \
  --launch-template LaunchTemplateName=my-launch-template,Version='1' \
```

```
--auto-scaling-group-name my-asg \
--min-size 1 \
--max-size 10 \
--desired-capacity 5 \
--availability-zones us-east-1a us-east-1b us-east-1c \
--availability-zone-impairment-policy '{
    "ZonalShiftEnabled": true,
    "ImpairedZoneHealthCheckBehavior": IgnoreUnhealthy
  }'
```

Console

Untuk mengaktifkan pergeseran zona pada grup yang ada (konsol)

1. Buka EC2 konsol Amazon di https://console.aws.amazon.com/ec2/, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pada bilah navigasi di bagian atas layar, pilih tempat Wilayah AWS Anda membuat grup Auto Scaling.
3. Pilih kotak centang di samping grup Auto Scaling.

   Panel split terbuka di bagian bawah halaman.
4. Pada tab Integrasi, di bawah ARC zonal shift, pilih Edit.
5. Pilih kotak centang untuk mengaktifkan pergeseran zona.
6. Untuk perilaku pemeriksaan Kesehatan, pilih Abaikan tidak sehat atau Ganti tidak sehat. Jika disetel `kereplace-unhealthy`, instance yang tidak sehat akan diganti di Availability Zone dengan pergeseran zona aktif. Jika disetel `keignore-unhealthy`, instance yang tidak sehat tidak akan diganti di Availability Zone dengan pergeseran zona aktif.
7. Pilih Perbarui.

AWS CLI

Untuk mengaktifkan pergeseran zona pada grup yang ada ()AWS CLI

Tambahkan parameter `--availability-zone-impairment-policy` ke perintah update-auto-scaling-group.

`--availability-zone-impairment-policy`Parameter memiliki dua opsi:

- ZonalShiftEnabled— Jika diatur ketrue, Auto Scaling mendaftarkan grup Auto Scaling dengan ARC zonal shift dan Anda dapat memulai, memperbarui, atau membatalkan pergeseran zona pada konsol ARC. Jika disetel kefalse, Auto Scaling membatalkan pendaftaran grup Auto Scaling dari pergeseran zona ARC. Anda harus sudah mengaktifkan zonal shift untuk disetel kefalse.
- ImpairedZoneHealthCheckBehavior— Jika disetel kereplace-unhealthy, instance yang tidak sehat akan diganti di Availability Zone dengan pergeseran zona aktif. Jika disetel keignore-unhealthy, instance yang tidak sehat tidak akan diganti di Availability Zone dengan pergeseran zona aktif.

Contoh berikut memungkinkan pergeseran zona pada grup Auto Scaling yang ditentukan.

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-asg \
  --availability-zone-impairment-policy '{
      "ZonalShiftEnabled": true,
      "ImpairedZoneHealthCheckBehavior": IgnoreUnhealthy
    }'
```

Untuk memicu pergeseran zona, lihat Memulai, memperbarui, atau membatalkan pergeseran zona.

Cara kerja zonal shift untuk grup Auto Scaling

Misalkan Anda memiliki grup Auto Scaling dengan Availability Zone berikut:

- us-east-1a
- us-east-1b
- us-east-1c

Anda melihat kegagalan us-east-1a dan memicu pergeseran zona. Perilaku berikut terjadi ketika pergeseran zona dipicu. us-east-1a

- Penskalaan - Auto Scaling akan meluncurkan semua permintaan kapasitas baru di Availability Zone yang sehat us-east-1b (us-east-1cdan).
- Penskalaan dinamis — Auto Scaling akan memblokir kebijakan penskalaan agar tidak mengurangi kapasitas yang diinginkan. Auto Scaling tidak akan memblokir kebijakan penskalaan dari peningkatan kapasitas yang diinginkan.

- Penyegaran instans — Auto Scaling akan memperpanjang waktu habis untuk setiap proses penyegaran instans yang tertunda selama pergeseran zona aktif.

Pemilihan perilaku pemeriksaan kesehatan Zona Ketersediaan Gangguan	Perilaku pemeriksaan kesehatan
Ganti yang tidak sehat	Instance yang tampak tidak sehat akan diganti di semua Availability Zone (`us-east-1a` ,`us-east-1b` , dan`us-east-1c`).
Abaikan tidak sehat	Contoh yang tampak tidak sehat akan diganti `us-east-1b` dan`us-east-1c` . Instance tidak akan diganti di Availability Zone dengan active zonal shift ()`us-east-1a` .

Praktik terbaik untuk menggunakan pergeseran zona

Untuk menjaga ketersediaan tinggi untuk aplikasi Anda saat menggunakan zonal shift, kami merekomendasikan praktik terbaik berikut.

- Pantau EventBridge pemberitahuan untuk menentukan kapan ada peristiwa penurunan zona ketersediaan yang sedang berlangsung. Untuk informasi selengkapnya, lihat Mengotomatiskan EC2 Auto Scaling Amazon dengan Event Bridge.
- Gunakan kebijakan penskalaan dengan ambang batas yang sesuai untuk memastikan bahwa Anda memiliki kapasitas yang cukup untuk mentolerir hilangnya zona ketersediaan.
- Tetapkan kebijakan pemeliharaan instans dengan persentase sehat minimum 100. Dengan pengaturan ini, Auto Scaling menunggu instance baru siap digunakan sebelum menghentikan instance yang tidak sehat.

Untuk pelanggan prescaled, kami juga merekomendasikan yang berikut:

- Pilih Abaikan tidak sehat sebagai perilaku pemeriksaan kesehatan untuk zona ketersediaan yang terganggu karena Anda tidak perlu mengganti instance yang tidak sehat selama peristiwa gangguan.
- Gunakan pergeseran otomatis zona di ARC untuk grup Auto Scaling Anda. Kemampuan pergeseran otomatis zona Amazon Application Recovery Controller (ARC) memungkinkan AWS

untuk mengalihkan lalu lintas untuk sumber daya yang jauh dari zona ketersediaan saat AWS mendeteksi gangguan di zona ketersediaan. Untuk informasi selengkapnya, lihat Zonal autoshift di ARC di Panduan Pengembang Amazon Application Recovery Controller (ARC).

Untuk pelanggan dengan penyeimbang beban dinonaktifkan lintas zona, kami juga merekomendasikan:

- Gunakan balanced hanya untuk distribusi zona ketersediaan Anda.
- Jika Anda menggunakan pergeseran zona pada grup Auto Scaling dan penyeimbang beban Anda, pastikan untuk membatalkan pergeseran zona pada grup Auto Scaling Anda terlebih dahulu. Kemudian, tunggu sampai kapasitas seimbang di semua zona ketersediaan. sebelum Anda membatalkan pergeseran zona pada penyeimbang beban.
- Karena kemungkinan kapasitas yang tidak seimbang saat Anda mengaktifkan pergeseran zona dan Anda menggunakan penyeimbang beban dinonaktifkan lintas zona, Auto Scaling memiliki validasi tambahan. Jika Anda mengikuti praktik terbaik, Anda dapat mengetahui kemungkinan ini dengan memilih kotak centang di AWS Management Console atau menggunakan `skip-zonal-shift-validation` bendera di`CreateAutoScalingGroup`,`UpdateAutoScalingGroup`, atau. `AttachTrafficSources`

Amazon Elastic Kubernetes Service

Amazon EKS menyediakan fitur yang memungkinkan Anda membuat aplikasi Anda lebih tahan terhadap peristiwa seperti penurunan kesehatan atau gangguan Availability Zone (AZ). Saat menjalankan beban kerja di klaster Amazon EKS, Anda dapat lebih meningkatkan toleransi kesalahan lingkungan aplikasi dan pemulihan aplikasi menggunakan zonal shift atau zonal autoshift.

Menggunakan pergeseran zona untuk Amazon Elastic Kubernetes Service

Untuk mengaktifkan pergeseran zona, gunakan salah satu metode berikut Untuk informasi selengkapnya, lihat Aktifkan Amazon EKS Zonal Shift untuk menghindari gangguan Availability Zone.

Console

Untuk mengaktifkan pergeseran zona pada cluster Amazon EKS baru (Konsol)

1. Temukan nama dan Wilayah cluster Amazon EKS yang ingin Anda daftarkan dengan ARC.
2. Buka konsol Amazon EKS di https://console.aws.amazon.com/eks/rumah#/cluster.
3. Pilih klaster Anda.

4. Pada halaman Info cluster, pilih tab Ikhtisar.
5. Di bawah judul Zonal Shift, pilih tombol Kelola.
6. Pilih aktifkan atau nonaktifkan untuk EKS Zonal Shift.

AWS CLI

Untuk mengaktifkan pergeseran zona pada cluster Amazon EKS baru ()AWS CLI

- Masukkan perintah berikut:

```
aws eks create-cluster --name my-eks-cluster --role-
arn my-role-arn-to-create-cluster --resources-vpc-config
 subnetIds=string,string,securityGroupIds=string,string,endpointPublicAccess=boolean,end
 --zonal-shift-config enabled=true
```

Untuk mengaktifkan pergeseran zona pada kluster Amazon EKS yang ada ()AWS CLI

- Masukkan perintah berikut:

```
aws eks update-cluster-config --name my-eks-cluster --zonal-shift-config
 enabled=true
```

Anda dapat memicu pergeseran zona untuk kluster Amazon EKS, atau Anda dapat mengizinkan melakukannya AWS untuk Anda dengan mengaktifkan pergeseran otomatis zona. Setelah pergeseran zona klaster Amazon EKS diaktifkan dengan ARC, Anda dapat memicu pergeseran zona atau mengaktifkan pergeseran otomatis zona menggunakan Konsol ARC, AWS CLI, atau pergeseran zona dan pergeseran otomatis zona. APIs

Untuk informasi lebih lanjut tentang memicu pergeseran zona, lihat. Memulai, memperbarui, atau membatalkan pergeseran zona

Untuk informasi selengkapnya tentang mengaktifkan Amazon EKS dengan pergeseran zona, lihat topik Pelajari tentang ARC Zonal Shift di Amazon EKS di Panduan Pengguna Amazon Elastic Kubernetes Service.

Cara kerja zonal shift untuk Amazon Elastic Kubernetes Service

Selama pergeseran zona Amazon EKS, berikut ini akan secara otomatis terjadi:

- Semua node di AZ yang terkena dampak akan ditutup. Ini akan mencegah Kubernetes Scheduler menjadwalkan Pod baru ke node di AZ yang tidak sehat.
- Jika Anda menggunakan Grup Node Terkelola, penyeimbangan ulang Availability Zone akan ditangguhkan, dan Grup Auto Scaling (ASG) Anda akan diperbarui untuk memastikan bahwa node Amazon EKS Data Plane baru hanya diluncurkan dalam kondisi sehat. AZs
- Node di AZ yang tidak sehat tidak akan dihentikan dan Pod tidak akan diusir dari node ini. Ini untuk memastikan bahwa ketika pergeseran zona berakhir atau dibatalkan, lalu lintas Anda dapat dikembalikan dengan aman ke AZ yang masih memiliki kapasitas penuh.
- EndpointSlice Pengontrol akan menemukan semua titik akhir Pod di AZ yang rusak dan menghapusnya dari yang relevan EndpointSlices. Ini akan memastikan bahwa hanya titik akhir Pod yang sehat yang AZs ditargetkan untuk menerima lalu lintas jaringan. Ketika pergeseran zona dibatalkan atau kedaluwarsa, EndpointSlice pengontrol akan memperbarui EndpointSlices untuk menyertakan titik akhir di AZ yang dipulihkan.

Untuk informasi lebih lanjut, lihat blog AWS Containers.

Application Load Balancer

Menggunakan pergeseran zona untuk Application Load Balancers

Untuk menggunakan Application Load Balancers dengan pergeseran zona, Anda harus mengaktifkan integrasi pergeseran zona ARC dalam atribut Application Load Balancer. Application Load Balancer mendukung pergeseran zona dengan konfigurasi yang diaktifkan lintas zona atau lintas zona dinonaktifkan.

Sebelum Anda mengaktifkan integrasi ARC dan mulai menggunakan zonal shift, tinjau hal-hal berikut:

- Anda dapat memulai pergeseran zona untuk penyeimbang beban tertentu hanya untuk satu Availability Zone. Anda tidak dapat memulai pergeseran zona untuk beberapa Availability Zone.
- AWS secara proaktif menghapus alamat IP penyeimbang beban zonal dari DNS ketika beberapa masalah infrastruktur berdampak pada layanan. Selalu periksa kapasitas Availability Zone saat ini sebelum Anda memulai pergeseran zona.
- Ketika Application Load Balancer adalah target Network Load Balancer, selalu mulai pergeseran zona dari Network Load Balancer. Jika Anda memulai pergeseran zona dari Application Load Balancer, Network Load Balancer tidak mengenali shift dan terus mengirim lalu lintas ke Application Load Balancer.

Anda dapat memulai pergeseran zona untuk penyeimbang beban di konsol Elastic Load Balancing (Wilayah AWS sebagian besar) atau di konsol ARC.

Console

Untuk mengaktifkan pergeseran zona pada penyeimbang beban (Konsol)

1. Buka EC2 konsol Amazon di https://console.aws.amazon.com/ec2/.
2. Pada halaman Navigasi, di bawah Load Balancing, pilih Load Balancers.
3. Pilih nama Application Load Balancer.
4. Pada tab Atribut, pilih Edit.
5. Di bawah konfigurasi perutean Availability Zone, atur integrasi pergeseran zona ARC ke Aktifkan.
6. Pilih Simpan.

AWS CLI

Untuk mengaktifkan pergeseran zona pada penyeimbang beban ()AWS CLI

- Masukkan perintah berikut:

```
aws elbv2 modify-load-balancer-attributes --load-balancer-arn my-alb-arn --
attributes Key=zonal_shift.config.enabled,Value=true
```

Untuk informasi lebih lanjut tentang memicu pergeseran zona, lihat. Memulai, memperbarui, atau membatalkan pergeseran zona

Anda dapat menggunakan `keepalive` opsi untuk mengonfigurasi berapa lama koneksi berlanjut. Untuk informasi selengkapnya, lihat durasi keepalive klien HTTP di Panduan Pengguna Application Load Balancer. Secara default, Application Load Balancers menetapkan nilai durasi keepalive klien HTTP menjadi 3600 detik, atau 1 jam. Kami menyarankan agar Anda menurunkan nilai agar sesuai dengan sasaran waktu pemulihan untuk aplikasi Anda, misalnya, 300 detik. Saat Anda memilih waktu durasi keepalive klien HTTP, pertimbangkan bahwa nilai ini adalah pertukaran antara menghubungkan kembali lebih sering secara umum, yang dapat memengaruhi latensi, dan lebih cepat memindahkan semua klien dari AZ atau Wilayah yang terganggu.

Cara kerja zonal shift untuk Application Load Balancers

Ketika pergeseran zona dimulai pada Application Load Balancer dengan penyeimbangan beban lintas zona diaktifkan, semua lalu lintas ke target diblokir di zona ketersediaan yang terpengaruh, dan menghapus alamat IP zona dari DNS.

Untuk informasi selengkapnya lihat Integrasi untuk Application Load Balancer Anda di Panduan Pengguna Application Load Balancer.

Penyeimbang Beban Jaringan

Menggunakan pergeseran zona untuk Network Load Balancers

Untuk menggunakan Network Load Balancers dengan pergeseran zona, Anda harus mengaktifkan integrasi pergeseran zona ARC di atribut Network Load Balancer. Network Load Balancer mendukung pergeseran zona dengan konfigurasi yang diaktifkan lintas zona atau dinonaktifkan lintas zona.

Anda dapat memilih sumber daya mana yang akan dipilih untuk menggunakan pergeseran zona dan pergeseran otomatis zona, dan kapan Anda ingin gagal menjauh dari Zona Ketersediaan yang terganggu. Baik Network Load Balancer yang menghadap ke internet dan internal didukung.

Untuk mengaktifkan pergeseran zona untuk Network Load Balancer yang diaktifkan lintas zona, semua grup target yang terpasang pada penyeimbang beban harus memenuhi persyaratan berikut.

- Penyeimbangan beban lintas zona harus diaktifkan, atau disetel ke. `use_load_balancer_configuration`
  - Untuk informasi selengkapnya tentang penyeimbangan beban lintas zona kelompok target, lihat Penyeimbangan beban lintas zona untuk grup target.
- Protokol grup target harus TCP atau TLS.
  - Untuk informasi selengkapnya tentang protokol grup target Network Load Balancer, lihat Konfigurasi perutean.
- Pengakhiran koneksi untuk target yang tidak sehat harus dinonaktifkan.
  - Untuk informasi selengkapnya tentang penghentian koneksi grup target, lihat Pengakhiran koneksi untuk target yang tidak sehat.
- Kelompok sasaran tidak boleh memiliki Application Load Balancer sebagai target.
  - Untuk informasi selengkapnya tentang Application Load Balancer sebagai target, lihat Menggunakan Application Load Balancer sebagai target Network Load Balancer.

Anda dapat memulai pergeseran zona untuk Network Load Balancer dengan menggunakan AWS CLI, konsol, AWS atau widget Elastic Load Balancing. Ketika Application Load Balancer adalah target Network Load Balancer, Anda harus memulai pergeseran zona dari Network Load Balancer. Jika Anda memulai pergeseran zona dari Application Load Balancer, Network Load Balancer tidak akan berhenti mengirimkan lalu lintas ke Application Load Balancer dan targetnya.

Console

Untuk mengaktifkan pergeseran zona pada penyeimbang beban (Konsol)

1. Buka EC2 konsol Amazon di https://console.aws.amazon.com/ec2/.
2. Pada halaman Navigasi, di bawah Load Balancing, pilih Load Balancers.
3. Pilih nama Network Load Balancer.
4. Pada tab Atribut, pilih Edit.
5. Di bawah konfigurasi perutean Availability Zone, atur integrasi pergeseran zona ARC ke Aktifkan.
6. Pilih Simpan.

AWS CLI

Untuk mengaktifkan pergeseran zona pada penyeimbang beban ()AWS CLI

- Masukkan perintah berikut:

```
aws elbv2 modify-load-balancer-attributes --load-balancer-arn my-nlb-arn --
attributes Key=zonal_shift.config.enabled,Value=true
```

Untuk informasi lebih lanjut tentang memicu pergeseran zona, lihat. Memulai, memperbarui, atau membatalkan pergeseran zona

Cara kerja zonal shift untuk Network Load Balancers

ARC menginduksi kegagalan pemeriksaan kesehatan untuk Network Load Balancer yang terdaftar sehingga node Network Load Balancer di AZ yang rusak dihapus dari DNS saat Anda memicu pergeseran zona. Network Load Balancer akan menonaktifkan target di zona yang terkena dampak sehingga mereka berhenti menerima lalu lintas, dan Elastic Load Balancing memperlakukan target ini sebagai target yang dinonaktifkan oleh pergeseran zona. Target di negara cacat terus menerima

pemeriksaan kesehatan. Ketika target sehat dan pergeseran zona berakhir (atau dibatalkan), perutean ke target di zona yang sebelumnya terganggu dilanjutkan.

Selama pergeseran zona pada Network Load Balancers dengan penyeimbangan beban lintas zona diaktifkan, alamat IP penyeimbang beban zonal dihapus dari DNS. Koneksi yang ada ke target di Zona Ketersediaan yang terganggu tetap ada hingga ditutup secara organik, sementara koneksi baru tidak lagi diarahkan ke target di Zona Ketersediaan yang terganggu.

Untuk informasi selengkapnya lihat topik Zonal Shift untuk Network Load Balancer Anda di Panduan Pengguna Network Load Balancer.

Memulai, memperbarui, atau membatalkan pergeseran zona

Bagian ini menyediakan prosedur untuk bekerja dengan pergeseran zona, termasuk memulai pergeseran zona dan membatalkan pergeseran zona.

Memulai pergeseran zona

Langkah-langkah di bagian ini menjelaskan cara memulai pergeseran zona yang dimulai pelanggan di konsol Amazon Application Recovery Controller (ARC). Untuk bekerja dengan pergeseran zona secara terprogram, lihat Panduan Referensi API Zonal Shift.

Selain memulai pergeseran zona di ARC, Anda juga dapat memulai pergeseran zona untuk penyeimbang beban di konsol Elastic Load Balancing (di Wilayah yang didukung). Untuk informasi selengkapnya, lihat Pergeseran zona di Panduan Pengguna Elastic Load Balancing.

Untuk memulai pergeseran zona

1. Buka konsol ARC dihttps://console.aws.amazon.com/route53recovery/home#/dashboard.
2. Di bawah Multi-AZ, pilih Zonal shift.
3. Pada halaman Zonal shift, pilih Mulai pergeseran zona.
4. Pilih Availability Zone yang ingin Anda alihkan lalu lintas.
5. Pilih sumber daya yang didukung dari tabel Resources untuk mengalihkan lalu lintas.
6. Untuk Mengatur kedaluwarsa pergeseran zona, pilih atau masukkan kedaluwarsa untuk pergeseran zona. Pergeseran zona dapat diatur untuk aktif pada awalnya selama 1 menit atau hingga tiga hari (72 jam).

   Semua pergeseran zona bersifat sementara. Anda harus menetapkan kedaluwarsa, tetapi Anda dapat memperbarui shift aktif nanti untuk menetapkan periode kedaluwarsa baru hingga tiga hari.

7. Masukkan komentar. Anda dapat memperbarui pergeseran zona nanti untuk mengedit komentar, jika Anda mau.
8. Pilih kotak centang untuk mengetahui bahwa memulai pergeseran zona akan mengurangi kapasitas yang tersedia untuk aplikasi Anda dengan mengalihkan lalu lintas dari Availability Zone.
9. Pilih Mulai.

Memperbarui atau membatalkan pergeseran zona

Langkah-langkah di bagian ini menjelaskan cara memperbarui pergeseran zona yang Anda mulai, atau membatalkan pergeseran zona, di konsol Amazon Application Recovery Controller (ARC). Untuk bekerja dengan pergeseran zona secara terprogram, lihat Panduan Referensi API Zonal Shift.

Anda dapat memperbarui pergeseran zona untuk menetapkan kedaluwarsa baru, atau mengedit atau mengganti komentar untuk pergeseran zona. Anda dapat membatalkan pergeseran zona kapan saja sebelum kedaluwarsa.

Anda dapat membatalkan pergeseran zona yang Anda mulai, atau pergeseran zona yang AWS dimulai untuk sumber daya untuk latihan yang dijalankan untuk pergeseran otomatis zona. Untuk mempelajari lebih lanjut tentang pergeseran latihan dalam pergeseran otomatis zona, lihat. Bagaimana zonal autoshift dan praktek berjalan bekerja

Untuk memperbarui pergeseran zona

1. Buka konsol ARC dihttps://console.aws.amazon.com/route53recovery/home#/dashboard.
2. Di bawah Multi-AZ, pilih Zonal shift.
3. Pilih pergeseran zona yang ingin Anda perbarui, lalu pilih Perbarui pergeseran zona.
4. Untuk Mengatur kedaluwarsa pergeseran zona, pilih atau masukkan kedaluwarsa secara opsional.
5. Untuk Komentar, secara opsional edit komentar yang ada atau masukkan komentar baru.
6. Pilih Perbarui.

Untuk membatalkan pergeseran zona

1. Buka konsol ARC dihttps://console.aws.amazon.com/route53recovery/home#/dashboard.
2. Di bawah Multi-AZ, pilih Zonal shift.

3. Pilih pergeseran zona yang ingin Anda batalkan, lalu pilih Batalkan pergeseran zona.
4. Pada dialog modal konfirmasi, pilih Konfirmasi.

Pencatatan dan pemantauan untuk pergeseran zona di Amazon Application Recovery Controller (ARC)

Anda dapat menggunakan AWS CloudTrail untuk memantau pergeseran zona di Amazon Application Recovery Controller (ARC), untuk menganalisis pola dan membantu memecahkan masalah.

Topik

- Mencatat panggilan API shift zona menggunakan AWS CloudTrail

Mencatat panggilan API shift zona menggunakan AWS CloudTrail

Zonal shift untuk ARC terintegrasi dengan AWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan di ARC. CloudTrail menangkap semua panggilan API untuk pergeseran zona sebagai peristiwa. Panggilan yang diambil termasuk panggilan dari konsol ARC dan panggilan kode ke operasi ARC API untuk pergeseran zona.

Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail acara secara terus menerus ke bucket Amazon S3, termasuk peristiwa untuk pergeseran zona. Jika Anda tidak mengonfigurasi jejak, Anda masih dapat melihat peristiwa terbaru di CloudTrail konsol dalam Riwayat acara.

Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat ke ARC untuk pergeseran zona, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan.

Untuk mempelajari selengkapnya CloudTrail, lihat Panduan AWS CloudTrail Pengguna.

Informasi pergeseran zona di CloudTrail

CloudTrail diaktifkan pada Akun AWS saat Anda membuat akun. Ketika aktivitas terjadi di ARC untuk pergeseran zona, aktivitas tersebut dicatat dalam suatu CloudTrail peristiwa bersama dengan peristiwa AWS layanan lainnya dalam riwayat Peristiwa. Anda dapat melihat, mencari, dan mengunduh acara terbaru di situs Anda Akun AWS. Untuk informasi selengkapnya, lihat Bekerja dengan riwayat CloudTrail Acara.

Untuk catatan acara yang sedang berlangsung di Anda Akun AWS, termasuk acara untuk pergeseran zona di ARC, buat jejak. Jejak memungkinkan CloudTrail untuk mengirimkan file log ke bucket Amazon S3. Secara default, saat Anda membuat jejak di konsol, jejak tersebut berlaku untuk semua Wilayah AWS. Jejak mencatat peristiwa dari semua Wilayah di AWS partisi dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat mengonfigurasi AWS layanan lain, untuk menganalisis lebih lanjut dan menindaklanjuti data peristiwa yang dikumpulkan dalam CloudTrail log. Untuk informasi selengkapnya, lihat berikut:

- Gambaran umum untuk membuat jejak
- CloudTrail layanan dan integrasi yang didukung
- Mengonfigurasi notifikasi Amazon SNS untuk CloudTrail
- Menerima file CloudTrail log dari beberapa wilayah dan Menerima file CloudTrail log dari beberapa akun

Semua tindakan ARC dicatat oleh CloudTrail dan didokumentasikan dalam Panduan Referensi API Kontrol Perutean untuk Pengontrol Pemulihan Aplikasi Amazon. Misalnya, panggilan ke `StartZonalShift` dan `ListManagedResources` tindakan menghasilkan entri dalam file CloudTrail log.

Setiap entri peristiwa atau log berisi informasi tentang siapa yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan berikut ini:

- Apakah permintaan itu dibuat dengan kredenal pengguna root atau AWS Identity and Access Management (IAM).
- Apakah permintaan tersebut dibuat dengan kredensial keamanan sementara untuk satu peran atau pengguna gabungan.
- Apakah permintaan itu dibuat oleh AWS layanan lain.

Untuk informasi selengkapnya, lihat Elemen userIdentity CloudTrail .

Melihat peristiwa ARC dalam sejarah acara

CloudTrail memungkinkan Anda melihat peristiwa terbaru dalam riwayat Acara. Untuk informasi selengkapnya, lihat Bekerja dengan riwayat CloudTrail Acara di Panduan AWS CloudTrail Pengguna.

Memahami entri file log shift zona

Trail adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai file log ke bucket Amazon S3 yang Anda tentukan. CloudTrail file log berisi satu atau lebih entri log. Peristiwa mewakili permintaan tunggal dari sumber manapun dan mencakup informasi tentang tindakan yang diminta, tanggal dan waktu tindakan, parameter permintaan, dan sebagainya. CloudTrail file log bukanlah jejak tumpukan yang diurutkan dari panggilan API publik, jadi file tersebut tidak muncul dalam urutan tertentu.

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan `ListManagedResources` tindakan untuk pergeseran zona.


```
{
      "eventVersion": "1.08",
      "userIdentity": {
        "type": "AssumedRole",
        "principalId": "A1B2C3D4E5F6G7EXAMPLE",
        "arn": "arn:aws:iam::111122223333:role/admin",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
          "sessionIssuer": {
            "type": "Role",
            "principalId": "AROA33L3W36EXAMPLE",
            "arn": "arn:aws:iam::111122223333:role/admin",
            "accountId": "111122223333",
            "userName": "EXAMPLENAME"
          },
          "webIdFederationData": {},
          "attributes": {
            "creationDate": "2022-11-14T16:01:51Z",
            "mfaAuthenticated": "false"
          }
        }
      },
      "eventTime": "2022-11-14T16:14:41Z",
      "eventSource": "arc-zonal-shift.amazonaws.com",
      "eventName": "ListManagedResources",
      "awsRegion": "us-west-2",
      "sourceIPAddress": "192.0.2.50",
      "userAgent": "Boto3/1.17.101 Python/3.8.10 Linux/4.14.231-180.360.amzn2.x86_64
 exec-env/AWS_Lambda_python3.8 Botocore/1.20.102",
      "requestParameters": null,
```

```
      "responseElements": null,
      "requestID": "VGXG4ZUE7UZTVCMTJGIAF_EXAMPLE",
      "eventID": "4b5c42df-1174-46c8-be99-d67_EXAMPLE",
      "readOnly": true,
      "eventType": "AwsApiCall",
      "managementEvent": true,
      "recipientAccountId": "111122223333"
      "eventCategory": "Management"
      }
    }
```

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan `StartZonalShift` tindakan dengan pengecualian konflik untuk pergeseran zona.

```
{
      "eventVersion": "1.08",
      "userIdentity": {
        "type": "AssumedRole",
        "principalId": "A1B2C3D4E5F6G7EXAMPLE",
        "arn": "arn:aws:iam::111122223333:role/admin",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
          "sessionIssuer": {
            "type": "Role",
            "principalId": "AROA33L3W36EXAMPLE",
            "arn": "arn:aws:iam::111122223333:role/admin",
            "accountId": "111122223333",
            "userName": "EXAMPLENAME"
          },
          "webIdFederationData": {},
          "attributes": {
            "creationDate": "2022-11-14T16:01:51Z",
            "mfaAuthenticated": "false"
          }
        }
      },
      "eventTime": "2022-11-14T16:10:38Z",
      "eventSource": "arc-zonal-shift.amazonaws.com",
      "eventName": "StartZonalShift",
     "awsRegion": "us-west-2",
      "sourceIPAddress": "192.0.2.50",
```

```
      "userAgent": "Boto3/1.17.101 Python/3.8.10 Linux/4.14.231-180.360.amzn2.x86_64
 exec-env/AWS_Lambda_python3.8 Botocore/1.20.102",
      "errorCode": "ConflictException",
      "errorMessage": "There's already an active zonal shift for that resource
 identifier: 'arn:aws:testservice:us-west-2:077059137270:testResource/456apples'.
 Active zonal shift: 'bac23b74-176e-c073-de8f-484ca508910f'",
      "requestParameters": {
        "resourceIdentifier": "arn:aws:testservice:us-
west-2:077059137270:testResource/456apples",
        "awayFrom": "usw2-az1",
        "expiresIn": "2m",
        "comment": "HIDDEN_FOR_SECURITY_REASONS"
      },
      "responseElements": null,
      "requestID": "OP4OYXZ54HUPMIPGWH_EXAMPLE",
      "eventID": "0bca6660-e999-43a5-9008-EXAMPLE",
      "readOnly": false,
      "eventType": "AwsApiCall",
      "managementEvent": true,
      "recipientAccountId": "111122223333"
      "eventCategory": "Management"
      }
    }
```


Identity and Access Management untuk pergeseran zona di ARC

AWS Identity and Access Management (IAM) adalah Layanan AWS yang membantu administrator mengontrol akses ke AWS sumber daya dengan aman. Administrator IAM mengontrol siapa yang dapat diautentikasi (masuk) dan diberi wewenang (memiliki izin) untuk menggunakan sumber daya ARC. IAM adalah Layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

Daftar Isi

Bagaimana zonal shift bekerja dengan IAM

Sebelum Anda menggunakan IAM untuk mengelola akses ke pergeseran zona di Amazon Application Recovery Controller (ARC), pelajari fitur IAM apa yang tersedia untuk digunakan dengan zonal shift.

Fitur IAM yang dapat Anda gunakan dengan zonal shift

Fitur IAM	Dukungan pergeseran zona
Kebijakan berbasis identitas	Ya
Kebijakan berbasis sumber daya	Tidak
Tindakan kebijakan	Ya
Sumber daya kebijakan	Ya
Kunci kondisi kebijakan	Ya
ACLs	Tidak
ABAC (tanda dalam kebijakan)	Parsial
Kredensial sementara	Ya
Izin principal	Ya
Peran layanan	Tidak
Peran terkait layanan	Ya

Untuk mendapatkan tampilan tingkat tinggi dan keseluruhan tentang cara kerja AWS layanan dengan sebagian besar fitur IAM, lihat AWS layanan yang bekerja dengan IAM di Panduan Pengguna IAM.

Kebijakan berbasis identitas untuk ARC

Mendukung kebijakan berbasis identitas: Ya

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, seperti pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana,

dan berdasarkan kondisi seperti apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Tentukan izin IAM kustom dengan kebijakan terkelola pelanggan dalam Panduan Pengguna IAM.

Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan secara spesifik apakah tindakan dan sumber daya diizinkan atau ditolak, serta kondisi yang menjadi dasar dikabulkan atau ditolaknya tindakan tersebut. Anda tidak dapat menentukan secara spesifik prinsipal dalam sebuah kebijakan berbasis identitas karena prinsipal berlaku bagi pengguna atau peran yang melekat kepadanya. Untuk mempelajari semua elemen yang dapat Anda gunakan dalam kebijakan JSON, lihat Referensi elemen kebijakan JSON IAM dalam Panduan Pengguna IAM.

Untuk melihat contoh kebijakan berbasis identitas ARC, lihat. Contoh kebijakan berbasis identitas di Amazon Application Recovery Controller (ARC)

Kebijakan berbasis sumber daya dalam ARC

Mendukung kebijakan berbasis sumber daya: Tidak

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu.

Tindakan kebijakan untuk pergeseran zona

Mendukung tindakan kebijakan: Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen `Action` dari kebijakan JSON menjelaskan tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam sebuah kebijakan. Tindakan kebijakan biasanya memiliki nama yang sama dengan operasi AWS API terkait. Ada beberapa pengecualian, misalnya tindakan hanya izin yang tidak memiliki operasi API yang cocok. Ada juga beberapa operasi yang memerlukan beberapa tindakan dalam suatu kebijakan. Tindakan tambahan ini disebut tindakan dependen.

Sertakan tindakan dalam kebijakan untuk memberikan izin untuk melakukan operasi terkait.

Untuk melihat daftar tindakan ARC untuk pergeseran zona, lihat Tindakan yang ditentukan oleh Amazon Route 53 Zonal Shift di Referensi Otorisasi Layanan.

Tindakan kebijakan di ARC untuk pergeseran zona menggunakan awalan berikut sebelum tindakan:

```
arc-zonal-shift
```

Untuk menetapkan secara spesifik beberapa tindakan dalam satu pernyataan, pisahkan tindakan tersebut dengan koma. Misalnya, berikut ini:

```
"Action": [
      "arc-zonal-shift:action1",
      "arc-zonal-shift:action2"
         ]
```

Anda dapat menentukan beberapa tindakan menggunakan wildcard (*). Sebagai contoh, untuk menentukan semua tindakan yang dimulai dengan kata `Describe`, sertakan tindakan berikut:

```
"Action": "arc-zonal-shift:Describe*"
```

Untuk melihat contoh kebijakan berbasis identitas ARC untuk pergeseran zona, lihat. Contoh kebijakan berbasis identitas untuk pergeseran zona di ARC

Sumber daya kebijakan untuk pergeseran zona

Mendukung sumber daya kebijakan: Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen kebijakan JSON `Resource` menentukan objek yang menjadi target penerapan tindakan. Pernyataan harus menyertakan elemen `Resource` atau `NotResource`. Praktik terbaiknya, tentukan sumber daya menggunakan Amazon Resource Name (ARN). Anda dapat melakukan ini untuk tindakan yang mendukung jenis sumber daya tertentu, yang dikenal sebagai izin tingkat sumber daya.

Untuk tindakan yang tidak mendukung izin di tingkat sumber daya, misalnya operasi pencantuman, gunakan wildcard (*) untuk menunjukkan bahwa pernyataan tersebut berlaku untuk semua sumber daya.

```
"Resource": "*"
```

Untuk melihat daftar jenis sumber daya dan mereka ARNs, dan tindakan yang dapat Anda tentukan dengan ARN dari setiap sumber daya, lihat topik berikut di Referensi Otorisasi Layanan:

- Tindakan yang ditentukan oleh Amazon Route 53 - Zonal Shift

Untuk melihat tindakan dan sumber daya yang dapat Anda gunakan dengan kunci kondisi, lihat topik berikut di Referensi Otorisasi Layanan:

- Kunci kondisi ditentukan oleh Amazon Route 53 - Zonal Shift

Untuk melihat contoh kebijakan berbasis identitas ARC untuk pergeseran zona, lihat. Contoh kebijakan berbasis identitas untuk pergeseran zona di ARC

Kunci kondisi kebijakan untuk pergeseran zona

Mendukung kunci kondisi kebijakan khusus layanan: Yes

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen `Condition` (atau blok `Condition`) akan memungkinkan Anda menentukan kondisi yang menjadi dasar suatu pernyataan berlaku. Elemen `Condition` bersifat opsional. Anda dapat membuat ekspresi bersyarat yang menggunakan operator kondisi, misalnya sama dengan atau kurang dari, untuk mencocokkan kondisi dalam kebijakan dengan nilai-nilai yang diminta.

Jika Anda menentukan beberapa elemen `Condition` dalam sebuah pernyataan, atau beberapa kunci dalam elemen `Condition` tunggal, maka AWS akan mengevaluasinya menggunakan operasi `AND` logis. Jika Anda menentukan beberapa nilai untuk satu kunci kondisi, AWS mengevaluasi kondisi menggunakan `OR` operasi logis. Semua kondisi harus dipenuhi sebelum izin pernyataan diberikan.

Anda juga dapat menggunakan variabel placeholder saat menentukan kondisi. Sebagai contoh, Anda dapat memberikan izin kepada pengguna IAM untuk mengakses sumber daya hanya jika izin tersebut mempunyai tanda yang sesuai dengan nama pengguna IAM mereka. Untuk informasi selengkapnya, lihat Elemen kebijakan IAM: variabel dan tanda dalam Panduan Pengguna IAM.

AWS mendukung kunci kondisi global dan kunci kondisi khusus layanan. Untuk melihat semua kunci kondisi AWS global, lihat kunci konteks kondisi AWS global di Panduan Pengguna IAM.

Untuk melihat daftar tombol kondisi pergeseran zona, lihat topik berikut di Referensi Otorisasi Layanan:

- Kunci kondisi ditentukan oleh Amazon Route 53 - Zonal Shift

Untuk melihat tindakan dan sumber daya yang dapat Anda gunakan dengan kunci kondisi, lihat topik berikut di Referensi Otorisasi Layanan:

- Tindakan yang ditentukan oleh Amazon Route 53 - Zonal Shift
- Jenis sumber daya yang ditentukan oleh Amazon Route 53 - Zonal Shift

Untuk melihat contoh kebijakan berbasis identitas ARC untuk pergeseran zona, lihat. Contoh kebijakan berbasis identitas untuk pergeseran zona di ARC

Daftar kontrol akses (ACLs) di ARC

Mendukung ACLs: Tidak

Access control lists (ACLs) mengontrol prinsipal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACLs mirip dengan kebijakan berbasis sumber daya, meskipun mereka tidak menggunakan format dokumen kebijakan JSON.

Kontrol akses berbasis atribut (ABAC) dengan ARC

Mendukung ABAC (tag dalam kebijakan): Sebagian

Kontrol akses berbasis atribut (ABAC) adalah strategi otorisasi yang menentukan izin berdasarkan atribut. Dalam AWS, atribut ini disebut tag. Anda dapat melampirkan tag ke entitas IAM (pengguna atau peran) dan ke banyak AWS sumber daya. Penandaan ke entitas dan sumber daya adalah langkah pertama dari ABAC. Kemudian rancanglah kebijakan ABAC untuk mengizinkan operasi ketika tanda milik prinsipal cocok dengan tanda yang ada di sumber daya yang ingin diakses.

ABAC sangat berguna di lingkungan yang berkembang dengan cepat dan berguna di situasi saat manajemen kebijakan menjadi rumit.

Untuk mengendalikan akses berdasarkan tanda, berikan informasi tentang tanda di elemen kondisi dari kebijakan menggunakan kunci kondisi `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`*, atau `aws:TagKeys`.

Jika sebuah layanan mendukung ketiga kunci kondisi untuk setiap jenis sumber daya, nilainya adalah Ya untuk layanan tersebut. Jika suatu layanan mendukung ketiga kunci kondisi untuk hanya beberapa jenis sumber daya, nilainya adalah Parsial.

Untuk informasi selengkapnya tentang ABAC, lihat Tentukan izin dengan otorisasi ABAC dalam Panduan Pengguna IAM. Untuk melihat tutorial yang menguraikan langkah-langkah pengaturan ABAC, lihat Menggunakan kontrol akses berbasis atribut (ABAC) dalam Panduan Pengguna IAM.

ARC mencakup dukungan sebagian berikut untuk ABAC:

- Pergeseran zona mendukung ABAC untuk sumber daya terkelola yang terdaftar di ARC untuk pergeseran zona. Untuk informasi selengkapnya tentang ABAC untuk Network Load Balancer dan sumber daya yang dikelola Application Load Balancer, lihat ABAC dengan Elastic Load Balancing dalam Panduan Pengguna Elastic Load Balancing.

Menggunakan kredensi sementara dengan ARC

Mendukung kredensial sementara: Ya

Beberapa Layanan AWS tidak berfungsi saat Anda masuk menggunakan kredensi sementara. Untuk informasi tambahan, termasuk yang Layanan AWS bekerja dengan kredensi sementara, lihat Layanan AWS yang bekerja dengan IAM di Panduan Pengguna IAM.

Anda menggunakan kredensi sementara jika Anda masuk AWS Management Console menggunakan metode apa pun kecuali nama pengguna dan kata sandi. Misalnya, ketika Anda mengakses AWS menggunakan tautan masuk tunggal (SSO) perusahaan Anda, proses tersebut secara otomatis membuat kredensitas sementara. Anda juga akan secara otomatis membuat kredensial sementara ketika Anda masuk ke konsol sebagai seorang pengguna lalu beralih peran. Untuk informasi selengkapnya tentang peralihan peran, lihat Beralih dari pengguna ke peran IAM (konsol) dalam Panduan Pengguna IAM.

Anda dapat membuat kredenal sementara secara manual menggunakan API AWS CLI atau AWS . Anda kemudian dapat menggunakan kredensi sementara tersebut untuk mengakses. AWS AWS merekomendasikan agar Anda secara dinamis menghasilkan kredensi sementara alih-alih menggunakan kunci akses jangka panjang. Untuk informasi selengkapnya, lihat Kredensial keamanan sementara di IAM.

Izin utama lintas layanan untuk ARC

Mendukung sesi akses maju (FAS): Ya

Saat Anda menggunakan entitas IAM (pengguna atau peran) untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Kebijakan memberikan izin kepada principal. Saat Anda menggunakan beberapa layanan, Anda mungkin melakukan tindakan yang kemudian memicu tindakan lain di layanan yang berbeda. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut.

Untuk melihat apakah suatu tindakan memerlukan tindakan dependen tambahan dalam kebijakan, lihat topik berikut di Referensi Otorisasi Layanan:

- Amazon Route 53 Pergeseran Zonal

Peran layanan untuk ARC

Mendukung peran layanan: Tidak

Peran layanan adalah peran IAM yang diambil oleh sebuah layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, mengubah, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat Buat sebuah peran untuk mendelegasikan izin ke Layanan AWS dalam Panduan pengguna IAM.

Peran terkait layanan untuk ARC

Mendukung peran terkait layanan: Ya

Peran terkait layanan adalah jenis peran layanan yang ditautkan ke. Layanan AWS Layanan tersebut dapat menjalankan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Anda Akun AWS dan dimiliki oleh layanan. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.

Pergeseran zona tidak menggunakan peran terkait layanan.

IAM dan izin untuk pergeseran zona

Bagian ini memberikan informasi tambahan tentang cara kerja izin untuk fitur zonal shift di Amazon Application Recovery Controller (ARC), terutama jika Anda bekerja dengan fitur dari AWS layanan lain, seperti Elastic Load Balancing. Untuk mempelajari tentang cara kerja fitur ARC dengan IAM dan izin secara umum, tinjau informasi dalam topik ikhtisar,. Identity and Access Management untuk pergeseran zona di ARC

Zonal shift mendukung Application Load Balancers, Network Load Balancers, grup Amazon Auto EC2 Scaling, dan Amazon EKS. Anda dapat menggunakan kunci kondisi IAM untuk membuat cakupan

kebijakan izin IAM ke sumber daya ini. Berikut ini adalah contoh kebijakan menggunakan kunci kondisi dengan beberapa sumber daya dari berbagai jenis:

```
{
    "Condition": {
        "StringLike": {
            "arc-zonal-shift:ResourceIdentifier": [
                "arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/net/
*",
                "arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/app/
*",
                "arn:aws:eks:us-east-1:123456789012:cluster/*"
            ]
        }
    },
    "Action": [
        "arc-zonal-shift:StartZonalShift"
    ],
    "Resource": "*",
    "Effect": "Allow"
}
```

Untuk informasi selengkapnya, lihat Sumber daya yang didukung.

Selain izin yang diuraikan dalam topik ikhtisar IAM, berikut ini berlaku untuk pergeseran zona untuk IAM dan izin:

- Pastikan Anda memiliki izin yang diperlukan untuk bekerja dengan pergeseran zona di ARC. Untuk informasi selengkapnya, lihat akses konsol pergeseran zona dan akses operasi pergeseran zona.
- Anda tidak perlu menambahkan izin Elastic Load Balancing tambahan dengan IAM untuk bekerja dengan pergeseran zona untuk sumber daya penyeimbang beban terkelola di akun Anda di ARC.
- Kebijakan AWS terkelola yang menyediakan akses penuh untuk Elastic Load Balancing mencakup izin untuk bekerja dengan shift zona. Jika Anda menggunakan kebijakan AWS terkelola untuk akses Elastic Load Balancing, Anda tidak memerlukan izin tambahan di IAM untuk pergeseran zona untuk memulai pergeseran zona untuk penyeimbang beban atau bekerja dengan di konsol Elastic Load Balancing. Untuk informasi selengkapnya, lihat kebijakan AWS terkelola untuk Elastic Load Balancing.

Contoh kebijakan berbasis identitas untuk pergeseran zona di ARC

Secara default, pengguna dan peran tidak memiliki izin untuk membuat atau memodifikasi sumber daya ARC. Mereka juga tidak dapat melakukan tugas dengan menggunakan AWS Management Console, AWS Command Line Interface (AWS CLI), atau AWS API. Untuk memberikan izin kepada pengguna untuk melakukan tindakan di sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM. Administrator kemudian dapat menambahkan kebijakan IAM ke peran, dan pengguna dapat mengambil peran.

Untuk mempelajari cara membuat kebijakan berbasis identitas IAM dengan menggunakan contoh dokumen kebijakan JSON ini, lihat Membuat kebijakan IAM (konsol) di Panduan Pengguna IAM.

Untuk detail tentang tindakan dan jenis sumber daya yang ditentukan oleh ARC, termasuk format ARNs untuk setiap jenis sumber daya, lihat Kunci tindakan, sumber daya, dan kondisi untuk Amazon Application Recovery Controller (ARC) di Referensi Otorisasi Layanan.

Topik

Praktik terbaik kebijakan

Kebijakan berbasis identitas menentukan apakah seseorang dapat membuat, mengakses, atau menghapus sumber daya ARC di akun Anda. Tindakan ini membuat Akun AWS Anda dikenai biaya. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

- Mulailah dengan kebijakan AWS terkelola dan beralih ke izin hak istimewa paling sedikit — Untuk mulai memberikan izin kepada pengguna dan beban kerja Anda, gunakan kebijakan AWS terkelola yang memberikan izin untuk banyak kasus penggunaan umum. Mereka tersedia di Anda Akun AWS. Kami menyarankan Anda mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola AWS pelanggan yang khusus untuk kasus penggunaan Anda. Untuk informasi selengkapnya, lihat Kebijakan yang dikelola AWS atau Kebijakan yang dikelola AWS untuk fungsi tugas dalam Panduan Pengguna IAM.
- Menerapkan izin dengan hak akses paling rendah – Ketika Anda menetapkan izin dengan kebijakan IAM, hanya berikan izin yang diperlukan untuk melakukan tugas. Anda melakukannya dengan mendefinisikan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi

tertentu, yang juga dikenal sebagai izin dengan hak akses paling rendah. Untuk informasi selengkapnya tentang cara menggunakan IAM untuk mengajukan izin, lihat Kebijakan dan izin dalam IAM dalam Panduan Pengguna IAM.

- Gunakan kondisi dalam kebijakan IAM untuk membatasi akses lebih lanjut – Anda dapat menambahkan suatu kondisi ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. Sebagai contoh, Anda dapat menulis kondisi kebijakan untuk menentukan bahwa semua permintaan harus dikirim menggunakan SSL. Anda juga dapat menggunakan ketentuan untuk memberikan akses ke tindakan layanan jika digunakan melalui yang spesifik Layanan AWS, seperti AWS CloudFormation. Untuk informasi selengkapnya, lihat Elemen kebijakan JSON IAM: Kondisi dalam Panduan Pengguna IAM.
- Gunakan IAM Access Analyzer untuk memvalidasi kebijakan IAM Anda untuk memastikan izin yang aman dan fungsional – IAM Access Analyzer memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan tersebut mematuhi bahasa kebijakan IAM (JSON) dan praktik terbaik IAM. IAM Access Analyzer menyediakan lebih dari 100 pemeriksaan kebijakan dan rekomendasi yang dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang aman dan fungsional. Untuk informasi selengkapnya, lihat Validasi kebijakan dengan IAM Access Analyzer dalam Panduan Pengguna IAM.
- Memerlukan otentikasi multi-faktor (MFA) - Jika Anda memiliki skenario yang mengharuskan pengguna IAM atau pengguna root di Anda, Akun AWS aktifkan MFA untuk keamanan tambahan. Untuk meminta MFA ketika operasi API dipanggil, tambahkan kondisi MFA pada kebijakan Anda. Untuk informasi selengkapnya, lihat Amankan akses API dengan MFA dalam Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang praktik terbaik dalam IAM, lihat Praktik terbaik keamanan di IAM dalam Panduan Pengguna IAM.

Contoh: Akses konsol shift zonal

Untuk mengakses konsol Amazon Application Recovery Controller (ARC), Anda harus memiliki seperangkat izin minimum. Izin ini harus memungkinkan Anda untuk membuat daftar dan melihat detail tentang sumber daya ARC di Anda Akun AWS. Jika Anda membuat kebijakan berbasis identitas yang lebih ketat daripada izin minimum yang diperlukan, konsol tidak akan berfungsi sebagaimana mestinya untuk entitas (pengguna atau peran) dengan kebijakan tersebut.

Anda tidak perlu mengizinkan izin konsol minimum untuk pengguna yang melakukan panggilan hanya ke AWS CLI atau AWS API. Sebagai gantinya, izinkan akses hanya ke tindakan yang sesuai dengan operasi API yang coba mereka lakukan.

Untuk memberi pengguna akses penuh untuk menggunakan pergeseran zona di dalamnya AWS Management Console, lampirkan kebijakan seperti berikut ini ke pengguna:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                   "arc-zonal-shift:ListManagedResources",
                   "arc-zonal-shift:GetManagedResource",
                   "arc-zonal-shift:ListZonalShifts",
                   "arc-zonal-shift:StartZonalShift",
                   "arc-zonal-shift:UpdateZonalShift",
                   "arc-zonal-shift:CancelZonalShift"
             ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:DescribeAvailabilityZones",
            "Resource": "*"
        }
    ]
}
```

Contoh: Tindakan API pergeseran zona

API pergeseran zona untuk sementara memindahkan lalu lintas dari Availability Zone untuk memulihkan aplikasi.

Untuk memastikan bahwa pengguna dapat menggunakan tindakan API shift zona, lampirkan kebijakan yang sesuai dengan operasi API yang perlu dikerjakan pengguna, seperti berikut ini:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                   "arc-zonal-shift:ListManagedResources",
                   "arc-zonal-shift:GetManagedResource",
                   "arc-zonal-shift:ListZonalShifts",
```

```
                "arc-zonal-shift:StartZonalShift",
                "arc-zonal-shift:UpdateZonalShift",
                "arc-zonal-shift:CancelZonalShift"
            ],
            "Resource": "*"
        }
    ]
}
```

Autoshift zona di ARC

Dengan zonal autoshift, Anda mengizinkan AWS untuk mengalihkan lalu lintas sumber daya untuk aplikasi dari Availability Zone (AZ) selama acara, atas nama Anda, untuk membantu mengurangi waktu pemulihan. AWS memulai pergeseran otomatis ketika telemetri internal menunjukkan bahwa ada gangguan Availability Zone yang berpotensi berdampak pada pelanggan. Saat AWS memulai perpindahan otomatis, lalu lintas aplikasi ke sumber daya yang telah Anda konfigurasikan untuk pergeseran otomatis zona mulai bergeser dari Availability Zone.

Ketahuilah bahwa ARC tidak memeriksa kesehatan sumber daya individu. AWS memulai pergeseran otomatis ketika AWS telemetri mendeteksi bahwa ada gangguan Availability Zone yang berpotensi berdampak pada pelanggan. Dalam beberapa kasus, lalu lintas mungkin dialihkan untuk sumber daya yang tidak mengalami dampak.

Dengan zonal autoshift, Anda juga mengizinkan AWS untuk mengalihkan lalu lintas sumber daya untuk aplikasi dari Availability Zone, atas nama Anda, untuk latihan rutin berjalan. Latihan berjalan diperlukan untuk pergeseran otomatis zona. Pergeseran zona yang dimulai ARC untuk latihan berjalan membantu Anda memastikan bahwa mengalihkan lalu lintas dari Availability Zone selama autoshift aman untuk aplikasi Anda. Praktek berjalan secara teratur menguji bahwa aplikasi Anda dapat beroperasi secara normal tanpa satu Availability Zone dengan memulai pergeseran zona yang mengalihkan lalu lintas untuk sumber daya dari Availability Zone. Latihan berjalan berlangsung setiap minggu, dan memberikan hasil — seperti `SUCCEEDED` atau `FAILED` — untuk membantu Anda memahami apakah aplikasi beroperasi seperti yang diharapkan.

> **Important**
>
> Sebelum Anda mengonfigurasi praktik berjalan atau mengaktifkan pergeseran otomatis zona, kami sangat menyarankan Anda melakukan pra-skala kapasitas sumber daya aplikasi di semua Availability Zone di Wilayah tempat sumber daya aplikasi Anda digunakan. Anda tidak boleh mengandalkan penskalaan sesuai permintaan saat autoshift atau latihan dijalankan.

Zonal autoshift, termasuk latihan berjalan, bekerja secara independen, dan tidak menunggu tindakan penskalaan otomatis selesai. Mengandalkan penskalaan otomatis, alih-alih pra-penskalaan, dapat mengakibatkan aplikasi Anda pulih lebih lama.
Jika Anda menggunakan penskalaan otomatis untuk menangani siklus lalu lintas reguler, kami sangat menyarankan Anda mengonfigurasi kapasitas minimum penskalaan otomatis Anda untuk terus beroperasi secara normal dengan hilangnya Availability Zone.

Jika Anda berencana untuk mengaktifkan zonal autoshift atau mengkonfigurasi praktik berjalan, setelah Anda melakukan pra-skala kapasitas sumber daya aplikasi, uji apakah aplikasi Anda dapat beroperasi secara normal tanpa satu Availability Zone. Untuk menguji ini, mulai pergeseran zona untuk memindahkan lalu lintas untuk sumber daya dari Availability Zone.

Setelah Anda mengaktifkan zonal autoshift, kami sarankan Anda memverifikasi, dengan memulai dan mengevaluasi praktik on-demand run zonal shift, bahwa aplikasi Anda dapat terus beroperasi secara normal dengan lalu lintas bergeser dari Availability Zone. Kemudian, latihan rutin yang dilakukan ARC membantu Anda mengonfirmasi, secara berkelanjutan, bahwa Anda memiliki kapasitas yang cukup untuk pergeseran otomatis.

Untuk memastikan bahwa pengujian Anda dengan pergeseran zona efektif, penting untuk memvalidasi bahwa lalu lintas mengalir seperti yang diharapkan dari AZ tempat Anda menjauh. Misalnya, Application Load Balancer dan Network Load Balancer menyediakan metrik per AZ di Amazon CloudWatch yang dapat Anda gunakan untuk memantau ini. Bergantung pada berapa lama layanan dan klien menggunakan kembali koneksi, lalu lintas mungkin berlanjut ke AZ yang telah Anda geser lebih lama dari yang Anda harapkan. Untuk mempelajari lebih lanjut, lihat Batasi waktu klien tetap terhubung ke titik akhir Anda.

Anda dapat mengaktifkan pergeseran otomatis zona, untuk sumber daya yang didukung, di konsol ARC. Atau, di EC2 konsol Amazon, Anda memiliki opsi untuk mengaktifkan pergeseran otomatis zona untuk sumber daya penyeimbang beban tertentu. Untuk mempelajari selengkapnya tentang mengaktifkan pergeseran otomatis zona dengan Elastic Load Balancing, lihat Pergeseran zona di Panduan Pengguna Elastic Load Balancing.

Autoshift dan praktek run zonal shift bersifat sementara. Dengan pergeseran otomatis, saat Availability Zone yang terpengaruh pulih, AWS berhenti mengalihkan lalu lintas untuk sumber daya dari Availability Zone. Lalu lintas aplikasi untuk pelanggan kembali ke semua Availability Zone di Wilayah. Dengan latihan berjalan, lalu lintas digeser dari Availability Zone untuk satu sumber daya selama sekitar 30 menit, dan kemudian dipindahkan kembali ke semua Availability Zone di Wilayah.

Anda dapat mengonfigurasi EventBridge notifikasi Amazon untuk memberi tahu Anda tentang pergeseran otomatis dan praktik berjalan. Untuk informasi selengkapnya, lihat Menggunakan zonal autoshift dengan Amazon EventBridge.

Bagaimana zonal autoshift dan praktek berjalan bekerja

Kemampuan pergeseran otomatis zona di Amazon Application Recovery Controller (ARC) memungkinkan AWS untuk mengalihkan lalu lintas untuk sumber daya dari Availability Zone, atas nama Anda, ketika AWS menentukan bahwa ada gangguan yang berpotensi memengaruhi pelanggan di Availability Zone. Zonal autoshift dirancang untuk sumber daya yang telah diskalakan sebelumnya di semua Availability Zone dalam sebuah Wilayah AWS, sehingga aplikasi dapat beroperasi secara normal dengan hilangnya satu Availability Zone.

Dengan zonal autoshift, Anda diharuskan untuk mengonfigurasi praktik berjalan, di mana ARC secara teratur menggeser lalu lintas untuk sumber daya dari satu Availability Zone. Latihan jadwal ARC berjalan sekitar setiap minggu untuk setiap sumber daya yang memiliki konfigurasi praktik lari yang terkait dengannya. Latihan berjalan untuk setiap sumber daya dijadwalkan secara independen.

Untuk setiap latihan lari, ARC mencatat hasilnya. Jika latihan lari terganggu oleh kondisi pemblokiran, hasil latihan lari tidak ditandai sebagai berhasil. Untuk informasi lebih lanjut tentang hasil latihan lari, lihat Hasil untuk latihan lari.

Anda dapat mengonfigurasi EventBridge notifikasi Amazon untuk mengirimi Anda informasi tentang pergeseran otomatis dan praktik berjalan. Untuk informasi selengkapnya, lihat Menggunakan zonal autoshift dengan Amazon EventBridge.

Daftar Isi

Tentang zonal autoshift

Zonal autoshift adalah kemampuan di mana AWS mengalihkan lalu lintas sumber daya aplikasi dari Availability Zone, atas nama Anda. AWS memulai pergeseran otomatis ketika telemetri internal menunjukkan bahwa ada gangguan Availability Zone yang berpotensi berdampak pada pelanggan. Telemetri internal menggabungkan metrik dari beberapa sumber, termasuk AWS jaringan, dan layanan Amazon dan Elastic EC2 Load Balancing.

Anda harus mengaktifkan pergeseran otomatis zona secara manual untuk sumber daya yang didukung AWS .

Saat Anda menerapkan dan menjalankan AWS aplikasi pada penyeimbang beban dalam beberapa (biasanya tiga) AZs di Wilayah, dan Anda melakukan pra-skala untuk mendukung stabilitas statis, AWS dapat dengan cepat memulihkan aplikasi pelanggan di AZ dengan mengalihkan lalu lintas dengan perpindahan otomatis. Dengan mengalihkan lalu lintas sumber daya ke tempat lain AZs di Wilayah, AWS dapat mengurangi durasi dan tingkat keparahan dampak potensial yang disebabkan oleh pemadaman listrik, masalah perangkat keras atau perangkat lunak di AZ, atau gangguan lainnya.

Sumber daya yang didukung oleh ARC menyediakan integrasi yang menandai AZ yang ditentukan sebagai tidak sehat, yang mengakibatkan lalu lintas bergeser dari AZ yang terganggu.

Saat Anda mengaktifkan pergeseran otomatis zona untuk sumber daya, Anda juga harus mengonfigurasi praktik yang dijalankan untuk sumber daya tersebut. AWS melakukan latihan berjalan sekitar mingguan, selama 30 menit, untuk membantu Anda memastikan bahwa Anda memiliki kapasitas yang cukup untuk menjalankan aplikasi Anda tanpa salah satu Availability Zone di Wilayah.

Seperti halnya pergeseran zona, ada beberapa skenario spesifik di mana pergeseran otomatis zona tidak menggeser lalu lintas dari AZ. Misalnya, jika grup target penyeimbang beban di AZs tidak memiliki instance apa pun, atau jika semua instance tidak sehat, maka penyeimbang beban berada dalam status gagal terbuka dan Anda tidak dapat mengalihkan salah satunya. AZs

Untuk mempelajari selengkapnya tentang pergeseran otomatis zona, lihat. Autoshift zona di ARC

Saat AWS memulai dan menghentikan pergeseran otomatis

Saat Anda mengaktifkan pergeseran otomatis zona untuk sumber daya, Anda mengizinkan AWS untuk mengalihkan lalu lintas sumber daya untuk aplikasi dari Availability Zone selama acara, atas nama Anda, untuk membantu mengurangi waktu pemulihan.

Untuk mencapai hal ini, pergeseran otomatis zona menggunakan AWS telemetri untuk mendeteksi, sedini mungkin, bahwa ada gangguan Availability Zone yang berpotensi berdampak pada pelanggan. Saat AWS memulai perpindahan otomatis, lalu lintas ke sumber daya yang dikonfigurasi segera mulai bergeser dari Zona Ketersediaan yang terganggu yang berpotensi berdampak pada pelanggan.

Zonal autoshift adalah kemampuan yang dirancang untuk pelanggan yang telah menskalakan sumber daya aplikasi mereka untuk semua Availability Zone dalam file. Wilayah AWS Anda tidak boleh mengandalkan penskalaan sesuai permintaan saat autoshift atau latihan dijalankan.

AWS mengakhiri pergeseran otomatis ketika menentukan bahwa Availability Zone telah pulih.

Saat ARC menjadwalkan, memulai, dan mengakhiri latihan berjalan

ARC menjadwalkan latihan lari untuk sumber daya setiap minggu, selama sekitar 30 menit. ARC menjadwalkan, memulai, dan mengelola latihan berjalan untuk setiap sumber daya secara mandiri. ARC tidak menggabungkan latihan berjalan untuk sumber daya di akun yang sama. Anda juga dapat memulai latihan sesuai permintaan sendiri, untuk membantu memverifikasi bahwa penyiapan Anda aman untuk acara pergeseran otomatis zona.

Ketika latihan berjalan berlanjut untuk durasi yang diharapkan, tanpa gangguan, itu ditandai dengan hasil dari. `SUCCESSFUL` Ada beberapa kemungkinan hasil lainnya:`FAILED`,`INTERRUPTED`, dan`PENDING`. Nilai dan deskripsi hasil disertakan dalam bagian Hasil untuk latihan berjalan.

Ada beberapa skenario ketika ARC menyela latihan dan mengakhirinya. Misalnya, jika pergeseran otomatis dimulai selama latihan, ARC menyela lari latihan dan mengakhirinya. Sebagai contoh lain, katakan bahwa sumber daya memiliki respons yang merugikan terhadap praktik berjalan dan menyebabkan alarm yang telah Anda tentukan untuk memantau praktik berjalan ke `ALARM` keadaan. Dalam skenario ini, ARC juga menyela latihan lari dan mengakhirinya.

Selain itu, ada beberapa skenario ketika ARC tidak memulai latihan jadwal untuk sumber daya.

Menanggapi praktik yang terputus dan diblokir untuk sumber daya, ARC melakukan hal berikut:

- Jika latihan yang dijalankan untuk sumber daya terganggu saat sedang berlangsung, ARC menganggap latihan mingguan telah berakhir, dan menjadwalkan latihan baru untuk sumber daya untuk minggu depan. Hasil latihan mingguan ada `INTERRUPTED` dalam skenario ini, tidak`FAILED`. Hasil latihan berjalan diatur `FAILED` hanya ketika alarm hasil yang memantau latihan berjalan masuk ke `ALARM` keadaan selama latihan dijalankan.
- Jika ada kendala pemblokiran saat latihan yang dijalankan untuk sumber daya dijadwalkan akan dimulai, ARC tidak memulai latihan. ARC melanjutkan pemantauan rutin, untuk menentukan

apakah masih ada satu atau lebih kendala pemblokiran. Ketika tidak ada batasan pemblokiran, ARC memulai praktik yang dijalankan untuk sumber daya.

Berikut ini adalah contoh batasan pemblokiran yang menghentikan ARC untuk memulai, atau melanjutkan, praktik yang dijalankan untuk sumber daya:

- ARC tidak memulai atau melanjutkan latihan berjalan ketika ada AWS Fault Injection Service eksperimen yang sedang berlangsung. Jika suatu AWS FIS acara aktif ketika ARC telah menjadwalkan latihan lari untuk memulai, ARC tidak memulai latihan lari. Monitor ARC di seluruh latihan berjalan untuk memblokir kendala, termasuk acara. AWS FIS Jika suatu AWS FIS acara dimulai saat latihan berjalan aktif, ARC mengakhiri latihan dan tidak mencoba untuk memulai yang lain sampai latihan yang dijadwalkan secara teratur berikutnya dijalankan untuk sumber daya.
- Jika ada AWS peristiwa terkini di suatu Wilayah, ARC tidak memulai latihan untuk sumber daya, dan mengakhiri latihan aktif, di Wilayah.

Ketika latihan berjalan selesai tanpa terganggu, ARC menjadwalkan latihan berikutnya dalam seminggu, seperti biasa. Jika praktik berjalan tidak dimulai karena kendala pemblokiran, seperti AWS FIS eksperimen atau jendela waktu yang diblokir yang telah Anda tentukan, ARC terus mencoba memulai latihan hingga latihan dijalankan dapat dimulai.

Pemeriksaan kapasitas untuk latihan berjalan

Ketika latihan dijalankan, untuk sementara memindahkan lalu lintas dari Availability Zone, ARC menjalankan pemeriksaan untuk memverifikasi bahwa Anda memiliki kapasitas yang cukup di Availability Zone lainnya untuk memindahkan lalu lintas dengan aman dari AZ. Jika tidak ada kapasitas yang cukup tersedia, pergeseran lalu lintas untuk latihan tidak dimulai dan latihan berjalan berakhir.

Selain itu, ARC menjalankan pemeriksaan kapasitas untuk sumber daya penyeimbang beban ketika pergeseran otomatis zona selesai, sebelum ARC mengakhiri pergeseran lalu lintas yang dimulai oleh pergeseran otomatis. Jika pemeriksaan kapasitas gagal saat perpindahan otomatis berakhir, lalu lintas tidak digeser kembali ke Availability Zone tempat ia dipindahkan.

Pemeriksaan kapasitas seimbang hanya diselesaikan untuk load balancer dan grup Auto Scaling.

Untuk sumber daya penyeimbang beban, pemeriksaan kapasitas memvalidasi bahwa host sehat yang terkait dengan penyeimbang beban didistribusikan di seluruh Availability Zone. Secara khusus, pemeriksaan kapasitas memastikan bahwa jumlah host yang sehat di semua Availability Zone tempat

sumber daya terdaftar seimbang. Untuk pemeriksaan kapasitas, seimbang berarti bahwa kapasitas sehat untuk setiap Availability Zone setara dengan zona lain, dalam varians kecil.

Perhatikan bahwa pemeriksaan kapasitas tidak diterapkan pada penyeimbang beban dengan kelompok target tipe Lambda maupun Application Load Balancer, karena target tersebut tidak dikonfigurasi secara zonal.

Pemeriksaan kapasitas juga diselesaikan untuk grup Auto Scaling. Untuk grup Auto Scaling, pemeriksaan kapasitas memvalidasi bahwa total kapasitas zona sehat grup Auto Scaling—yaitu, jumlah total host sehat di semua Zona Ketersediaan—memenuhi kapasitas yang diinginkan untuk grup Auto Scaling tersebut.

Ketika pemeriksaan kapasitas gagal

Ketika pemeriksaan kapasitas menemukan bahwa kapasitas yang tersedia tidak seimbang untuk sumber daya, hasil untuk latihan dijalankan adalah`CAPACITY_CHECK_FAILED`. Untuk mempelajari lebih lanjut tentang mengapa pemeriksaan kapasitas gagal, lihat kolom komentar untuk`ZonalShiftSummary`. Untuk menemukan kolom komentar untuk latihan Anda jalankan zonal shift, lakukan hal berikut:

1. Dengan menggunakan AWS CLI, daftarkan pergeseran zona untuk sumber daya yang Anda tentukan dalam praktik yang dijalankan menggunakan operasi ListZonalShiftsAPI.

   FOr contoh, untuk mengembalikan pergeseran zona, Anda dapat menjalankan perintah yang mirip dengan berikut ini:

   ```
   aws arc-zonal-shift start-practice-run
       --resource-
   identifier="arn:aws:elasticloadbalancing:Region:111122223333:ExampleALB123456890"
   ```

2. Tinjau array `ZonalShiftSummary` objek yang dikembalikan untuk menemukan pergeseran zona untuk latihan yang gagal karena pemeriksaan kapasitas.
3. Untuk pergeseran zona yang berlaku, tinjau informasi di `Comment` lapangan.

Pemberitahuan untuk latihan berjalan dan autoshift

Anda dapat memilih untuk diberi tahu tentang praktik berjalan dan pergeseran otomatis untuk sumber daya Anda dengan menyiapkan notifikasi Amazon. EventBridge Anda dapat mengatur EventBridge notifikasi bahkan ketika Anda belum mengaktifkan pergeseran otomatis zona untuk sumber daya apa

pun, yang dikenal sebagai notifikasi pengamat pergeseran otomatis. Dengan notifikasi pengamat pergeseran otomatis, Anda akan diberi tahu tentang semua pergeseran otomatis bahwa ARC dimulai saat Availability Zone berpotensi terganggu. Perhatikan bahwa Anda harus mengonfigurasi opsi ini di setiap Wilayah AWS yang ingin Anda terima notifikasi.

Untuk melihat langkah-langkah untuk mengaktifkan notifikasi pengamat pergeseran otomatis, lihat. Mengaktifkan atau menonaktifkan notifikasi pengamat autoshift Untuk mempelajari lebih lanjut tentang opsi notifikasi dan cara mengonfigurasinya EventBridge, lihatMenggunakan zonal autoshift dengan Amazon EventBridge.

Prioritas untuk pergeseran zona

Tidak boleh ada lebih dari satu pergeseran zona yang diterapkan pada waktu tertentu. Artinya, hanya satu latihan yang menjalankan pergeseran zona, pergeseran zona yang diprakarsai pelanggan, pergeseran otomatis, atau eksperimen untuk sumber daya. AWS FIS Ketika pergeseran zona kedua dimulai, ARC mengikuti prioritas untuk menentukan jenis pergeseran zona mana yang berlaku untuk sumber daya.

Prinsip umum untuk diutamakan adalah bahwa pergeseran zona yang Anda mulai sebagai pelanggan lebih diutamakan daripada jenis shift lainnya. Namun, ketahuilah bahwa lari latihan yang AWS dimulai saat ini mencegah Anda memulai latihan sesuai permintaan.

Untuk mengilustrasikan prioritas dalam ARC, berikut ini adalah cara kerja prioritas misalnya skenario:

Jenis pergeseran zona diterapkan	Jenis pergeseran zona dimulai	Hasil
AWS FIS percobaan	Berlatih lari	Latihan lari akan gagal dimulai, karena AWS FIS eksperimen diutamakan.
AWS FIS percobaan	Pergeseran zona manual	AWS FIS Eksperimen akan dibatalkan, dan pergeseran zona manual akan diterapkan.
AWS FIS percobaan	Pergeseran otomatis zona	AWS FIS Eksperimen akan dibatalkan, dan pergeseran otomatis zona akan diterapka n.

Jenis pergeseran zona diterapkan	Jenis pergeseran zona dimulai	Hasil
AWS FIS percobaan	AWS FIS percobaan	AWS FIS Eksperimen yang dimulai akan gagal dimulai karena ada eksperimen yang berjalan yang memicu tindakan AWS FIS autoshift.
Berlatih lari	Pergeseran zona manual	Latihan lari akan dibatalkan dan hasilnya diatur ke`INTERRUPTED` , dan pergeseran zona akan diterapkan.
Berlatih lari	AWS FIS percobaan	Lari latihan akan dibatalkan dan hasilnya diatur ke`INTERRUPTED` , dan AWS FIS eksperimen akan diterapkan.
Berlatih lari	Pergeseran otomatis zona	Latihan lari akan dibatalkan dan hasilnya diatur ke`INTERRUPTED` , dan pergeseran otomatis zona akan diterapkan.
Pergeseran zona manual	Berlatih lari	Latihan lari akan gagal untuk memulai.
Pergeseran zona manual	AWS FIS percobaan	AWS FIS Eksperimen akan gagal dimulai, atau gagal jika sudah berlangsung.
Pergeseran zona manual	Pergeseran otomatis zona	Autoshift zonal akan `ACTIVE` tetapi tidak `APPLIED` pada sumber daya. Pergeseran zona manual diutamakan.

Jenis pergeseran zona diterapkan	Jenis pergeseran zona dimulai	Hasil
Pergeseran otomatis zona	AWS FIS percobaan	AWS FIS Eksperimen akan gagal untuk memulai, atau akan gagal jika sedang berlangsung.
Pergeseran otomatis zona	Pergeseran zona manual	Autoshift zonal akan `ACTIVE` tetapi tidak `APPLIED` pada sumber daya. Pergeseran zona manual diutamakan.
Pergeseran otomatis zona	Berlatih lari	Latihan lari akan gagal dimulai, karena pergeseran otomatis zona diutamakan.

Pergeseran lalu lintas yang saat ini berlaku untuk sumber daya memiliki status pergeseran zona yang diterapkan yang disetel ke`APPLIED`. Hanya satu shift yang diatur `APPLIED` kapan saja. Pergeseran lain yang sedang berlangsung diatur ke`NOT_APPLIED`, tetapi tetap dengan `ACTIVE` status.

Menghentikan autoshift aktif atau menjalankan latihan untuk sumber daya

Untuk menghentikan pergeseran otomatis yang sedang berlangsung untuk sumber daya, Anda harus membatalkan pergeseran zona.

Latihan rutin masih berlangsung untuk sumber daya, pada jadwal yang sama. Jika Anda ingin menghentikan latihan berjalan selain menonaktifkan autoshift, Anda harus menghapus konfigurasi praktik jalankan yang terkait dengan sumber daya.

Saat Anda menghapus konfigurasi run praktik, AWS berhenti menjalankan praktik yang mengalihkan lalu lintas untuk sumber daya dari Availability Zone setiap minggu. Selain itu, karena pergeseran otomatis zona memerlukan latihan berjalan, saat Anda menghapus konfigurasi run praktik menggunakan konsol ARC, tindakan ini juga menonaktifkan pergeseran otomatis zona untuk sumber daya. Namun, perhatikan bahwa jika Anda menggunakan Zonal Autoshift API untuk menghapus praktik yang dijalankan, Anda harus terlebih dahulu menonaktifkan pergeseran otomatis zonal untuk sumber daya.

Untuk informasi selengkapnya, lihat Membatalkan pergeseran otomatis zona dan Mengaktifkan dan bekerja dengan zonal autoshift.

Bagaimana lalu lintas digeser

Untuk pergeseran otomatis dan untuk praktik menjalankan pergeseran zona, lalu lintas digeser dari Availability Zone menggunakan mekanisme yang sama yang digunakan ARC untuk pergeseran zona yang diprakarsai pelanggan. Pemeriksaan kesehatan yang tidak sehat menghasilkan Amazon Route 53 menarik alamat IP yang sesuai untuk sumber daya dari DNS, sehingga lalu lintas dialihkan dari Availability Zone. Koneksi baru sekarang dirutekan ke Availability Zone lainnya Wilayah AWS sebagai gantinya.

Dengan autoshift, ketika Availability Zone pulih dan AWS memutuskan untuk mengakhiri autoshift, ARC membalikkan proses pemeriksaan kesehatan, meminta pemeriksaan kesehatan Route 53 dikembalikan. Kemudian, alamat IP zonal asli dipulihkan dan, jika pemeriksaan kesehatan terus sehat, Availability Zone disertakan dalam perutean aplikasi lagi.

Penting untuk diperhatikan bahwa pergeseran otomatis tidak didasarkan pada pemeriksaan kesehatan yang memantau kesehatan yang mendasari penyeimbang beban atau aplikasi. ARC menggunakan pemeriksaan kesehatan untuk memindahkan lalu lintas dari Availability Zones, dengan meminta pemeriksaan kesehatan diatur ke tidak sehat, dan kemudian mengembalikan pemeriksaan kesehatan ke normal kembali ketika berakhir autoshift atau pergeseran zona.

Alarm untuk latihan berjalan

Anda dapat menentukan dua CloudWatch alarm untuk latihan berjalan di zonal autoshift. Alarm pertama, alarm hasil, diperlukan. Anda harus mengonfigurasi alarm hasil untuk memantau kesehatan aplikasi Anda saat lalu lintas digeser dari Availability Zone selama setiap 30 menit latihan dijalankan.

Agar praktik berjalan efektif, tentukan sebagai alarm hasil CloudWatch alarm yang memantau metrik untuk sumber daya, atau aplikasi Anda, yang merespons dengan `ALARM` status saat aplikasi Anda terpengaruh secara negatif oleh hilangnya satu Availability Zone. Untuk informasi selengkapnya, lihat bagian Alarm yang Anda tentukan untuk latihan berjalan. Praktik terbaik saat Anda mengonfigurasi pergeseran otomatis zona

Alarm hasil juga memberikan informasi untuk hasil latihan lari yang dilaporkan ARC untuk setiap latihan yang dijalankan. Jika alarm memasuki `ALARM` keadaan, latihan berjalan berakhir dan hasil latihan lari dikembalikan sebagai`FAILED`. Jika latihan berjalan menyelesaikan periode uji terjadwal 30 menit dan alarm hasil tidak memasuki `ALARM` keadaan, hasilnya dikembalikan sebagai`SUCCEEDED`. Daftar semua nilai hasil, dengan deskripsi, disediakan di bagian Hasil untuk latihan berjalan.

Secara opsional, Anda dapat menentukan alarm kedua, alarm pemblokiran. Latihan blok alarm pemblokiran berjalan dari awal, atau melanjutkan, ketika dalam `ALARM` keadaan. Alarm ini memblokir praktik pergeseran lalu lintas agar tidak dimulai—dan menghentikan praktik apa pun yang sedang berlangsung—saat alarm dalam keadaan. `ALARM`

Misalnya, dalam arsitektur besar dengan beberapa layanan mikro, ketika satu layanan mikro mengalami masalah, Anda biasanya ingin menghentikan semua perubahan lain di lingkungan aplikasi, yang termasuk praktik pemblokiran yang berjalan.

Tanggal yang diblokir dan jendela yang diblokir (UTC)

Anda memiliki opsi untuk memblokir praktik berjalan untuk tanggal kalender tertentu, atau untuk jendela waktu tertentu, yaitu hari dan waktu, di UTC.

Misalnya, jika Anda memiliki pembaruan aplikasi yang dijadwalkan untuk diluncurkan pada 1 Mei 2024, dan Anda tidak ingin latihan berjalan untuk mengalihkan lalu lintas pada waktu itu, Anda dapat menetapkan tanggal yang diblokir untuk`2024-05-01`.

Atau, katakanlah Anda menjalankan ringkasan laporan bisnis tiga hari seminggu. Untuk skenario ini, Anda dapat mengatur hari dan waktu berulang berikut sebagai jendela yang diblokir, misalnya, di UTC:. `MON-20:30-21:30 WED-20:30-21:30 FRI-20:30-21:30`

Wilayah AWS ketersediaan untuk pergeseran otomatis zona

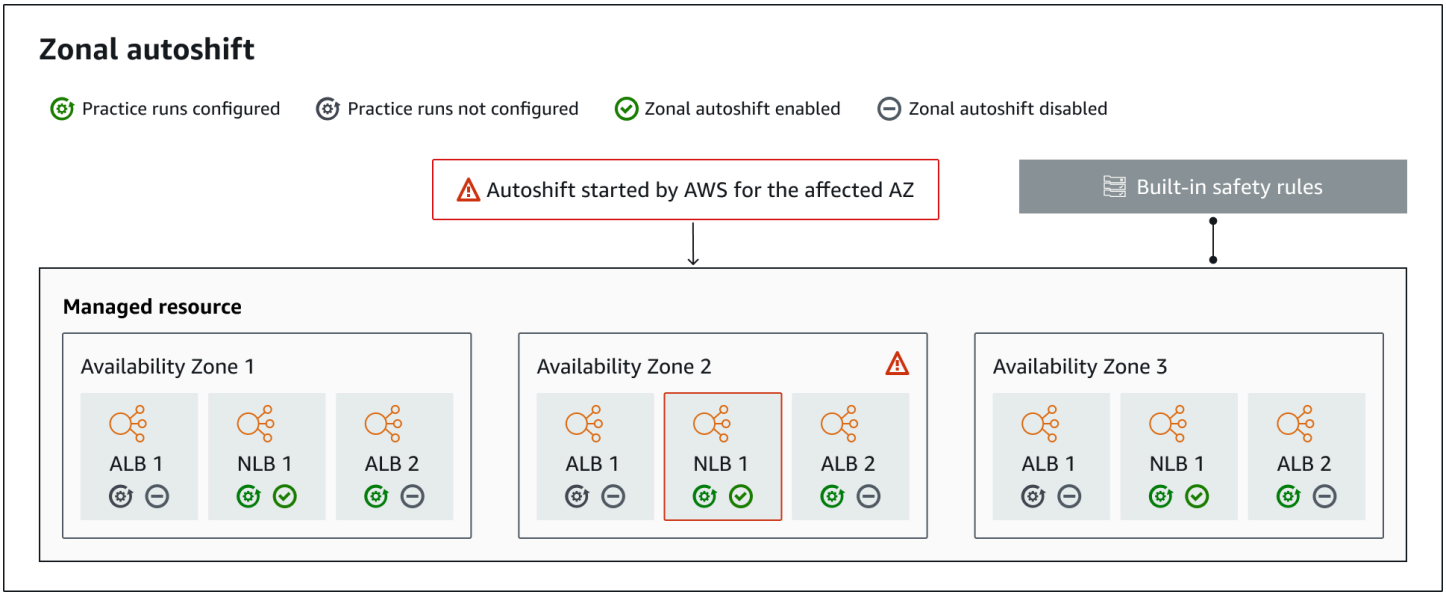
Pergeseran zona dan pergeseran otomatis zona saat ini tersedia di komersial Wilayah AWS, serta Wilayah Tiongkok, yaitu Wilayah China (Beijing) dan Wilayah China (Ningxia).

Sumber daya yang menggunakan Amazon Application Recovery Controller (ARC) dapat mencakup pertimbangan tambahan. Untuk informasi selengkapnya, lihat Sumber daya yang didukung.

Untuk daftar Wilayah dan informasi terperinci tentang dukungan Regional dan titik akhir layanan untuk ARC, lihat titik akhir dan kuota Amazon Application Recovery Controller (ARC) di Referensi Umum Amazon Web Services.

Komponen autoshift zona

Diagram berikut menggambarkan contoh perpindahan otomatis lalu lintas yang menjauh dari Availability Zone. AWS memulai pergeseran otomatis ketika telemetri internal menunjukkan bahwa ada gangguan Availability Zone yang berpotensi berdampak pada pelanggan.



Berikut ini adalah komponen kemampuan pergeseran otomatis zonal di ARC.

Pergeseran otomatis zona

Zonal autoshift menggeser lalu lintas untuk sumber daya, tanpa mengharuskan Anda untuk mengambil tindakan apa pun. Zonal autoshift adalah kemampuan di ARC di mana AWS memulai autoshift ketika telemetri internal menunjukkan bahwa ada gangguan Availability Zone yang berpotensi berdampak pada pelanggan. Ketahuilah bahwa, dalam beberapa kasus, sumber daya mungkin bergeser yang tidak mengalami dampak.

Latihan berjalan

Saat Anda mengaktifkan pergeseran otomatis zona untuk sumber daya, Anda juga harus mengonfigurasi praktik pergeseran otomatis zonal untuk sumber daya. AWS melakukan pergeseran zona untuk latihan berjalan sekitar setiap minggu, selama sekitar 30 menit. Anda juga dapat menjadwalkan latihan berjalan sesuai permintaan.

Latihan berjalan memastikan bahwa aplikasi Anda dapat berjalan normal dengan hilangnya satu Availability Zone. Dalam praktik berjalan, AWS menggeser lalu lintas untuk sumber daya dari satu Availability Zone dengan pergeseran zona, dan kemudian menggeser lalu lintas kembali ketika latihan berjalan berakhir.

Berlatih konfigurasi run

Konfigurasi praktik jalankan mendefinisikan tanggal dan jendela yang diblokir, jika ada, dan CloudWatch alarm yang Anda tentukan untuk AWS latihan dijalankan untuk sumber daya dalam pergeseran otomatis zona. Anda dapat mengedit konfigurasi praktik jalankan kapan saja, untuk

menambah atau mengubah tanggal atau jendela yang diblokir, atau memperbarui alarm untuk menjalankan latihan.

Untuk mengaktifkan pergeseran otomatis zona, Anda harus memiliki konfigurasi praktik jalankan untuk sumber daya. Anda juga dapat menghapus latihan lari. Untuk menghapus konfigurasi praktik jalankan untuk sumber daya, pergeseran otomatis zona harus dinonaktifkan.

Berlatih menjalankan alarm

Ketika Anda mengonfigurasi latihan berjalan, Anda menentukan CloudWatch alarm yang Anda buat CloudWatch, berdasarkan sumber daya dan persyaratan aplikasi Anda. Alarm yang Anda tentukan dapat memblokir proses latihan agar tidak dimulai, atau dapat menghentikan proses latihan yang sedang berlangsung, jika aplikasi Anda terpengaruh oleh praktik yang dijalankan.

Jika alarm yang Anda tentukan masuk ke `ALARM` status, ARC mengakhiri pergeseran zona untuk menjalankan latihan, sehingga lalu lintas untuk sumber daya tidak lagi bergeser dari Availability Zone.

Ada dua jenis alarm yang Anda tentukan untuk latihan berjalan: alarm hasil, untuk memantau kesehatan sumber daya dan aplikasi Anda selama latihan berjalan, dan alarm pemblokiran, yang dapat Anda konfigurasi untuk mencegah latihan berjalan dari awal, atau untuk menghentikan latihan yang sedang berjalan. Alarm hasil diperlukan; alarm pemblokiran adalah opsional.

Hasil latihan lari

ARC melaporkan hasil untuk setiap latihan lari. Berikut ini adalah hasil praktik lari yang mungkin:

- PENDING: Pergeseran zona untuk latihan berjalan aktif (sedang berlangsung). Belum ada hasil untuk kembali.
- BERHASIL: Alarm hasil tidak memasuki `ALARM` keadaan selama latihan berjalan, dan latihan berjalan menyelesaikan periode tes 30 menit penuh.
- INTERRUPTED: Latihan berjalan berakhir karena alasan yang bukan alarm hasil memasuki suatu `ALARM` keadaan. Latihan lari dapat terganggu karena berbagai alasan. Misalnya, latihan lari yang berakhir karena alarm pemblokiran yang ditentukan untuk menjalankan latihan memasuki `ALARM` status memiliki hasil`INTERRUPTED`. Untuk informasi lebih lanjut tentang alasan suatu `INTERRUPTED` hasil, lihat Hasil untuk latihan berjalan.
- GAGAL: Alarm hasil memasuki `ALARM` status selama latihan dijalankan.
- CAPACITY_CHECK_FAILED: Pemeriksaan kapasitas seimbang di seluruh Availability Zone untuk load balancing dan sumber daya grup Auto Scaling Anda gagal.

Aturan keselamatan bawaan

Aturan keselamatan yang dibangun ke dalam ARC mencegah lebih dari satu pergeseran lalu lintas untuk sumber daya berlaku pada suatu waktu. Artinya, hanya satu pergeseran zona yang diprakarsai pelanggan, praktik menjalankan pergeseran zona (diprakarsai oleh AWS atau oleh pelanggan), atau pergeseran otomatis untuk sumber daya dapat secara aktif mengalihkan lalu lintas dari Availability Zone. Misalnya, jika Anda memulai pergeseran zona untuk sumber daya ketika saat ini digeser dengan pergeseran otomatis, pergeseran zona Anda diutamakan. Untuk informasi lebih lanjut, lihat Prioritas untuk pergeseran zona.

Pengidentifikasi sumber daya

Pengenal sumber daya untuk mengaktifkan pergeseran otomatis zona, yang merupakan Nama Sumber Daya Amazon (ARN) untuk sumber daya. Anda hanya dapat mengaktifkan pergeseran otomatis zona untuk sumber daya di akun Anda yang ada di AWS layanan yang didukung oleh ARC.

Sumber daya terkelola

Application Load Balancers mendaftarkan sumber daya secara otomatis dengan ARC untuk pergeseran otomatis zona. Anda harus secara manual memilih sumber daya lain untuk pergeseran otomatis zona.

Nama sumber daya

Nama sumber daya yang dikelola di ARC.

Status terapan

Status yang diterapkan menunjukkan apakah pergeseran lalu lintas berlaku untuk sumber daya. Saat Anda mengonfigurasi pergeseran otomatis zona, sumber daya dapat memiliki lebih dari satu pergeseran lalu lintas aktif—yaitu, praktik menjalankan pergeseran zona, pergeseran zona yang dimulai pelanggan, atau pergeseran otomatis. Namun, hanya satu yang diterapkan, yaitu, berlaku untuk sumber daya pada suatu waktu. Pergeseran yang memiliki status `APPLIED` menentukan Availability Zone di mana lalu lintas aplikasi telah digeser untuk sumber daya, dan kapan pergeseran lalu lintas itu berakhir.

Jenis shift

Mendefinisikan jenis pergeseran zona. Pergeseran zona dapat memiliki salah satu dari jenis berikut:

- ZONAL_SHIFT

- ZONAL_AUTOSHIFT
- PRACTICE_RUN
- EKSPERIMEN FIS_

Bidang data dan kontrol untuk pergeseran otomatis zona

Saat Anda merencanakan kegagalan dan pemulihan bencana, pertimbangkan seberapa tangguh mekanisme failover Anda. Kami menyarankan Anda memastikan bahwa mekanisme yang Anda andalkan selama failover sangat tersedia, sehingga Anda dapat menggunakannya saat Anda membutuhkannya dalam skenario bencana. Biasanya, Anda harus menggunakan fungsi bidang data untuk mekanisme Anda kapan pun Anda bisa, untuk keandalan dan toleransi kesalahan terbesar. Dengan mengingat hal itu, penting untuk memahami bagaimana fungsionalitas layanan dibagi antara bidang kontrol dan pesawat data, dan kapan Anda dapat mengandalkan harapan keandalan ekstrim dengan bidang data layanan.

Secara umum, bidang kontrol memungkinkan Anda melakukan fungsi manajemen dasar, seperti membuat, memperbarui, dan menghapus sumber daya dalam layanan. Pesawat data menyediakan fungsionalitas inti layanan.

Untuk informasi selengkapnya tentang bidang data, pesawat kontrol, dan cara AWS membangun layanan untuk memenuhi target ketersediaan tinggi, lihat paper Stabilitas statis menggunakan Availability Zones di Amazon Builders' Library.

Harga untuk pergeseran otomatis zona di ARC

Untuk pergeseran otomatis zona, AWS mengalihkan lalu lintas dari Availability Zone atas nama Anda untuk sumber daya yang didukung saat AWS menentukan bahwa ada potensi masalah yang dapat mempengaruhi aplikasi pelanggan. Tidak ada biaya tambahan untuk mengaktifkan zonal autoshift.

Untuk informasi harga terperinci untuk ARC dan contoh harga, lihat Harga ARC.

Praktik terbaik saat Anda mengonfigurasi pergeseran otomatis zona

Perhatikan praktik dan pertimbangan terbaik berikut saat Anda mengaktifkan pergeseran otomatis zona di Amazon Application Recovery Controller (ARC).

Zonal autoshift mencakup dua jenis pergeseran lalu lintas: autoshift dan praktek menjalankan pergeseran zona.

- Dengan autoshift, AWS membantu mengurangi waktu Anda untuk pemulihan dengan mengalihkan lalu lintas sumber daya aplikasi dari Availability Zone selama acara, atas nama Anda.
- Dengan latihan lari, ARC memulai pergeseran zona atas nama Anda atau Anda memulai latihan shift zona. AWS Praktek menjalankan zonal shift menggeser lalu lintas dari Availability Zone untuk sumber daya, dan kembali lagi, dengan irama mingguan. Latihan berjalan membantu Anda memastikan bahwa Anda telah meningkatkan kapasitas yang cukup untuk Availability Zone di suatu Wilayah agar aplikasi Anda dapat mentolerir hilangnya satu Availability Zone.

Ada beberapa praktik dan pertimbangan terbaik yang perlu diingat dengan autoshift dan praktik lari. Tinjau topik berikut sebelum Anda mengaktifkan pergeseran otomatis zona atau mengkonfigurasi praktik berjalan untuk sumber daya.

Topik

Batasi waktu klien tetap terhubung ke titik akhir Anda

Ketika Amazon Application Recovery Controller (ARC) mengalihkan lalu lintas dari gangguan, misalnya, dengan menggunakan zonal shift atau zonal autoshift, mekanisme yang digunakan ARC untuk memindahkan lalu lintas aplikasi Anda adalah pembaruan DNS. Pembaruan DNS menyebabkan semua koneksi baru diarahkan menjauh dari lokasi yang rusak. Namun, klien dengan koneksi terbuka yang sudah ada sebelumnya mungkin terus membuat permintaan terhadap lokasi yang rusak sampai klien terhubung kembali. Untuk memastikan pemulihan yang cepat, kami sarankan Anda membatasi jumlah waktu klien tetap terhubung ke titik akhir Anda.

Jika Anda menggunakan Application Load Balancer, Anda dapat menggunakan `keepalive` opsi untuk mengonfigurasi berapa lama koneksi berlanjut. Kami menyarankan agar Anda menurunkan `keepalive` nilai agar sesuai dengan sasaran waktu pemulihan untuk aplikasi Anda, misalnya, 300 detik. Ketika Anda memilih `keepalive` waktu, pertimbangkan bahwa nilai ini adalah pertukaran antara menghubungkan kembali lebih sering secara umum, yang dapat memengaruhi latensi, dan lebih cepat memindahkan semua klien dari AZ atau Wilayah yang terganggu.

Untuk informasi selengkapnya tentang pengaturan `keepalive` opsi untuk Application Load Balancer, lihat durasi keepalive klien HTTP di Panduan Pengguna Application Load Balancer.

Tingkatkan kapasitas sumber daya Anda dan uji lalu lintas yang bergeser

Saat AWS mengalihkan lalu lintas dari satu Availability Zone untuk pergeseran zona atau pergeseran otomatis, penting bahwa Availability Zone yang tersisa dapat melayani peningkatan tarif permintaan untuk sumber daya Anda. Pola ini dikenal sebagai stabilitas statis. Untuk informasi selengkapnya, lihat whitepaper Stabilitas statis menggunakan Availability Zones di Library Amazon Builder.

Misalnya, jika aplikasi Anda memerlukan 30 instans untuk melayani kliennya, Anda harus menyediakan 15 instans di tiga Availability Zone, dengan total 45 instans. Dengan melakukan ini, ketika AWS mengalihkan lalu lintas dari satu Availability Zone—dengan autoshift atau selama latihan dijalankan— masih AWS dapat melayani klien aplikasi Anda dengan total 30 instans yang tersisa, di dua Availability Zone.

Kemampuan pergeseran otomatis zona di ARC membantu Anda memulihkan dengan cepat dari AWS peristiwa di Availability Zone ketika Anda memiliki aplikasi dengan sumber daya yang telah diskalakan sebelumnya agar berfungsi secara normal dengan hilangnya satu Availability Zone. Sebelum Anda mengaktifkan pergeseran otomatis zona untuk sumber daya, skala kapasitas sumber daya Anda di semua Availability Zone yang dikonfigurasi dalam file. Wilayah AWS Kemudian, mulai pergeseran zona untuk sumber daya, untuk menguji bahwa aplikasi Anda masih berjalan normal ketika lalu lintas digeser dari Availability Zone.

Setelah Anda menguji dengan pergeseran zona, aktifkan pergeseran otomatis zona dan konfigurasikan praktik berjalan untuk sumber daya aplikasi. Jalankan praktik sesuai permintaan Anda sendiri untuk membantu memastikan bahwa konfigurasi Anda diskalakan dengan benar. Latihan rutin berjalan dengan pergeseran otomatis zona membantu Anda memastikan—secara berkelanjutan—bahwa kapasitas Anda masih diskalakan dengan tepat. Dengan kapasitas yang cukup di seluruh Availability Zone, aplikasi Anda dapat terus melayani klien, tanpa gangguan, selama autoshift.

Untuk informasi selengkapnya tentang memulai pergeseran zona untuk sumber daya, lihatPergeseran zona di ARC.

Waspadai jenis dan batasan sumber daya

Zonal autoshift mendukung perpindahan lalu lintas dari Availability Zone untuk semua sumber daya yang didukung oleh pergeseran zona. Dalam beberapa skenario sumber daya tertentu,

pergeseran otomatis zona tidak menggeser lalu lintas dari Availability Zone untuk pergeseran otomatis.

Misalnya, jika grup target penyeimbang beban di Availability Zones tidak memiliki instance apa pun, atau jika semua instance tidak sehat, maka penyeimbang beban berada dalam status gagal terbuka. Jika AWS memulai pergeseran otomatis untuk penyeimbang beban dalam skenario ini, pergeseran otomatis tidak mengubah Zona Ketersediaan mana yang digunakan penyeimbang beban karena penyeimbang beban sudah dalam status terbuka gagal. Ini adalah perilaku yang diharapkan. Autoshift tidak dapat menyebabkan satu Availability Zone menjadi tidak sehat dan mengalihkan lalu lintas ke Availability Zone lainnya Wilayah AWS jika semua Availability Zone gagal dibuka (tidak sehat).

Untuk melihat detail tentang sumber daya yang didukung, termasuk semua persyaratan dan pengecualian yang harus diperhatikan, lihatSumber daya yang didukung.

Tentukan alarm untuk latihan berjalan

Anda mengonfigurasi setidaknya satu alarm (alarm hasil) untuk latihan berjalan dengan pergeseran otomatis zona. Secara opsional, Anda juga dapat mengkonfigurasi alarm kedua (alarm pemblokiran).

Saat Anda mempertimbangkan CloudWatch alarm yang Anda konfigurasikan untuk latihan berjalan untuk sumber daya Anda, ingatlah hal berikut:

- Untuk alarm hasil, yang diperlukan, sebaiknya Anda mengonfigurasi CloudWatch alarm agar masuk ke `ALARM` status ketika metrik untuk sumber daya, atau aplikasi Anda, menunjukkan bahwa mengalihkan lalu lintas dari Availability Zone berdampak buruk pada kinerja. Misalnya, Anda dapat menentukan ambang batas untuk tingkat permintaan untuk sumber daya Anda, dan kemudian mengonfigurasi alarm untuk masuk ke `ALARM` status ketika ambang batas terlampaui. Anda bertanggung jawab untuk mengonfigurasi alarm yang sesuai yang AWS menyebabkan mengakhiri latihan dan mengembalikan `FAILED` hasilnya.
- Kami menyarankan Anda mengikuti AWS Well Architected Framework, yang menyarankan Anda untuk menerapkan indikator kinerja utama (KPIs) sebagai CloudWatch alarm. Jika Anda melakukannya, Anda dapat menggunakan alarm ini untuk membuat alarm komposit untuk digunakan sebagai pemicu keamanan, untuk mencegah latihan berjalan dari awal jika mereka dapat menyebabkan aplikasi Anda kehilangan KPI. Ketika alarm tidak lagi dalam `ALARM` keadaan, ARC memulai latihan berjalan saat latihan dijalankan berikutnya dijadwalkan untuk sumber daya.

- Untuk alarm pemblokiran praktik, jika Anda memilih untuk mengonfigurasinya, Anda dapat memilih untuk melacak metrik tertentu yang Anda gunakan untuk menunjukkan bahwa Anda tidak ingin AWS latihan dijalankan.
- Untuk alarm lari latihan, Anda menentukan Nama Sumber Daya Amazon (ARN) untuk setiap alarm, yang harus Anda konfigurasikan terlebih dahulu di Amazon. CloudWatch CloudWatch Alarm yang Anda tentukan dapat berupa alarm komposit, untuk memungkinkan Anda menyertakan beberapa metrik dan memeriksa aplikasi dan sumber daya Anda yang dapat memicu alarm masuk ke status. `ALARM` Untuk informasi selengkapnya, lihat Menggabungkan alarm di Panduan CloudWatch Pengguna Amazon.
- Pastikan bahwa CloudWatch alarm yang Anda tentukan untuk latihan berjalan berada di Wilayah yang sama dengan sumber daya yang Anda konfigurasikan untuk latihan dijalankan.

Evaluasi hasil untuk latihan lari

ARC melaporkan hasil untuk setiap latihan lari. Setelah latihan berjalan, evaluasi hasilnya, dan tentukan apakah Anda perlu mengambil tindakan. Misalnya, Anda mungkin perlu menskalakan kapasitas atau menyesuaikan konfigurasi untuk alarm.

Berikut ini adalah hasil praktik lari yang mungkin:

- BERHASIL: Alarm hasil tidak memasuki `ALARM` keadaan selama latihan berjalan, dan latihan berjalan menyelesaikan periode tes 30 menit penuh.
- GAGAL: Alarm hasil memasuki `ALARM` status selama latihan dijalankan.
- INTERRUPTED: Latihan berjalan berakhir karena alasan yang bukan alarm hasil memasuki suatu `ALARM` keadaan. Latihan lari dapat terganggu karena berbagai alasan, termasuk yang berikut:
  - Praktek lari berakhir karena AWS mulai autoshift di Wilayah AWS atau ada kondisi alarm di Wilayah.
  - Practice run berakhir karena konfigurasi practice run telah dihapus untuk sumber daya.
  - Latihan lari berakhir karena pergeseran zona yang diprakarsai pelanggan dimulai untuk sumber daya di Availability Zone tempat praktik menjalankan pergeseran zona mengalihkan lalu lintas.
  - Praktek berjalan berakhir karena CloudWatch alarm yang ditentukan untuk konfigurasi praktek run tidak dapat lagi diakses.
  - Latihan lari berakhir karena alarm pemblokiran yang ditentukan untuk latihan berjalan memasuki `ALARM` status.
  - Latihan lari berakhir karena alasan yang tidak diketahui.

- Latihan lari berakhir karena pergeseran otomatis zona dengan prioritas dimulai. Lihat Prioritas untuk pergeseran zona.
- CAPACITY_CHECK_FAILED: Pemeriksaan kapasitas seimbang di seluruh Availability Zone untuk load balancing dan sumber daya grup Auto Scaling Anda gagal.
- PENDING: Latihan berjalan aktif (sedang berlangsung). Belum ada hasil untuk kembali.

Operasi API autoshift zona

Tabel berikut mencantumkan operasi ARC API yang dapat Anda gunakan dengan zonal autoshift. Untuk contoh penggunaan operasi API pergeseran otomatis zona dengan AWS CLI, lihat.

Untuk contoh cara menggunakan operasi API pergeseran otomatis zona umum dengan AWS Command Line Interface, lihat. Contoh menggunakan AWS CLI with zonal autoshift

Tindakan	Menggunakan konsol ARC	Menggunakan ARC API
Buat konfigurasi praktek run	Lihat Mengaktifkan atau menonaktifkan pergeseran otomatis zona	Lihat CreatePracticeRunConfiguration
Hapus konfigurasi praktek run	Lihat Mengkonfigurasi, mengedit, atau menghapus konfigurasi praktik yang dijalankan	Lihat DeletePracticeRunConfiguration
Daftar autoshift	Lihat Autoshift zona di ARC	Lihat ListAutoshifts
Daftar sumber daya untuk pergeseran otomatis zona	Lihat Sumber daya yang didukung	Lihat ListManagedResources
Dapatkan sumber daya untuk pergeseran otomatis zona	Lihat Sumber daya yang didukung	Lihat GetManagedResource
Mengedit konfigurasi praktek run	Lihat Mengkonfigurasi, mengedit, atau menghapus konfigurasi praktik yang dijalankan	Lihat UpdatePracticeRunConfiguration

Tindakan	Menggunakan konsol ARC	Menggunakan ARC API
Mengaktifkan atau menonaktifkan pergeseran otomatis zona	Lihat Mengaktifkan atau menonaktifkan pergeseran otomatis zona	Lihat UpdateZonalAutoshiftConfiguration
Mengaktifkan atau menonaktifkan pemberitahuan pengamat pergeseran otomatis	Lihat Mengaktifkan dan bekerja dengan zonal autoshift	Lihat UpdateAutoshiftObserverNotificationStatus
Mulai latihan lari	Lihat Memulai latihan lari zonal shift	Lihat StartPracticeRun
Batalkan latihan lari	Lihat Membatalkan latihan lari zonal shift	Lihat CancelPracticeRun

Contoh menggunakan AWS CLI with zonal autoshift

Bagian ini membahas contoh aplikasi sederhana bekerja dengan zonal autoshift, menggunakan AWS Command Line Interface untuk bekerja dengan kemampuan pergeseran otomatis zonal di Amazon Application Recovery Controller (ARC) menggunakan operasi API. Contoh-contoh tersebut dimaksudkan untuk membantu Anda mengembangkan pemahaman dasar tentang cara bekerja dengan pergeseran otomatis zonal menggunakan CLI.

Zonal autoshift adalah kemampuan dalam ARC. Dengan zonal autoshift, Anda berwenang AWS untuk mengalihkan lalu lintas sumber daya aplikasi yang didukung dari Availability Zone selama acara, atas nama Anda, untuk membantu mengurangi waktu Anda menuju pemulihan. Untuk informasi selengkapnya tentang sumber daya yang dapat Anda gunakan dengan zonal autoshift, lihat. Sumber daya yang didukung

Zonal autoshift mencakup praktik berjalan, yang juga mengalihkan lalu lintas dari Availability Zones, untuk membantu memverifikasi bahwa pergeseran otomatis aman untuk aplikasi Anda.

Untuk daftar tindakan API pergeseran otomatis zona dan tautan ke informasi selengkapnya, lihat. Operasi API autoshift zona Untuk informasi selengkapnya tentang penggunaan AWS CLI, lihat Referensi AWS CLI Perintah.

Daftar Isi

Buat konfigurasi praktek run

Sebelum Anda dapat mengaktifkan pergeseran otomatis zona untuk sumber daya, Anda harus membuat konfigurasi praktik jalankan untuk sumber daya, untuk memilih opsi untuk latihan yang diperlukan. Anda membuat konfigurasi praktek run untuk sumber daya dengan CLI dengan menggunakan perintah. `create-practice-run-configuration`

Perhatikan hal berikut saat Anda membuat konfigurasi run praktik untuk sumber daya:

- Satu-satunya jenis alarm yang didukung saat ini adalah`CLOUDWATCH`.
- Anda harus menggunakan alarm Wilayah AWS yang sama dengan sumber daya Anda digunakan.
- Menentukan alarm hasil diperlukan. Menentukan alarm pemblokiran adalah opsional.
- Menentukan tanggal yang diblokir atau jendela yang diblokir adalah opsional.

Anda membuat konfigurasi praktek run dengan CLI dengan menggunakan perintah. `create-practice-run-configuration`

Misalnya, untuk membuat konfigurasi run practice untuk sumber daya, gunakan perintah seperti berikut ini:

```
aws arc-zonal-shift create-practice-run-configuration \
      --resource-
identifier="arn:aws:elasticloadbalancing:Region:111122223333:ExampleALB123456890" \
      --outcome-alarms
 type=CLOUDWATCH,alarmIdentifier=arn:aws:cloudwatch:Region:111122223333:alarm:Region-
MyAppHealthAlarm \
      --blocking-alarms
 type=CLOUDWATCH,alarmIdentifier=arn:aws:cloudwatch:Region:111122223333:alarm:Region-
BlockWhenALARM \
```

```
      --blocked-dates 2023-12-01 --blocked-windows Mon:10:00-Mon:10:30
```

```
{
   "arn": "arn:aws:elasticloadbalancing:us-west-2:111122223333:ExampleALB123456890",
   "name": "zonal-shift-elb"
   "zonalAutoshiftStatus": "DISABLED",
   "practiceRunConfiguration": {
       "blockingAlarms": [
                   {
               "type": "CLOUDWATCH",
               "alarmIdentifier": "arn:aws:cloudwatch:us-west-2:111122223333:alarm:us-
west-2-BlockWhenALARM"
           }
       ]
       "outcomeAlarms": [
           {
               "type": "CLOUDWATCH",
               "alarmIdentifier": "arn:aws:cloudwatch:us-west-2:111122223333:alarm:us-
west-2-MyAppHealthAlarm"
           }
       ],
       "blockedWindows": [
           "Mon:10:00-Mon:10:30"
       ],
       "blockedDates": [
           "2023-12-01"
       ]
}
```

Mengaktifkan atau menonaktifkan pergeseran otomatis

Anda mengaktifkan atau menonaktifkan pergeseran otomatis untuk sumber daya dengan memperbarui status pergeseran otomatis zona dengan CLI. Untuk mengubah status autoshift zonal, gunakan perintah. `update-zonal-autoshift-configuration`

Misalnya, untuk mengaktifkan pergeseran otomatis untuk sumber daya, gunakan perintah seperti berikut:

```
aws arc-zonal-shift update-zonal-autoshift-configuration \
      --resource-
identifier="arn:aws:elasticloadbalancing:Region:111122223333:ExampleALB123456890" \
      --zonal-autoshift-status="ENABLED"
```

```
{
   "resourceIdentifier": "arn:aws:elasticloadbalancing:us-
west-2:111122223333:ExampleALB123456890",
   "zonalAutoshiftStatus": "ENABLED"
}
```

Mulai praktik sesuai permintaan

Anda dapat memulai praktik sesuai permintaan menjalankan zonal shift dengan CLI dengan menggunakan perintah. `start-practice-run`

Misalnya, untuk memulai latihan untuk sumber daya, gunakan perintah seperti berikut:

```
aws arc-zonal-shift start-practice-run
    --resource-
identifier="arn:aws:elasticloadbalancing:Region:111122223333:ExampleALB123456890" \
    "awayFrom": "usw2-az1",
```

```
{
    "awayFrom": "usw2-az1",
    "comment": "Practice run started. Shifting traffic away from Availability Zone
 usw2-az1.",
}
```

Batalkan proses latihan yang sedang berlangsung

Anda dapat membatalkan praktik yang sedang berjalan dengan CLI dengan menggunakan `cancel-practice-run` perintah.

Misalnya, untuk membatalkan praktik yang dijalankan untuk sumber daya, gunakan perintah seperti berikut ini:

```
aws arc-zonal-shift cancel-practice-run \
   --zonal-shift-id="="arn:aws:testservice::111122223333:ExampleALB123456890"
```

```
{
    "zonalShiftId": "2222222-3333-444-1111",
    "resourceIdentifier": "arn:aws:testservice::111122223333:ExampleALB123456890",
    "awayFrom": "usw2-az1",
    "expiryTime": 2024-11-15T10:35:42+00:00,
```

```
    "startTime": 2024-11-15T09:35:42+00:00,
    "status": "CANCELED",
    "comment": "Practice run canceled"
}
```

Membatalkan pergeseran otomatis yang sedang berlangsung

Anda dapat membatalkan pergeseran otomatis yang sedang berlangsung dengan CLI dengan membatalkan pergeseran otomatis zona untuk sumber daya. Untuk membatalkan pergeseran otomatis zona, gunakan file. `cancel-zonal-shift command`

```
aws arc-zonal-shift cancel-zonal-shift --zonal-shift-id
 9ac9ec1e-1df1-0755-3dc5-8cf573cd9c38
```

```
{
    "awayFrom": "usw2-az1",
    "comment": "Zonal autoshift started. Shifting traffic away from Availability Zone
 usw2-az1.",
    "expiryTime": "2024-12-17T22:29:38-08:00",
    "resourceIdentifier": "arn:aws:elasticloadbalancing:us-
east-1:111122223333:loadbalancer/app/Testing/5a19403ecd42dc05",
    "startTime": "2024-12-17T21:27:26-08:00",
    "status": "CANCELED",
    "zonalShiftId": "9ac9ec1e-1df1-0755-3dc5-8cf573cd9c38"
}
```

Mengedit konfigurasi praktek run

Anda dapat mengedit konfigurasi praktik jalankan untuk sumber daya dengan CLI untuk memperbarui opsi konfigurasi yang berbeda, seperti mengubah alarm untuk latihan berjalan atau memperbarui tanggal yang diblokir atau jendela yang diblokir, ketika ARC tidak akan memulai latihan berjalan. Untuk mengedit konfigurasi praktik jalankan, gunakan `update-practice-run-configuration` perintah.

Perhatikan hal berikut saat Anda mengedit konfigurasi praktik jalankan untuk sumber daya:

- Satu-satunya jenis alarm yang didukung saat ini adalah`CLOUDWATCH`.
- Anda harus menggunakan alarm Wilayah AWS yang sama dengan sumber daya Anda digunakan.
- Menentukan alarm hasil diperlukan. Menentukan alarm pemblokiran adalah opsional.
- Menentukan tanggal yang diblokir atau jendela yang diblokir adalah opsional.

- Tanggal yang diblokir atau jendela yang diblokir yang Anda tentukan menggantikan nilai yang ada.

Misalnya, untuk mengedit konfigurasi praktik jalankan sumber daya guna menentukan tanggal baru yang diblokir, gunakan perintah seperti berikut ini:

```
aws arc-zonal-shift update-practice-run-configuration \
      --resource-
identifier="arn:aws:elasticloadbalancing:Region:111122223333:ExampleALB123456890" \
      --blocked-dates 2024-03-01
```

```
{
   "arn": "arn:aws:elasticloadbalancing:us-west-2:111122223333:ExampleALB123456890",
   "name": "zonal-shift-elb"
   "zonalAutoshiftStatus": "DISABLED",
   "practiceRunConfiguration": {
       "blockingAlarms": [
                  {
              "type": "CLOUDWATCH",
              "alarmIdentifier": "arn:aws:cloudwatch:us-west-2:111122223333:alarm:us-
west-2-BlockWhenALARM"
           }
       ]
       "outcomeAlarms": [
           {
               "type": "CLOUDWATCH",
               "alarmIdentifier": "arn:aws:cloudwatch:us-west-2:111122223333:alarm:us-
west-2-MyAppHealthAlarm"
           }
       ],
       "blockedWindows": [
           "Mon:10:00-Mon:10:30"
       ],
       "blockedDates": [
           "2024-03-01"
       ]
}
```

Hapus konfigurasi praktek run

Anda dapat menghapus konfigurasi praktik jalankan untuk sumber daya, tetapi Anda harus terlebih dahulu menonaktifkan pergeseran otomatis zonal untuk sumber daya. Sumber daya diperlukan untuk

memiliki konfigurasi praktik jalankan agar pergeseran otomatis zona diaktifkan. Latihan rutin berjalan membantu Anda memastikan bahwa aplikasi Anda dapat berjalan normal tanpa satu Availability Zone.

Untuk menghapus konfigurasi practice run dengan menggunakan CLI, pertama, nonaktifkan zonal autoshift, jika diperlukan dengan menggunakan perintah. `update-zonal-autoshift` Kemudian, untuk menghapus konfigurasi praktik jalankan, gunakan `delete-practice-run-configuration` perintah.

Pertama, nonaktifkan zonal autoshift untuk sumber daya, menggunakan perintah seperti berikut:

```
aws arc-zonal-shift update-zonal-autoshift-configuration \
      --resource-
identifier="arn:aws:elasticloadbalancing:Region:111122223333:ExampleALB123456890" \
      --zonal-autoshift-status="DISABLED"
```

```
{
   "resourceIdentifier": "arn:aws:elasticloadbalancing:us-
west-2:111122223333:ExampleALB123456890",
   "zonalAutoshiftStatus": "DISABLED"
}
```

Kemudian, hapus konfigurasi practice run, menggunakan perintah seperti berikut:

```
aws arc-zonal-shift delete-practice-run-configuration \
      --resource-
identifier="arn:aws:elasticloadbalancing:Region:111122223333:ExampleALB123456890"
```

```
{
   "arn": "arn:aws:elasticloadbalancing:us-west-2:111122223333:ExampleALB123456890",
   "name": "TestResource",
   "zonalAutoshiftStatus": "DISABLED"
}
```

Mengaktifkan dan bekerja dengan zonal autoshift

Bagian ini menyediakan prosedur untuk bekerja dengan pergeseran otomatis zona di Amazon Application Recovery Controller (ARC). Setelah mengaktifkan pergeseran otomatis zona, Anda dapat membuat perubahan untuk mempraktikkan konfigurasi run, memulai praktik sesuai permintaan,

membatalkan shift yang sedang berlangsung, termasuk praktik berjalan, atau mengaktifkan notifikasi pengamat pergeseran otomatis.

Mengaktifkan atau menonaktifkan pergeseran otomatis zona

Langkah-langkah di sini menjelaskan cara mengaktifkan atau menonaktifkan pergeseran otomatis zona pada konsol Amazon Application Recovery Controller (ARC). Untuk bekerja dengan pergeseran otomatis zona secara terprogram, lihat Panduan Referensi API Zonal Shift dan Zonal Autoshift.

Saat pergeseran otomatis zona diaktifkan, Anda mengizinkan AWS untuk mengalihkan lalu lintas sumber daya aplikasi dari Availability Zone selama acara, atas nama Anda, untuk membantu mengurangi waktu Anda menuju pemulihan.

Untuk mengaktifkan atau menonaktifkan pergeseran otomatis zona

1. Buka konsol ARC dihttps://console.aws.amazon.com/route53recovery/home#/dashboard.
2. Di bawah Multi-AZ, pilih Zonal autoshift.
3. Di bawah konfigurasi autoshift zona sumber daya, pilih sumber daya.
4. Di menu Tindakan, pilih Aktifkan pergeseran otomatis zona, lalu ikuti langkah-langkah untuk menyelesaikan pembaruan.

Jika sumber daya tidak memiliki konfigurasi praktik jalankan, Aktifkan pergeseran otomatis zona tidak tersedia. Untuk mengkonfigurasi konfigurasi praktek run dan mengaktifkan zonal autoshift, pilih Configure zonal autoshift.

Daftar Isi

Mengkonfigurasi, mengedit, atau menghapus konfigurasi praktik yang dijalankan

Langkah-langkah di bagian ini menjelaskan cara mengedit atau menghapus konfigurasi praktik jalankan di konsol Amazon Application Recovery Controller (ARC). Untuk bekerja dengan pergeseran

otomatis zona secara terprogram, termasuk perubahan untuk mempraktikkan konfigurasi run, lihat Panduan Referensi API Zonal Shift dan Zonal Autoshift.

Jika Anda menghapus konfigurasi practice run di konsol, zonal autoshift dinonaktifkan. Sebelum Anda dapat menghapus konfigurasi practice run dengan operasi API, Anda harus menonaktifkan zonal autoshift. Anda dapat mengonfigurasi praktik yang dijalankan tanpa mengaktifkan pergeseran otomatis zona. Namun, agar pergeseran otomatis zona diaktifkan untuk sumber daya, Anda diharuskan menjalankan praktik yang dikonfigurasi untuk sumber daya.

Untuk mengkonfigurasi lari latihan

1. Buka konsol ARC dihttps://console.aws.amazon.com/route53recovery/home#/dashboard.
2. Di bawah Multi-AZ, pilih Zonal autoshift.
3. Pilih Konfigurasi pergeseran otomatis zona.
4. Pilih sumber daya untuk dikonfigurasi untuk pergeseran otomatis zona.
5. Pilih untuk menonaktifkan pergeseran otomatis zona jika Anda tidak AWS ingin memulai pergeseran otomatis untuk sumber daya saat ada acara. AWS Anda dapat melanjutkan dengan wizard untuk mengonfigurasi konfigurasi praktik jalankan tanpa mengaktifkan pergeseran otomatis, jika Anda mau.
6. Pilih opsi untuk latihan berjalan untuk sumber daya. Untuk alarm, Anda dapat melakukan hal berikut:

   - (Wajib) Tentukan alarm hasil untuk memantau praktik berjalan untuk sumber daya ini.
   - (Opsional) Tentukan alarm pemblokiran untuk latihan yang dijalankan untuk sumber daya ini.

   Untuk informasi selengkapnya, lihat bagian Alarm yang Anda tentukan untuk latihan berjalan. Praktik terbaik saat Anda mengonfigurasi pergeseran otomatis zona
7. Secara opsional, tentukan tanggal yang diblokir dan jendela yang diblokir. Pilih tanggal atau jendela (hari dan waktu) untuk memblokir ARC dari memulai praktik berjalan untuk sumber daya ini. Semua tanggal dan waktu dalam UTC.
8. Pilih kotak centang untuk mengonfirmasi bahwa Anda telah membaca catatan pengakuan.
9. Pilih Buat.

Untuk mengedit konfigurasi praktik jalankan

1. Buka konsol ARC dihttps://console.aws.amazon.com/route53recovery/home#/dashboard.

2. Di bawah Multi-AZ, pilih Zonal autoshift.
3. Di bawah konfigurasi autoshift zona sumber daya, pilih sumber daya.
4. Di menu Actions, pilih Edit practice run configuration.
5. Buat perubahan pada konfigurasi praktik jalankan, untuk melakukan satu atau beberapa hal berikut:

   - Untuk alarm, Anda dapat melakukan hal berikut:
     - Untuk alarm pemblokiran, Anda dapat menambahkan alarm, menghapus alarm, atau menentukan alarm pemblokiran yang berbeda.
     - Untuk alarm hasil yang memantau latihan berjalan, Anda dapat menentukan CloudWatch alarm yang berbeda untuk digunakan. Alarm hasil diperlukan, sehingga Anda tidak dapat menghapus alarm hasil.
   - Untuk tanggal yang diblokir dan jendela yang diblokir, Anda dapat menambahkan tanggal atau hari dan waktu baru, atau Anda dapat menghapus atau memperbarui tanggal atau hari dan waktu yang ada. Semua tanggal dan waktu dalam UTC.
6. Pilih Simpan.

Untuk menghapus konfigurasi praktik jalankan

1. Buka konsol ARC dihttps://console.aws.amazon.com/route53recovery/home#/dashboard.
2. Di bawah Multi-AZ, pilih Zonal autoshift.
3. Di bawah konfigurasi autoshift zona sumber daya, pilih sumber daya.
4. Di menu Tindakan, pilih Hapus konfigurasi praktik jalankan.
5. Pada dialog modal konfirmasi, ketik`Delete`, lalu pilih Hapus.

   Perhatikan bahwa menghapus konfigurasi praktik jalankan di konsol juga menonaktifkan pergeseran otomatis zona untuk sumber daya. Zonal autoshift membutuhkan latihan dijalankan untuk dikonfigurasi untuk sumber daya.

Membatalkan pergeseran otomatis zona

Untuk menghentikan pergeseran otomatis zona yang sedang berlangsung untuk sumber daya, Anda harus membatalkan pergeseran otomatis zona.

Untuk menghentikan pergeseran otomatis zona yang sedang berlangsung

1. Buka konsol ARC dihttps://console.aws.amazon.com/route53recovery/home#/dashboard.
2. Di bawah Multi-AZ, pilih Zonal shift.
3. Pilih pergeseran otomatis zona yang ingin Anda batalkan, lalu pilih Batalkan pergeseran zona.
4. Pada dialog modal konfirmasi, pilih Konfirmasi.

Memulai latihan lari zonal shift

Langkah-langkah di bagian ini menjelaskan cara memulai praktik sesuai permintaan, jalankan pergeseran zona di konsol ARC. Untuk bekerja dengan zonal shift dan zonal autoshift secara terprogram, lihat Zonal Shift dan Zonal Autoshift API Reference Guide.

Anda dapat memulai latihan menjalankan zonal shift setelah Anda mengonfigurasi zonal autoshift dan membuat konfigurasi run praktik.

Untuk memulai latihan, jalankan zonal shift

1. Buka konsol ARC dihttps://console.aws.amazon.com/route53recovery/home#/dashboard.
2. Di bawah Multi-AZ, pilih Zonal autoshift.
3. Di bawah sumber daya pergeseran otomatis Zonal, telusuri ke sumber daya individual yang memiliki pergeseran otomatis zona yang dikonfigurasi.
4. Pada halaman Ikhtisar sumber daya, pilih Mulai latihan lari.
5. Pilih Availability Zone, lalu masukkan komentar untuk menjalankan latihan Anda. Latihan berjalan akan mengalihkan lalu lintas dari Availability Zone yang Anda pilih.
6. Pilih Mulai.

Membatalkan latihan lari zonal shift

Langkah-langkah di bagian ini menjelaskan cara membatalkan pergeseran zona pada konsol ARC. Untuk bekerja dengan zonal shift dan zonal autoshift secara terprogram, lihat Zonal Shift dan Zonal Autoshift API Reference Guide.

Anda dapat membatalkan shift zona atau latihan lari yang Anda mulai sendiri. Anda juga dapat membatalkan pergeseran zona yang AWS dimulai untuk sumber daya untuk latihan yang dijalankan untuk pergeseran otomatis zona.

Untuk membatalkan latihan, jalankan zonal shift

1. Buka konsol ARC dihttps://console.aws.amazon.com/route53recovery/home#/dashboard.
2. Di bawah Multi-AZ, pilih Zonal shift.
3. Pilih shift zona lari latihan yang ingin Anda batalkan, lalu pilih Batalkan pergeseran zona atau Batalkan latihan lari.
4. Pada dialog modal konfirmasi, pilih Konfirmasi.

Mengaktifkan atau menonaktifkan notifikasi pengamat autoshift

Anda dapat mengonfigurasi pergeseran otomatis zona untuk memberi tahu Anda, melalui Amazon EventBridge, setiap kali AWS memulai perpindahan otomatis untuk mengalihkan lalu lintas dari Zona Ketersediaan yang berpotensi terganggu. Anda harus mengonfigurasi opsi ini di setiap Wilayah AWS yang ingin Anda terima notifikasi. Anda tidak perlu mengonfigurasi sumber daya spesifik apa pun dengan pergeseran otomatis zona untuk mengaktifkan pemberitahuan terpisah ini. Untuk informasi selengkapnya, lihat Menggunakan zonal autoshift dengan Amazon EventBridge.

Langkah-langkah di bagian ini menjelaskan cara mengaktifkan notifikasi pengamat autoshift dengan menggunakan konsol Amazon Application Recovery Controller (ARC). Untuk bekerja dengan pergeseran otomatis zona secara terprogram, lihat Panduan Referensi API Zonal Shift dan Zonal Autoshift.

Untuk mengaktifkan atau menonaktifkan notifikasi pengamat pergeseran otomatis

1. Buka konsol ARC dihttps://console.aws.amazon.com/route53recovery/home#/dashboard.
2. Di bawah Memulai, pilih Aktifkan notifikasi pengamat pergeseran otomatis.
3. Di kotak dialog konfirmasi, pilih Aktifkan pemberitahuan pengamat.

Menguji pergeseran otomatis zona dengan AWS FIS

Anda dapat menggunakannya AWS Fault Injection Service untuk menyiapkan dan menjalankan eksperimen yang membantu Anda mensimulasikan kondisi dunia nyata, seperti skenario AZ Availability: Power Interruption, yang akan mendemonstrasikan apa yang terjadi saat AWS memulai pergeseran otomatis zona pada sumber daya berkemampuan autoshift Anda selama kerusakan AZ yang berpotensi meluas.

Tindakan `aws:arc:start-zonal-autoshift` pemulihan awal memungkinkan Anda untuk menunjukkan bagaimana secara otomatis AWS akan menggeser lalu lintas, untuk sumber daya yang diaktifkan pergeseran otomatis zona, menjauh dari AZ yang berpotensi terganggu dan mengubah rute mereka ke sehat AZs dalam hal yang sama Wilayah AWS selama pelaksanaan skenario ketersediaan AZ.

Misalnya, Anda dapat menggunakan pustaka AWS FIS skenario untuk mensimulasikan gangguan AZ yang disebabkan oleh gangguan daya. Dalam percobaan ini, lima menit setelah gangguan daya AZ dimulai, tindakan pemulihan secara `aws:arc:start-zonal-autoshift` otomatis mengalihkan lalu lintas sumber daya dari AZ yang ditentukan. Lalu lintas digeser selama 25 menit tersisa dari gangguan daya, untuk menunjukkan bagaimana autoshift akan dipicu ketika ada potensi kerusakan AZ yang meluas. Ketika percobaan selesai, pergeseran lalu lintas berakhir dan lalu lintas mulai mengalir ke semua AZs lagi. Proses ini menunjukkan pemulihan lengkap dari peristiwa daya yang berdampak pada AZ.

Bagaimana eksperimen berbeda dari praktik pergeseran otomatis zonal

AWS FIS eksperimen berbeda dari praktik pergeseran otomatis zonal karena, selama latihan berjalan, ARC menggeser lalu lintas untuk sumber daya Anda dari satu AZ sebagai bagian dari proses normal untuk memastikan bahwa aplikasi Anda dapat mentolerir hilangnya AZ. Namun, selama AWS FIS percobaan, AWS FIS tunjukkan bagaimana gangguan AZ dan pergeseran otomatis akan dipicu untuk sumber daya berkemampuan autoshift Anda atas nama Anda, dan kemudian membatalkan pergeseran otomatis saat gangguan telah diselesaikan.

Anda tidak dapat memperbarui pergeseran zona yang AWS diprakarsai FIS saat sedang berjalan. Selain itu, jika Anda membatalkan pergeseran zona di luar AWS FIS, AWS FIS percobaan berakhir.

AWS FIS mekanisme keamanan berbasis kedaluwarsa

AWS FIS mengelola pergeseran zona menggunakan operasi StartZonalShift, UpdateZonalShift, dan CancelZonalShiftAPI, dengan `expiresIn` bidang untuk permintaan ini disetel ke 1 menit sebagai mekanisme keamanan. Hal ini memungkinkan AWS FIS untuk dengan cepat memutar kembali pergeseran zona jika ada kejadian tak terduga, seperti pemadaman jaringan atau masalah sistem. Di konsol ARC, bidang waktu kedaluwarsa akan menampilkan AWS FIS-managed, dan kadaluarsa yang diharapkan sebenarnya ditentukan oleh durasi yang ditentukan dalam aksi pergeseran zona. Untuk informasi selengkapnya tentang latihan lari, lihat Cara kerja pergeseran otomatis zona dan praktik berjalan

Tidak boleh ada lebih dari satu pergeseran zona yang diterapkan pada waktu tertentu. Artinya, hanya satu latihan yang menjalankan pergeseran zona, pergeseran zona yang diprakarsai pelanggan, pergeseran otomatis, atau eksperimen untuk sumber daya. AWS FIS Ketika pergeseran zona kedua dimulai, ARC mengikuti prioritas untuk menentukan jenis pergeseran zona mana yang berlaku untuk sumber daya. Untuk informasi lebih lanjut tentang prioritas untuk pergeseran zona, lihat. Prioritas untuk pergeseran zona

Untuk informasi selengkapnya tentang tindakan AWS FIS pemulihan, lihat tindakan AWS FIS pemulihan di Panduan AWS Fault Injection Service Pengguna.

Pencatatan dan pemantauan untuk pergeseran otomatis zona di Amazon Application Recovery Controller (ARC)

Anda dapat menggunakan AWS CloudTrail dan Amazon EventBridge untuk memantau pergeseran otomatis zona di Amazon Application Recovery Controller (ARC), untuk menganalisis pola dan membantu memecahkan masalah.

Topik

- Mencatat panggilan API autoshift zonal menggunakan AWS CloudTrail
- Menggunakan zonal autoshift dengan Amazon EventBridge

Mencatat panggilan API autoshift zonal menggunakan AWS CloudTrail

Zonal autoshift untuk ARC terintegrasi dengan AWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan di ARC. CloudTrail menangkap semua panggilan API untuk pergeseran zona sebagai peristiwa. Panggilan yang diambil termasuk panggilan dari konsol ARC dan panggilan kode ke operasi ARC API untuk pergeseran zona.

Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail acara secara terus menerus ke bucket Amazon S3, termasuk peristiwa untuk pergeseran zona. Jika Anda tidak mengonfigurasi jejak, Anda masih dapat melihat peristiwa terbaru di CloudTrail konsol dalam Riwayat acara.

Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat ke ARC untuk pergeseran zona, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan.

Untuk mempelajari selengkapnya CloudTrail, lihat Panduan AWS CloudTrail Pengguna.

Informasi pergeseran otomatis zona di CloudTrail

CloudTrail diaktifkan pada Akun AWS saat Anda membuat akun. Ketika aktivitas terjadi di ARC untuk pergeseran otomatis zona, aktivitas tersebut direkam dalam suatu CloudTrail peristiwa bersama dengan peristiwa AWS layanan lainnya dalam riwayat Peristiwa. Anda dapat melihat, mencari, dan mengunduh acara terbaru di situs Anda Akun AWS. Untuk informasi selengkapnya, lihat Bekerja dengan riwayat CloudTrail Acara.

Untuk catatan peristiwa yang sedang berlangsung di Anda Akun AWS, termasuk acara untuk pergeseran otomatis zona di ARC, buat jejak. Jejak memungkinkan CloudTrail untuk mengirimkan file log ke bucket Amazon S3. Secara default, saat Anda membuat jejak di konsol, jejak tersebut berlaku untuk semua Wilayah AWS. Jejak mencatat peristiwa dari semua Wilayah di AWS partisi dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat mengonfigurasi AWS layanan lain, untuk menganalisis lebih lanjut dan menindaklanjuti data peristiwa yang dikumpulkan dalam CloudTrail log. Untuk informasi selengkapnya, lihat berikut:

- Gambaran umum untuk membuat jejak
- CloudTrail layanan dan integrasi yang didukung
- Mengonfigurasi notifikasi Amazon SNS untuk CloudTrail
- Menerima file CloudTrail log dari beberapa wilayah dan Menerima file CloudTrail log dari beberapa akun

Semua tindakan ARC dicatat oleh CloudTrail dan didokumentasikan dalam Panduan Referensi API Kontrol Perutean untuk Pengontrol Pemulihan Aplikasi Amazon. Misalnya, panggilan ke `StartZonalShift` dan `ListManagedResources` tindakan menghasilkan entri dalam file CloudTrail log.

Setiap entri peristiwa atau log berisi informasi tentang siapa yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan berikut ini:

- Apakah permintaan itu dibuat dengan kredenal pengguna root atau AWS Identity and Access Management (IAM).
- Apakah permintaan tersebut dibuat dengan kredensial keamanan sementara untuk satu peran atau pengguna gabungan.
- Apakah permintaan itu dibuat oleh AWS layanan lain.

Untuk informasi selengkapnya, lihat Elemen userIdentity CloudTrail .

Melihat peristiwa ARC dalam sejarah acara

CloudTrail memungkinkan Anda melihat peristiwa terbaru dalam riwayat Acara. Untuk informasi selengkapnya, lihat Bekerja dengan riwayat CloudTrail Acara di Panduan AWS CloudTrail Pengguna.

Memahami entri file log autoshift zona

Trail adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai file log ke bucket Amazon S3 yang Anda tentukan. CloudTrail file log berisi satu atau lebih entri log. Peristiwa mewakili permintaan tunggal dari sumber manapun dan mencakup informasi tentang tindakan yang diminta, tanggal dan waktu tindakan, parameter permintaan, dan sebagainya. CloudTrail file log bukanlah jejak tumpukan yang diurutkan dari panggilan API publik, jadi file tersebut tidak muncul dalam urutan tertentu.

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan `ListManagedResources` tindakan untuk pergeseran otomatis zona.


```
{
      "eventVersion": "1.08",
      "userIdentity": {
        "type": "AssumedRole",
        "principalId": "A1B2C3D4E5F6G7EXAMPLE",
        "arn": "arn:aws:iam::111122223333:role/admin",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
          "sessionIssuer": {
            "type": "Role",
            "principalId": "AROA33L3W36EXAMPLE",
            "arn": "arn:aws:iam::111122223333:role/admin",
            "accountId": "111122223333",
            "userName": "EXAMPLENAME"
          },
          "webIdFederationData": {},
          "attributes": {
            "creationDate": "2022-11-14T16:01:51Z",
            "mfaAuthenticated": "false"
          }
        }
      },
      "eventTime": "2022-11-14T16:14:41Z",
      "eventSource": "arc-zonal-shift.amazonaws.com",
      "eventName": "ListManagedResources",
```

```
      "awsRegion": "us-west-2",
      "sourceIPAddress": "192.0.2.50",
      "userAgent": "Boto3/1.17.101 Python/3.8.10 Linux/4.14.231-180.360.amzn2.x86_64
 exec-env/AWS_Lambda_python3.8 Botocore/1.20.102",
      "requestParameters": null,
      "responseElements": null,
      "requestID": "VGXG4ZUE7UZTVCMTJGIAF_EXAMPLE",
      "eventID": "4b5c42df-1174-46c8-be99-d67_EXAMPLE",
      "readOnly": true,
      "eventType": "AwsApiCall",
      "managementEvent": true,
      "recipientAccountId": "111122223333"
      "eventCategory": "Management"
      }
    }
```

Menggunakan zonal autoshift dengan Amazon EventBridge

Menggunakan Amazon EventBridge, Anda dapat menyiapkan aturan berbasis peristiwa yang memantau sumber daya perpindahan otomatis zona Anda dan memulai tindakan target yang menggunakan layanan lain. AWS Misalnya, Anda dapat menetapkan aturan untuk mengirimkan notifikasi email dengan memberi sinyal topik Amazon SNS saat latihan dijalankan untuk pergeseran otomatis zona.

Anda dapat membuat aturan di Amazon EventBridge untuk bertindak pada pergeseran otomatis zona. Peristiwa untuk pergeseran otomatis zona menentukan informasi status tentang praktik berjalan atau pergeseran otomatis, misalnya, saat latihan dijalankan. Anda dapat mengonfigurasi pergeseran otomatis zona untuk memberi tahu Anda tentang peristiwa pergeseran otomatis zona untuk sumber daya yang Anda aktifkan untuk layanan.

Anda juga dapat memilih, selain atau bukan notifikasi lain, untuk mengaktifkan notifikasi pengamat pergeseran otomatis, yang menyediakan acara notifikasi setiap kali AWS memulai perpindahan otomatis untuk Availability Zone yang berpotensi mengalami gangguan. Pemberitahuan pengamat pergeseran otomatis terpisah dari notifikasi yang Anda terima saat lalu lintas untuk sumber daya yang telah Anda aktifkan untuk pergeseran otomatis zona digeser dari Availability Zone. Anda tidak perlu mengonfigurasi sumber daya apa pun dengan pergeseran otomatis zona untuk mengaktifkan notifikasi pengamat pergeseran otomatis. Untuk informasi selengkapnya, lihat Mengaktifkan dan bekerja dengan zonal autoshift.

Untuk menangkap peristiwa pergeseran otomatis zona tertentu yang Anda minati, tentukan pola spesifik peristiwa yang EventBridge dapat digunakan untuk mendeteksi peristiwa. Pola acara memiliki

struktur yang sama dengan peristiwa yang cocok. Pola mengutip bidang yang ingin Anda cocokkan dan memberikan nilai yang Anda cari.

Peristiwa dipancarkan atas dasar upaya terbaik. Mereka dikirim dari ARC ke EventBridge dalam waktu dekat, dalam keadaan operasional normal. Namun, situasi dapat muncul yang mungkin menunda atau mencegah pengiriman suatu peristiwa.

Untuk informasi tentang cara kerja EventBridge aturan dengan pola peristiwa, lihat Peristiwa dan Pola Peristiwa di EventBridge.

Pantau sumber daya pergeseran otomatis zona dengan EventBridge

Dengan EventBridge, Anda dapat membuat aturan yang menentukan tindakan yang harus diambil saat ARC memancarkan peristiwa untuk sumber dayanya. Misalnya, Anda dapat membuat aturan yang mengirimkan pesan email saat latihan dijalankan untuk pergeseran otomatis zona.

Untuk mengetik atau menyalin dan menempelkan pola acara ke EventBridge konsol, pilih opsi untuk menggunakan Masukkan opsi saya sendiri di konsol. Untuk membantu Anda menentukan pola peristiwa yang mungkin berguna bagi Anda, topik ini mencakup contoh pola pencocokan peristiwa pergeseran otomatis zona dan peristiwa pergeseran otomatis zona yang dapat Anda gunakan.

Untuk membuat aturan untuk peristiwa sumber daya

1. Buka EventBridge konsol Amazon di https://console.aws.amazon.com/events/.
2. Pilih tempat Wilayah AWS Anda ingin membuat aturan, yaitu Wilayah yang Anda minati untuk menonton acara.
3. Pilih Buat aturan.
4. Masukkan Nama untuk aturan tersebut, dan, secara opsional, deskripsi.
5. Untuk bus Acara, biarkan nilai default, default.
6. Pilih Berikutnya.
7. Untuk langkah pola acara Build, untuk sumber Event, tinggalkan nilai default, AWS peristiwa.
8. Di bawah Contoh acara, pilih Masukkan milik saya.
9. Untuk contoh peristiwa, ketik atau salin dan tempel pola acara.

Contoh pola peristiwa autoshift zona

Pola acara memiliki struktur yang sama dengan peristiwa yang cocok. Pola mengutip bidang yang ingin Anda cocokkan dan memberikan nilai yang Anda cari.

Anda dapat menyalin dan menempelkan pola peristiwa dari bagian ini ke dalam EventBridge untuk membuat aturan yang dapat Anda gunakan untuk memantau tindakan dan sumber daya pergeseran otomatis zona.

Saat Anda membuat pola acara untuk acara pergeseran otomatis zona, Anda dapat menentukan salah satu dari berikut ini untuk: `detail-type`

- `Autoshift In Progress`
- `Autoshift Completed`
- `Practice Run Started`
- `Practice Run Succeeded`
- `Practice Run Interrupted`
- `Practice Run Failed`
- `FIS Experiment Autoshift In Progress`
- `FIS Experiment Autoshift Completed`
- `FIS Experiment Autoshift Canceled`

Ketika latihan lari terputus, untuk informasi lebih lanjut tentang apa yang menyebabkan gangguan, lihat lapangan. `additionalFailureInfo`

Anda dapat memilih untuk memantau semua AWS pergeseran otomatis dengan mengaktifkan notifikasi pengamat pergeseran otomatis. Setelah Anda mengaktifkan notifikasi pengamat pergeseran otomatis, untuk menerima notifikasi, pilih untuk diberi tahu untuk jenis detail pergeseran otomatis zona. `Autoshift In Progress` Untuk melihat langkah-langkah untuk mengaktifkan notifikasi pengamat pergeseran otomatis, lihat. Mengaktifkan dan bekerja dengan zonal autoshift

Sebagai contoh, lihat bagian Contoh peristiwa pergeseran otomatis zona.

- Pilih semua peristiwa dari pergeseran otomatis zona tempat perpindahan otomatis telah dimulai.

  Perhatikan hal berikut:

  - Jika Anda mengaktifkan notifikasi pengamat pergeseran otomatis, ARC mengembalikan semua peristiwa pergeseran otomatis.
  - Jika Anda tidak mengaktifkan notifikasi pengamat pergeseran otomatis, ARC mengembalikan peristiwa pergeseran otomatis hanya jika sumber daya yang telah Anda konfigurasikan untuk pergeseran otomatis zona disertakan dalam pergeseran otomatis.

```
{
    "source": [
        "aws.arc-zonal-shift"
    ],
    "detail-type": [
        "Autoshift In Progress"
    ]
}
```

- Pilih semua acara dari pergeseran otomatis zona tempat latihan telah dimulai.

```
{
    "source": [
        "aws.arc-zonal-shift"
    ],
    "detail-type": [
        "Practice Run Started"
    ]
}
```

- Pilih semua peristiwa dari pergeseran otomatis zona di mana latihan dijalankan gagal.

```
{
    "source": [
        "aws.arc-zonal-shift"
    ],
    "detail-type": [
        "Practice Run Failed"
    ]
}
```

Contoh peristiwa pergeseran otomatis zona

Bagian ini mencakup contoh peristiwa untuk tindakan pergeseran otomatis zona.

Berikut ini adalah contoh peristiwa untuk `Autoshift In Progress` tindakan, ketika 1) pemberitahuan pengamat pergeseran otomatis diaktifkan dan 2) Anda belum mengonfigurasi sumber daya dengan pergeseran otomatis zona yang disertakan dalam pergeseran otomatis:

```
{
    "version": "0",
```

```
    "id": "05d4d2d5-9c76-bfea-72d2-d4614802adb4",
    "detail-type": "Autoshift In Progress",
    "source": "aws.arc-zonal-shift",
    "account": "111122223333",
    "time": "2023-11-16T23:38:14Z",
    "region": "us-east-1",
    "resources": [],
    "detail": {
        "version": "0.0.1",
        "data": "",
        "metadata": {
            "awayFrom": "use1-az2",
            "notes":"AWS has started an autoshift for an impaired Availability Zone.
 This notification
            is separate from autoshift notifications for resources, if any, that you
 have configured for
            zonal autoshift. For details, see the Developer Guide."
        }
    }
}
```

Berikut ini adalah contoh peristiwa untuk `Autoshift In Progress` tindakan, ketika 1) pemberitahuan pengamat pergeseran otomatis dinonaktifkan dan 2) Anda telah mengonfigurasi sumber daya dengan pergeseran otomatis zona yang disertakan dalam pergeseran otomatis:

```
{
    "version": "0",
    "id": "05d4d2d5-9c76-bfea-72d2-d4614802adb4",
    "detail-type": "Autoshift In Progress",
    "source": "aws.arc-zonal-shift",
    "account": "111122223333",
    "time": "2023-11-16T23:38:14Z",
    "region": "us-east-1",
    "resources": [
        "TEST-EXAMPLE-2023-11-16-23-28-11-5"
    ],
    "detail": {
        "version": "0.0.1",
        "data": "",
        "metadata": {
            "awayFrom": "use1-az2",
            "notes":""
        }
```

```
    }
}
```

Berikut ini adalah contoh peristiwa untuk `Practice Run Interrupted` tindakan:

```
{
    "version": "0",
    "id": "05d4d2d5-9c76-bfea-72d2-d4614802adb4",
    "detail-type": "Practice Run Interrupted",
    "source": "aws.arc-zonal-shift",
    "account": "111122223333",
    "time": "2023-11-16T23:38:14Z",
    "region": "us-east-1",
    "resources": [
        "TEST-EXAMPLE-2023-11-16-23-28-11-5"
    ],
    "detail": {
        "version": "0.0.1",
        "data": {
            "additionalFailureInfo": "Practice run interrupted. The blocking alarm
 entered ALARM state."
        },
        "metadata": {
            "awayFrom": "use1-az2"
        }
    }
}
```

Berikut ini adalah contoh peristiwa untuk `FIS Experiment Autoshift In Progress` tindakan:

```
{
    "version": "0",
    "id": "05d4d2d5-9c76-bfea-72d2-d4614802adb4",
    "detail-type": "FIS Experiment Autoshift In Progress",
    "source": "aws.arc-zonal-shift",
    "account": "111122223333",
    "time": "2023-11-16T23:38:14Z",
    "region": "us-east-1",
    "resources": [
        "TEST-EXAMPLE-2023-11-16-23-28-11-5"
    ],
    "detail": {
        "version": "0.0.1",
```

```
        "data": "",
        "metadata": {
            "awayFrom": "use1-az2",
            "notes":""
        }
    }
}
```

Tentukan grup CloudWatch log yang akan digunakan sebagai target

Saat membuat EventBridge aturan, Anda harus menentukan target tempat peristiwa yang cocok dengan aturan dikirim. Untuk daftar target yang tersedia EventBridge, lihat Target yang tersedia di EventBridge konsol. Salah satu target yang dapat Anda tambahkan ke EventBridge aturan adalah grup CloudWatch log Amazon. Bagian ini menjelaskan persyaratan untuk menambahkan grup CloudWatch log sebagai target, dan menyediakan prosedur untuk menambahkan grup log saat Anda membuat aturan.

Untuk menambahkan grup CloudWatch log sebagai target, Anda dapat melakukan salah satu hal berikut:

- Buat grup log baru
- Pilih grup log yang ada

Jika Anda menentukan grup log baru menggunakan konsol saat membuat aturan, EventBridge secara otomatis membuat grup log untuk Anda. Pastikan grup log yang Anda gunakan sebagai target EventBridge aturan dimulai dengan`/aws/events`. Jika Anda ingin memilih grup log yang ada, ketahuilah bahwa hanya grup log yang dimulai dengan yang `/aws/events` muncul sebagai opsi di menu tarik-turun. Untuk informasi selengkapnya, lihat Membuat grup log baru di Panduan CloudWatch Pengguna Amazon.

Jika Anda membuat atau menggunakan grup CloudWatch log untuk digunakan sebagai target menggunakan CloudWatch operasi di luar konsol, pastikan Anda menetapkan izin dengan benar. Jika Anda menggunakan konsol untuk menambahkan grup log ke EventBridge aturan, maka kebijakan berbasis sumber daya untuk grup log diperbarui secara otomatis. Namun, jika Anda menggunakan AWS Command Line Interface atau AWS SDK untuk menentukan grup log, Anda harus memperbarui kebijakan berbasis sumber daya untuk grup log. Contoh kebijakan berikut menggambarkan izin yang harus Anda tentukan dalam kebijakan berbasis sumber daya untuk grup log:

```
{
  "Statement": [
    {
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "events.amazonaws.com",
          "delivery.logs.amazonaws.com"
        ]
      },
      "Resource": "arn:aws:logs:region:account:log-group:/aws/events/*:*",
      "Sid": "TrustEventsToStoreLogEvent"
    }
  ],
  "Version": "2012-10-17"
}
```

Anda tidak dapat mengonfigurasi kebijakan berbasis sumber daya untuk grup log menggunakan konsol. Untuk menambahkan izin yang diperlukan ke kebijakan berbasis sumber daya, gunakan operasi API. CloudWatch PutResourcePolicy Kemudian, Anda dapat menggunakan perintah describe-resource-policiesCLI untuk memeriksa apakah kebijakan Anda diterapkan dengan benar.

Untuk membuat aturan untuk acara sumber daya dan menentukan target grup CloudWatch log

1. Buka EventBridge konsol Amazon di https://console.aws.amazon.com/events/.
2. Pilih aturan Wilayah AWS yang ingin Anda buat.
3. Pilih Buat aturan lalu masukkan informasi apa pun tentang aturan itu, seperti pola acara atau detail jadwal.

   Untuk informasi selengkapnya tentang membuat EventBridge aturan untuk ARC, lihat bagian sebelumnya dalam topik ini.
4. Pada halaman Pilih target, pilih CloudWatchsebagai target Anda.
5. Pilih grup CloudWatch log dari menu tarik-turun.

Identity and Access Management untuk pergeseran otomatis zona di ARC

AWS Identity and Access Management (IAM) adalah Layanan AWS yang membantu administrator mengontrol akses ke AWS sumber daya dengan aman. Administrator IAM mengontrol siapa yang dapat diautentikasi (masuk) dan diberi wewenang (memiliki izin) untuk menggunakan sumber daya ARC. IAM adalah Layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

Daftar Isi

Bagaimana pergeseran otomatis zona di ARC bekerja dengan IAM

Sebelum Anda menggunakan IAM untuk mengelola akses ke pergeseran otomatis zona di Amazon Application Recovery Controller (ARC), pelajari fitur IAM apa yang tersedia untuk digunakan dengan pergeseran otomatis zonal.

Fitur IAM yang dapat Anda gunakan dengan zonal autoshift di ARC

Fitur IAM	Dukungan autoshift zonal
Kebijakan berbasis identitas	Ya
Kebijakan berbasis sumber daya	Tidak
Tindakan kebijakan	Ya
Sumber daya kebijakan	Ya
Kunci kondisi kebijakan	Ya
ACLs	Tidak
ABAC (tanda dalam kebijakan)	Parsial
Kredensial sementara	Ya

Fitur IAM	Dukungan autoshift zonal
Izin principal	Ya
Peran layanan	Tidak
Peran terkait layanan	Ya

Untuk mendapatkan tampilan tingkat tinggi dan keseluruhan tentang cara kerja AWS layanan dengan sebagian besar fitur IAM, lihat AWS layanan yang bekerja dengan IAM di Panduan Pengguna IAM.

Kebijakan berbasis identitas untuk ARC

Mendukung kebijakan berbasis identitas: Ya

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, seperti pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana, dan berdasarkan kondisi seperti apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Tentukan izin IAM kustom dengan kebijakan terkelola pelanggan dalam Panduan Pengguna IAM.

Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan secara spesifik apakah tindakan dan sumber daya diizinkan atau ditolak, serta kondisi yang menjadi dasar dikabulkan atau ditolaknya tindakan tersebut. Anda tidak dapat menentukan secara spesifik prinsipal dalam sebuah kebijakan berbasis identitas karena prinsipal berlaku bagi pengguna atau peran yang melekat kepadanya. Untuk mempelajari semua elemen yang dapat Anda gunakan dalam kebijakan JSON, lihat Referensi elemen kebijakan JSON IAM dalam Panduan Pengguna IAM.

Untuk melihat contoh kebijakan berbasis identitas ARC, lihat. Contoh kebijakan berbasis identitas di Amazon Application Recovery Controller (ARC)

Kebijakan berbasis sumber daya dalam ARC

Mendukung kebijakan berbasis sumber daya: Tidak

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu.

Tindakan kebijakan untuk ARC

Mendukung tindakan kebijakan: Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen `Action` dari kebijakan JSON menjelaskan tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam sebuah kebijakan. Tindakan kebijakan biasanya memiliki nama yang sama dengan operasi AWS API terkait. Ada beberapa pengecualian, misalnya tindakan hanya izin yang tidak memiliki operasi API yang cocok. Ada juga beberapa operasi yang memerlukan beberapa tindakan dalam suatu kebijakan. Tindakan tambahan ini disebut tindakan dependen.

Sertakan tindakan dalam kebijakan untuk memberikan izin untuk melakukan operasi terkait.

Untuk melihat daftar tindakan ARC untuk pergeseran otomatis zona, lihat Tindakan yang ditentukan oleh Amazon Route 53 Zonal Shift di Referensi Otorisasi Layanan.

Tindakan kebijakan di ARC untuk pergeseran otomatis zona menggunakan awalan berikut sebelum tindakan:

```
arc-zonal-shift
```

Untuk menetapkan secara spesifik beberapa tindakan dalam satu pernyataan, pisahkan tindakan tersebut dengan koma. Misalnya, berikut ini:

```
"Action": [
      "arc-zonal-shift:action1",
      "arc-zonal-shift:action2"
         ]
```

Anda dapat menentukan beberapa tindakan menggunakan wildcard (*). Sebagai contoh, untuk menentukan semua tindakan yang dimulai dengan kata `Describe`, sertakan tindakan berikut:

```
"Action": "arc-zonal-shift:Describe*"
```

Untuk melihat contoh kebijakan berbasis identitas ARC untuk pergeseran otomatis zona, lihat. Contoh kebijakan berbasis identitas untuk pergeseran otomatis zona di ARC

Sumber daya kebijakan untuk pergeseran otomatis zona di ARC

Mendukung sumber daya kebijakan: Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen kebijakan JSON `Resource` menentukan objek yang menjadi target penerapan tindakan. Pernyataan harus menyertakan elemen `Resource` atau `NotResource`. Praktik terbaiknya, tentukan sumber daya menggunakan Amazon Resource Name (ARN). Anda dapat melakukan ini untuk tindakan yang mendukung jenis sumber daya tertentu, yang dikenal sebagai izin tingkat sumber daya.

Untuk tindakan yang tidak mendukung izin di tingkat sumber daya, misalnya operasi pencantuman, gunakan wildcard (*) untuk menunjukkan bahwa pernyataan tersebut berlaku untuk semua sumber daya.

```
"Resource": "*"
```

Untuk melihat daftar jenis sumber daya dan mereka ARNs, dan tindakan yang dapat Anda tentukan dengan ARN dari setiap sumber daya, lihat topik berikut di Referensi Otorisasi Layanan:

- Tindakan yang ditentukan oleh Amazon Route 53 - Zonal Shift

Untuk melihat tindakan dan sumber daya yang dapat Anda gunakan dengan kunci kondisi, lihat topik berikut di Referensi Otorisasi Layanan:

- Kunci kondisi ditentukan oleh Amazon Route 53 - Zonal Shift

Untuk melihat contoh kebijakan berbasis identitas ARC untuk pergeseran otomatis zona, lihat. Contoh kebijakan berbasis identitas untuk pergeseran otomatis zona di ARC

Kunci kondisi kebijakan untuk pergeseran otomatis zona di ARC

Mendukung kunci kondisi kebijakan khusus layanan: Yes

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen `Condition` (atau blok `Condition`) akan memungkinkan Anda menentukan kondisi yang menjadi dasar suatu pernyataan berlaku. Elemen `Condition` bersifat opsional. Anda dapat membuat ekspresi bersyarat yang menggunakan operator kondisi, misalnya sama dengan atau kurang dari, untuk mencocokkan kondisi dalam kebijakan dengan nilai-nilai yang diminta.

Jika Anda menentukan beberapa elemen `Condition` dalam sebuah pernyataan, atau beberapa kunci dalam elemen `Condition` tunggal, maka AWS akan mengevaluasinya menggunakan operasi `AND` logis. Jika Anda menentukan beberapa nilai untuk satu kunci kondisi, AWS mengevaluasi kondisi menggunakan `OR` operasi logis. Semua kondisi harus dipenuhi sebelum izin pernyataan diberikan.

Anda juga dapat menggunakan variabel placeholder saat menentukan kondisi. Sebagai contoh, Anda dapat memberikan izin kepada pengguna IAM untuk mengakses sumber daya hanya jika izin tersebut mempunyai tanda yang sesuai dengan nama pengguna IAM mereka. Untuk informasi selengkapnya, lihat Elemen kebijakan IAM: variabel dan tanda dalam Panduan Pengguna IAM.

AWS mendukung kunci kondisi global dan kunci kondisi khusus layanan. Untuk melihat semua kunci kondisi AWS global, lihat kunci konteks kondisi AWS global di Panduan Pengguna IAM.

Untuk melihat daftar kunci kondisi ARC untuk pergeseran otomatis zona, lihat topik berikut di Referensi Otorisasi Layanan:

- Kunci kondisi untuk Amazon Route 53 Zonal Shift

Untuk melihat tindakan dan sumber daya yang dapat Anda gunakan dengan kunci kondisi, lihat topik berikut di Referensi Otorisasi Layanan:

- Tindakan yang ditentukan oleh Amazon Route 53 Zonal Shift

Untuk melihat contoh kebijakan berbasis identitas ARC untuk pergeseran otomatis zona, lihat. Contoh kebijakan berbasis identitas untuk pergeseran otomatis zona di ARC

Daftar kontrol akses (ACLs) di ARC

Mendukung ACLs: Tidak

Access control lists (ACLs) mengontrol prinsipal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACLs mirip dengan kebijakan berbasis sumber daya, meskipun mereka tidak menggunakan format dokumen kebijakan JSON.

Kontrol akses berbasis atribut (ABAC) dengan ARC

Mendukung ABAC (tag dalam kebijakan): Sebagian

Kontrol akses berbasis atribut (ABAC) adalah strategi otorisasi yang menentukan izin berdasarkan atribut. Dalam AWS, atribut ini disebut tag. Anda dapat melampirkan tag ke entitas IAM (pengguna atau peran) dan ke banyak AWS sumber daya. Penandaan ke entitas dan sumber daya adalah langkah pertama dari ABAC. Kemudian rancanglah kebijakan ABAC untuk mengizinkan operasi ketika tanda milik prinsipal cocok dengan tanda yang ada di sumber daya yang ingin diakses.

ABAC sangat berguna di lingkungan yang berkembang dengan cepat dan berguna di situasi saat manajemen kebijakan menjadi rumit.

Untuk mengendalikan akses berdasarkan tanda, berikan informasi tentang tanda di elemen kondisi dari kebijakan menggunakan kunci kondisi `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, atau `aws:TagKeys`.

Jika sebuah layanan mendukung ketiga kunci kondisi untuk setiap jenis sumber daya, nilainya adalah Ya untuk layanan tersebut. Jika suatu layanan mendukung ketiga kunci kondisi untuk hanya beberapa jenis sumber daya, nilainya adalah Parsial.

Untuk informasi selengkapnya tentang ABAC, lihat Tentukan izin dengan otorisasi ABAC dalam Panduan Pengguna IAM. Untuk melihat tutorial yang menguraikan langkah-langkah pengaturan ABAC, lihat Menggunakan kontrol akses berbasis atribut (ABAC) dalam Panduan Pengguna IAM.

Autoshift zonal di ARC mencakup dukungan parsional berikut untuk ABAC:

- Zonal autoshift mendukung ABAC untuk sumber daya terkelola yang terdaftar di ARC untuk pergeseran zona. Untuk informasi selengkapnya tentang ABAC untuk Network Load Balancer dan sumber daya yang dikelola Application Load Balancer, lihat ABAC dengan Elastic Load Balancing dalam Panduan Pengguna Elastic Load Balancing.

Menggunakan kredensi sementara dengan ARC

Mendukung kredensial sementara: Ya

Beberapa Layanan AWS tidak berfungsi saat Anda masuk menggunakan kredensi sementara. Untuk informasi tambahan, termasuk yang Layanan AWS bekerja dengan kredensi sementara, lihat Layanan AWS yang bekerja dengan IAM di Panduan Pengguna IAM.

Anda menggunakan kredensi sementara jika Anda masuk AWS Management Console menggunakan metode apa pun kecuali nama pengguna dan kata sandi. Misalnya, ketika Anda mengakses AWS menggunakan tautan masuk tunggal (SSO) perusahaan Anda, proses tersebut secara otomatis membuat kredensitas sementara. Anda juga akan secara otomatis membuat kredensial sementara ketika Anda masuk ke konsol sebagai seorang pengguna lalu beralih peran. Untuk informasi selengkapnya tentang peralihan peran, lihat Beralih dari pengguna ke peran IAM (konsol) dalam Panduan Pengguna IAM.

Anda dapat membuat kredenal sementara secara manual menggunakan API AWS CLI atau AWS . Anda kemudian dapat menggunakan kredensi sementara tersebut untuk mengakses. AWS AWS merekomendasikan agar Anda secara dinamis menghasilkan kredensi sementara alih-alih menggunakan kunci akses jangka panjang. Untuk informasi selengkapnya, lihat Kredensial keamanan sementara di IAM.

Izin utama lintas layanan untuk ARC

Mendukung sesi akses maju (FAS): Ya

Saat Anda menggunakan entitas IAM (pengguna atau peran) untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Kebijakan memberikan izin kepada principal. Saat Anda menggunakan beberapa layanan, Anda mungkin melakukan tindakan yang kemudian memicu tindakan lain di layanan yang berbeda. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut.

Untuk melihat apakah suatu tindakan memerlukan tindakan dependen tambahan dalam kebijakan, lihat topik berikut di Referensi Otorisasi Layanan:

- Amazon Route 53 Pergeseran Zonal

Peran layanan untuk ARC

Mendukung peran layanan: Tidak

Peran layanan adalah peran IAM yang diambil oleh sebuah layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, mengubah, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat Buat sebuah peran untuk mendelegasikan izin ke Layanan AWS dalam Panduan pengguna IAM.

Peran terkait layanan untuk ARC

Mendukung peran terkait layanan: Ya

Peran terkait layanan adalah jenis peran layanan yang ditautkan ke. Layanan AWS Layanan tersebut dapat menjalankan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Anda Akun AWS dan dimiliki oleh layanan. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.

Untuk detail tentang membuat atau mengelola peran terkait layanan ARC, lihat. Menggunakan peran terkait layanan untuk pergeseran otomatis zona di ARC

Untuk detail tentang pembuatan atau manajemen peran terkait layanan, lihat Layanan AWS yang berfungsi dengan IAM. Cari layanan dalam tabel yang memiliki `Yes` di kolom Peran terkait layanan. Pilih tautan Ya untuk melihat dokumentasi peran terkait layanan untuk layanan tersebut.

Contoh kebijakan berbasis identitas untuk pergeseran otomatis zona di ARC

Secara default, pengguna dan peran tidak memiliki izin untuk membuat atau memodifikasi sumber daya ARC. Mereka juga tidak dapat melakukan tugas dengan menggunakan AWS Management Console, AWS Command Line Interface (AWS CLI), atau AWS API. Untuk memberikan izin kepada pengguna untuk melakukan tindakan di sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM. Administrator kemudian dapat menambahkan kebijakan IAM ke peran, dan pengguna dapat mengambil peran.

Untuk mempelajari cara membuat kebijakan berbasis identitas IAM dengan menggunakan contoh dokumen kebijakan JSON ini, lihat Membuat kebijakan IAM (konsol) di Panduan Pengguna IAM.

Untuk detail tentang tindakan dan jenis sumber daya yang ditentukan oleh ARC, termasuk format ARNs untuk setiap jenis sumber daya, lihat Kunci tindakan, sumber daya, dan kondisi untuk Amazon Application Recovery Controller (ARC) di Referensi Otorisasi Layanan.

Topik

- Praktik terbaik kebijakan
- Contoh: Akses konsol pergeseran otomatis Zonal
- Contoh: Tindakan ARC API

Praktik terbaik kebijakan

Kebijakan berbasis identitas menentukan apakah seseorang dapat membuat, mengakses, atau menghapus sumber daya ARC di akun Anda. Tindakan ini membuat Akun AWS Anda dikenai biaya. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

- Mulailah dengan kebijakan AWS terkelola dan beralih ke izin hak istimewa paling sedikit — Untuk mulai memberikan izin kepada pengguna dan beban kerja Anda, gunakan kebijakan AWS terkelola yang memberikan izin untuk banyak kasus penggunaan umum. Mereka tersedia di Anda Akun AWS. Kami menyarankan Anda mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola AWS pelanggan yang khusus untuk kasus penggunaan Anda. Untuk informasi selengkapnya, lihat Kebijakan yang dikelola AWS atau Kebijakan yang dikelola AWS untuk fungsi tugas dalam Panduan Pengguna IAM.
- Menerapkan izin dengan hak akses paling rendah – Ketika Anda menetapkan izin dengan kebijakan IAM, hanya berikan izin yang diperlukan untuk melakukan tugas. Anda melakukannya dengan mendefinisikan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi tertentu, yang juga dikenal sebagai izin dengan hak akses paling rendah. Untuk informasi selengkapnya tentang cara menggunakan IAM untuk mengajukan izin, lihat Kebijakan dan izin dalam IAM dalam Panduan Pengguna IAM.
- Gunakan kondisi dalam kebijakan IAM untuk membatasi akses lebih lanjut – Anda dapat menambahkan suatu kondisi ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. Sebagai contoh, Anda dapat menulis kondisi kebijakan untuk menentukan bahwa semua permintaan harus dikirim menggunakan SSL. Anda juga dapat menggunakan ketentuan untuk memberikan akses ke tindakan layanan jika digunakan melalui yang spesifik Layanan AWS, seperti AWS CloudFormation. Untuk informasi selengkapnya, lihat Elemen kebijakan JSON IAM: Kondisi dalam Panduan Pengguna IAM.
- Gunakan IAM Access Analyzer untuk memvalidasi kebijakan IAM Anda untuk memastikan izin yang aman dan fungsional – IAM Access Analyzer memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan tersebut mematuhi bahasa kebijakan IAM (JSON) dan praktik terbaik IAM. IAM Access Analyzer menyediakan lebih dari 100 pemeriksaan kebijakan dan rekomendasi yang dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang aman dan fungsional. Untuk informasi selengkapnya, lihat Validasi kebijakan dengan IAM Access Analyzer dalam Panduan Pengguna IAM.
- Memerlukan otentikasi multi-faktor (MFA) - Jika Anda memiliki skenario yang mengharuskan pengguna IAM atau pengguna root di Anda, Akun AWS aktifkan MFA untuk keamanan tambahan. Untuk meminta MFA ketika operasi API dipanggil, tambahkan kondisi MFA pada kebijakan Anda. Untuk informasi selengkapnya, lihat Amankan akses API dengan MFA dalam Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang praktik terbaik dalam IAM, lihat Praktik terbaik keamanan di IAM dalam Panduan Pengguna IAM.

Contoh: Akses konsol pergeseran otomatis Zonal

Untuk mengakses konsol Amazon Application Recovery Controller (ARC), Anda harus memiliki seperangkat izin minimum. Izin ini harus memungkinkan Anda untuk membuat daftar dan melihat detail tentang sumber daya ARC di Anda Akun AWS. Jika Anda membuat kebijakan berbasis identitas yang lebih ketat daripada izin minimum yang diperlukan, konsol tidak akan berfungsi sebagaimana mestinya untuk entitas (pengguna atau peran) dengan kebijakan tersebut.

Anda tidak perlu mengizinkan izin konsol minimum untuk pengguna yang melakukan panggilan hanya ke AWS CLI atau AWS API. Sebagai gantinya, izinkan akses hanya ke tindakan yang sesuai dengan operasi API yang coba mereka lakukan.

Untuk melakukan beberapa tugas, pengguna harus memiliki izin untuk membuat peran terkait layanan yang terkait dengan pergeseran otomatis zona di ARC. Untuk mempelajari selengkapnya, lihat Menggunakan peran terkait layanan untuk pergeseran otomatis zona di ARC.

Untuk memberi pengguna akses penuh untuk menggunakan pergeseran otomatis zona di AWS Management Console, lampirkan kebijakan seperti berikut ini kepada pengguna:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                  "arc-zonal-shift:ListManagedResources",
                  "arc-zonal-shift:GetManagedResource",
                  "arc-zonal-shift:ListZonalShifts",
                  "arc-zonal-shift:StartZonalShift",
                  "arc-zonal-shift:UpdateZonalShift",
                  "arc-zonal-shift:CancelZonalShift",
                  "arc-zonal-shift:CreatePracticeRunConfiguration",
                  "arc-zonal-shift:DeletePracticeRunConfiguration",
                  "arc-zonal-shift:ListAutoshifts",
                  "arc-zonal-shift:UpdatePracticeRunConfiguration",
                  "arc-zonal-shift:UpdateZonalAutoshiftConfiguration"
             ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:DescribeAvailabilityZones",
```

```
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "cloudwatch:DescribeAlarms",
            "Resource": "*"
        }
    ]
}
```

Contoh: Tindakan ARC API

Anda dapat menggunakan kebijakan untuk memastikan bahwa pengguna dapat menggunakan tindakan ARC API untuk pergeseran otomatis zona guna mengonfigurasi pergeseran otomatis zona sehingga AWS mengalihkan lalu lintas sumber daya aplikasi dari Availability Zone, atas nama Anda, menjadi sehat AZs di, untuk membantu mengurangi waktu Anda menuju pemulihan selama acara. Wilayah AWS Untuk memberikan izin ini, lampirkan kebijakan yang sesuai dengan operasi API yang perlu dikerjakan pengguna, seperti yang dijelaskan di bawah ini.

Untuk melakukan beberapa tugas, pengguna harus memiliki izin untuk peran terkait layanan yang terkait dengan ARC. Izin yang diperlukan untuk membuat peran terkait layanan disertakan dalam kebijakan contoh berikut. Untuk mempelajari selengkapnya, lihat Menggunakan peran terkait layanan untuk pergeseran otomatis zona di ARC.

Untuk bekerja dengan operasi API untuk pergeseran otomatis zona, lampirkan kebijakan seperti berikut ini ke pengguna:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                   "arc-zonal-shift:ListManagedResources",
                   "arc-zonal-shift:GetManagedResource",
                   "arc-zonal-shift:ListZonalShifts",
                   "arc-zonal-shift:StartZonalShift",
                   "arc-zonal-shift:UpdateZonalShift",
                   "arc-zonal-shift:CancelZonalShift",
                   "arc-zonal-shift:CreatePracticeRunConfiguration",
                   "arc-zonal-shift:DeletePracticeRunConfiguration",
                   "arc-zonal-shift:ListAutoshifts",
```

```
                "arc-zonal-shift:UpdatePracticeRunConfiguration",
                "arc-zonal-shift:UpdateZonalAutoshiftConfiguration"
            ],
            "Resource": "*"
        },
        {
            "Effect" : "Allow",
            "Action" : [
                    "cloudwatch:DescribeAlarms",
                    "health:DescribeEvents"
            ],
            "Resource" : "*"
        },
        {
            "Effect" : "Allow",
            "Action" : [
                    "arc-zonal-shift:CancelZonalShift",
                    "arc-zonal-shift:GetManagedResource",
                    "arc-zonal-shift:StartZonalShift",
                    "arc-zonal-shift:UpdateZonalShift"
            ],
            "Resource" : "*"
        }
    ]
}
```

Menggunakan peran terkait layanan untuk pergeseran otomatis zona di ARC

Pergeseran otomatis zona di Amazon Application Recovery Controller menggunakan peran terkait AWS Identity and Access Management layanan (IAM). Peran terkait layanan adalah jenis unik peran IAM yang ditautkan langsung ke layanan — dalam hal ini, ARC. Peran terkait layanan telah ditentukan sebelumnya oleh ARC dan mencakup semua izin yang diperlukan layanan untuk memanggil AWS layanan lain atas nama Anda untuk tujuan tertentu.

Peran terkait layanan membuat pengaturan ARC lebih mudah karena Anda tidak perlu menambahkan izin yang diperlukan secara manual. ARC mendefinisikan izin untuk peran terkait layanan, dan kecuali ditentukan lain, hanya ARC yang dapat mengambil perannya. Izin-izin yang ditentukan mencakup kebijakan kepercayaan dan kebijakan izin, serta bahwa kebijakan izin tidak dapat dilampirkan ke entitas IAM lainnya.

Anda dapat menghapus peran tertaut layanan hanya setelah terlebih dahulu menghapus sumber dayanya yang terkait. Ini melindungi sumber daya autoshift zona ARC Anda karena Anda tidak dapat secara tidak sengaja menghapus izin untuk mengakses sumber daya.

Untuk informasi tentang layanan lain yang mendukung peran terkait layanan, lihat AWS Layanan yang bekerja dengan IAM dan cari layanan yang memiliki Ya di kolom peran terkait layanan. Pilih Ya bersama tautan untuk melihat dokumentasi peran tertaut layanan untuk layanan tersebut.

Izin peran terkait layanan untuk AWSService RoleForZonalAutoshiftPracticeRun

ARC menggunakan peran terkait layanan bernama AWSServiceRoleForZonalAutoshiftPracticeRununtuk melakukan hal berikut:

- Pantau CloudWatch alarm Amazon dan AWS Health Dashboard acara pelanggan yang disediakan pelanggan untuk menjalankan praktik
- Kelola latihan lari (berlatih shift zona)

Bagian ini menjelaskan izin untuk peran terkait layanan, dan informasi tentang membuat, mengedit, dan menghapus peran.

Izin peran terkait layanan untuk AWSService RoleForZonalAutoshiftPracticeRun

Peran terkait layanan ini menggunakan kebijakan terkelola. `AWSZonalAutoshiftPracticeRunSLRPolicy`

Peran terkait layanan AWSServiceRoleForZonalAutoshiftPracticeRun memercayai layanan berikut untuk mengambil peran tersebut:

- `practice-run.arc-zonal-shift.amazonaws.com`

Untuk melihat izin kebijakan ini, lihat AWSZonalAutoshiftPracticeRunSLRPolicydi Referensi Kebijakan AWS Terkelola.

Anda harus mengonfigurasi izin untuk mengizinkan entitas IAM (seperti pengguna, grup, atau peran) untuk membuat, mengedit, atau menghapus peran terkait layanan. Untuk informasi selengkapnya, lihat Izin peran tertaut layanan dalam Panduan Pengguna IAM.

Membuat peran AWSServiceRoleForZonalAutoshiftPracticeRunterkait layanan untuk ARC

Anda tidak perlu membuat peran tertaut layanan AWSServiceRoleForZonalAutoshiftPracticeRun secara manual. Saat Anda membuat konfigurasi run praktik pertama di AWS Management Console, the, atau AWS SDK AWS CLI, ARC akan membuat peran terkait layanan untuk Anda.

Jika Anda menghapus peran tertaut layanan ini, dan ingin membuatnya lagi, Anda dapat mengulangi proses yang sama untuk membuat kembali peran tersebut di akun Anda. Saat Anda membuat konfigurasi run praktik pertama, ARC akan membuat peran terkait layanan untuk Anda lagi.

Mengedit peran AWSServiceRoleForZonalAutoshiftPracticeRunterkait layanan untuk ARC

ARC tidak mengizinkan Anda mengedit peran AWSServiceRoleForZonalAutoshiftPracticeRunterkait layanan. Setelah membuat peran terkait layanan, Anda tidak dapat mengubah nama peran karena entitas lain mungkin mereferensikannya. Namun, Anda dapat mengedit penjelasan peran menggunakan IAM. Untuk informasi selengkapnya, lihat Mengedit peran tertaut layanan dalam Panduan Pengguna IAM.

Menghapus peran AWSServiceRoleForZonalAutoshiftPracticeRunterkait layanan untuk ARC

Jika Anda tidak perlu lagi menggunakan fitur atau layanan yang memerlukan peran terkait layanan, kami merekomendasikan Anda menghapus peran tersebut. Dengan begitu, Anda tidak memiliki entitas yang tidak digunakan yang tidak dipantau atau dipelihara secara aktif. Namun, Anda harus membersihkan sumber daya untuk peran terkait layanan sebelum Anda dapat menghapusnya secara manual.

Setelah Anda menonaktifkan autoshift, maka Anda dapat menghapus peran AWSServiceRoleForZonalAutoshiftPracticeRunterkait layanan. Untuk informasi selengkapnya tentang kemampuan autoshift, lihatPergeseran zona di ARC.

> **Note**
>
> Jika layanan ARC menggunakan peran saat Anda mencoba menghapus sumber daya, maka penghapusan peran layanan mungkin gagal. Jika itu terjadi, tunggu beberapa menit dan coba lagi untuk menghapus peran.

Untuk menghapus peran terkait layanan secara manual menggunakan IAM

Gunakan konsol IAM, the AWS CLI, atau AWS API untuk menghapus peran AWSService RoleForZonalAutoshiftPracticeRun terkait layanan. Untuk informasi selengkapnya, lihat Menghapus peran tertaut layanan dalam Panduan Pengguna IAM.

Pembaruan peran terkait layanan ARC untuk pergeseran otomatis zona

Untuk pembaruan kebijakan AWS terkelola untuk peran terkait layanan ARC, lihat tabel pembaruan kebijakan AWS terkelola untuk ARC. Anda juga dapat berlangganan peringatan RSS otomatis di halaman riwayat Dokumen ARC.

AWS kebijakan terkelola untuk pergeseran otomatis zona di ARC

Kebijakan AWS terkelola adalah kebijakan mandiri yang dibuat dan dikelola oleh AWS. AWS Kebijakan terkelola dirancang untuk memberikan izin bagi banyak kasus penggunaan umum sehingga Anda dapat mulai menetapkan izin kepada pengguna, grup, dan peran.

Perlu diingat bahwa kebijakan AWS terkelola mungkin tidak memberikan izin hak istimewa paling sedikit untuk kasus penggunaan spesifik Anda karena tersedia untuk digunakan semua pelanggan. AWS Kami menyarankan Anda untuk mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola pelanggan yang khusus untuk kasus penggunaan Anda.

Anda tidak dapat mengubah izin yang ditentukan dalam kebijakan AWS terkelola. Jika AWS memperbarui izin yang ditentukan dalam kebijakan AWS terkelola, pembaruan akan memengaruhi semua identitas utama (pengguna, grup, dan peran) yang dilampirkan kebijakan tersebut. AWS kemungkinan besar akan memperbarui kebijakan AWS terkelola saat baru Layanan AWS diluncurkan atau operasi API baru tersedia untuk layanan yang ada.

Untuk informasi selengkapnya, lihat Kebijakan terkelola AWS dalam Panduan Pengguna IAM.

AWS kebijakan terkelola: AWSZonal AutoshiftPracticeRun SLRPolicy

Anda tidak dapat melampirkan `AWSZonalAutoshiftPracticeRunSLRPolicy` ke entitas IAM Anda. Kebijakan ini dilampirkan ke peran terkait layanan yang memungkinkan Amazon Application Recovery Controller (ARC) melakukan hal berikut untuk pergeseran otomatis zona:

- Pantau CloudWatch alarm Amazon dan AWS Health Dashboard acara pelanggan yang disediakan pelanggan untuk menjalankan praktik
- Kelola latihan lari (berlatih shift zona)
- Kelola pemeriksaan kapasitas seimbang untuk latihan berjalan dan pergeseran otomatis

Untuk informasi selengkapnya, lihat Menggunakan peran terkait layanan untuk pergeseran otomatis zona di ARC.

Pembaruan untuk kebijakan AWS terkelola untuk pergeseran otomatis zona

Untuk detail tentang pembaruan kebijakan AWS terkelola untuk pergeseran otomatis zona di ARC sejak layanan ini mulai melacak perubahan ini, lihat. Pembaruan kebijakan AWS terkelola untuk Amazon Application Recovery Controller (ARC) Untuk peringatan otomatis tentang perubahan pada halaman ini, berlangganan umpan RSS di halaman riwayat Dokumen ARC.

Gunakan kontrol perutean untuk memulihkan aplikasi Multi-region di ARC

Bagian ini menjelaskan cara menggunakan kemampuan kontrol perutean di Amazon Application Recovery Controller (ARC) untuk meminimalkan gangguan dan membantu memberikan kontinuitas bagi pengguna Anda ketika Anda memiliki AWS aplikasi yang digunakan dalam beberapa. Wilayah AWS

Anda juga dapat mempelajari tentang pemeriksaan kesiapan, kemampuan dalam ARC yang dapat Anda gunakan untuk mendapatkan wawasan tentang apakah aplikasi dan sumber daya Anda siap untuk pemulihan.

Topik di bagian ini menjelaskan kontrol perutean dan kemampuan pemeriksaan kesiapan, cara mengaturnya, dan cara menggunakannya.

Topik

Kontrol perutean di ARC

Untuk gagal melewati lalu lintas ke replika aplikasi dalam beberapa Wilayah AWS, Anda dapat menggunakan kontrol perutean di Amazon Application Recovery Controller (ARC) yang terintegrasi dengan jenis pemeriksaan kesehatan tertentu di Amazon Route 53. Kontrol perutean adalah sakelar on-off sederhana yang memungkinkan Anda mengalihkan lalu lintas klien dari satu replika Regional ke replika Regional lainnya. Pengalihan rute lalu lintas dilakukan dengan merutekan pemeriksaan kesehatan kontrol yang diatur dengan catatan DNS Amazon Route 53. Misalnya, catatan failover DNS, yang terkait dengan nama domain yang menampilkan replika aplikasi Anda di setiap Region.

Bagian ini menjelaskan cara kerja kontrol perutean, cara mengatur komponen kontrol perutean, dan cara menggunakannya untuk mengubah rute lalu lintas untuk failover.

Komponen kontrol routing di ARC adalah: cluster, panel kontrol, kontrol routing, dan pemeriksaan kesehatan kontrol routing. Semua kontrol routing dikelompokkan pada panel kontrol. Anda dapat mengelompokkannya di panel kontrol default yang dibuat ARC untuk klaster Anda, atau membuat

panel kontrol kustom Anda sendiri. Anda harus membuat cluster sebelum Anda dapat membuat panel kontrol atau kontrol routing. Setiap cluster di ARC adalah bidang data dari titik akhir dalam lima Wilayah AWS.

Setelah membuat kontrol perutean dan pemeriksaan kesehatan kontrol perutean, Anda dapat membuat aturan keselamatan untuk kontrol perutean untuk membantu mencegah efek samping otomatisasi pemulihan yang tidak disengaja. Anda dapat memperbarui status kontrol perutean untuk mengubah rute lalu lintas, secara individu atau dalam kelompok, dengan menggunakan tindakan AWS CLI atau API (disarankan), atau dengan menggunakan. AWS Management Console

Bagian ini menjelaskan cara kerja kontrol routing, dan cara membuat dan menggunakannya untuk mengalihkan lalu lintas untuk aplikasi Anda.

> **Important**
>
> Untuk mempelajari persiapan menggunakan ARC untuk mengubah rute lalu lintas sebagai bagian dari rencana failover untuk aplikasi Anda dalam skenario bencana, lihat. Praktik terbaik untuk kontrol perutean di ARC

Tentang kontrol perutean

Kontrol perutean mengalihkan lalu lintas dengan menggunakan pemeriksaan kesehatan di Amazon Route 53 yang dikonfigurasi dengan catatan DNS yang terkait dengan sumber daya tingkat atas sel dalam grup pemulihan Anda, seperti penyeimbang beban Elastic Load Balancing. Anda dapat mengarahkan lalu lintas dari satu sel ke sel lainnya, misalnya, dengan memperbarui status kontrol perutean ke `Off` (untuk menghentikan arus lalu lintas ke satu sel) dan memperbarui status kontrol perutean lainnya ke On (untuk memulai arus lalu lintas ke sel lain). Proses yang mengubah arus lalu lintas adalah pemeriksaan kesehatan Route 53 yang terkait dengan kontrol perutean, setelah ARC memperbaruinya untuk mengaturnya sebagai sehat atau tidak sehat, berdasarkan status kontrol perutean yang sesuai.

Kontrol perutean mendukung failover di semua AWS layanan yang memiliki titik akhir DNS. Anda dapat memperbarui status kontrol perutean agar gagal atas lalu lintas untuk pemulihan bencana, atau ketika Anda mendeteksi penurunan latensi untuk aplikasi Anda, atau masalah lainnya.

Anda juga dapat mengonfigurasi aturan keselamatan untuk kontrol perutean, untuk memastikan bahwa mengalihkan lalu lintas dengan menggunakan kontrol perutean tidak mengganggu

ketersediaan. Untuk informasi selengkapnya, lihat Membuat aturan keselamatan untuk kontrol perutean .

Penting untuk dicatat bahwa kontrol perutean bukanlah pemeriksaan kesehatan yang memantau kesehatan yang mendasari titik akhir. Misalnya, tidak seperti pemeriksaan kesehatan Route 53, kontrol perutean tidak memantau waktu respons atau waktu koneksi TCP. Kontrol perutean adalah sakelar on-off sederhana yang mengontrol pemeriksaan kesehatan. Biasanya, Anda mengubah status untuk mengarahkan lalu lintas, dan perubahan status itu memindahkan lalu lintas untuk pergi ke titik akhir tertentu untuk seluruh tumpukan aplikasi, atau mencegah perutean ke seluruh tumpukan aplikasi. Misalnya, dalam skenario sederhana, ketika Anda mengubah status kontrol perutean dari `On` ke`Off`, itu memperbarui pemeriksaan kesehatan Route 53, yang telah Anda kaitkan dengan catatan failover DNS untuk memindahkan lalu lintas dari titik akhir.

Cara menggunakan kontrol perutean

Untuk memperbarui status kontrol perutean, sehingga Anda dapat mengubah rute lalu lintas, Anda harus terhubung ke salah satu titik akhir cluster Anda di ARC. Jika titik akhir yang Anda coba sambungkan tidak tersedia, coba ubah status dengan titik akhir cluster lain. Proses Anda untuk mengubah status kontrol perutean harus disiapkan untuk mencoba setiap titik akhir dalam rotasi, karena titik akhir cluster didaur ulang melalui status yang tersedia dan tidak tersedia untuk pemeliharaan dan pembaruan rutin.

Saat Anda membuat kontrol perutean, Anda mengonfigurasi catatan DNS Anda untuk mengaitkan pemeriksaan kesehatan kontrol perutean dengan nama DNS Route 53 yang mengutamakan setiap replika aplikasi. Misalnya, untuk mengontrol failover lalu lintas di dua penyeimbang beban, satu di masing-masing dua Wilayah, Anda membuat dua pemeriksaan kesehatan kontrol perutean dan mengaitkannya dengan dua catatan DNS, misalnya, catatan Alias dengan kebijakan perutean failover, dengan nama domain masing-masing penyeimbang beban.

Anda juga dapat mengatur skenario failover lalu lintas yang lebih kompleks dengan menggunakan kontrol perutean ARC bersama dengan pemeriksaan kesehatan Route 53 dan kumpulan catatan DNS, menggunakan catatan DNS dengan kebijakan perutean tertimbang. Untuk melihat contoh terperinci, lihat bagian tentang kegagalan lalu lintas pengguna di posting AWS blog berikut: Membangun aplikasi yang sangat tangguh menggunakan Amazon Application Recovery Controller (ARC), Bagian 2: Tumpukan Multi-Region

Ketika Anda memulai failover untuk Wilayah AWS menggunakan kontrol perutean, karena langkah-langkah yang terlibat dengan arus lalu lintas, Anda mungkin tidak melihat lalu lintas keluar dari Wilayah segera. Ini juga dapat memakan waktu singkat untuk menyelesaikan koneksi yang

sedang berlangsung di Wilayah, tergantung pada perilaku klien dan penggunaan kembali koneksi. Bergantung pada pengaturan DNS Anda dan faktor lainnya, koneksi yang ada dapat selesai hanya dalam beberapa menit, atau mungkin memakan waktu lebih lama. Untuk informasi selengkapnya, lihat Memastikan pergeseran lalu lintas selesai dengan cepat.

Manfaat kontrol routing

Kontrol perutean di ARC memiliki beberapa manfaat dibandingkan mengubah rute lalu lintas dengan pemeriksaan kesehatan tradisional. Misalnya:

- Kontrol routing memberi Anda cara untuk gagal di seluruh tumpukan aplikasi. Ini berbeda dengan kegagalan pada masing-masing komponen tumpukan, seperti yang dilakukan EC2 instance Amazon, berdasarkan pemeriksaan kesehatan tingkat sumber daya.
- Kontrol perutean memberi Anda penggantian manual yang aman dan sederhana yang dapat Anda gunakan untuk mengalihkan lalu lintas untuk melakukan pemeliharaan atau memulihkan dari kegagalan ketika monitor internal tidak mendeteksi masalah.
- Anda dapat menggunakan kontrol perutean bersama dengan aturan keselamatan untuk mencegah efek samping umum yang dapat terjadi dengan otomatisasi berbasis pemeriksaan kesehatan yang sepenuhnya otomatis, seperti gagal menggunakan infrastruktur siaga yang tidak siap untuk failover.

Berikut adalah contoh menggabungkan kontrol perutean ke dalam strategi failover Anda, untuk meningkatkan ketahanan dan ketersediaan aplikasi Anda. AWS

Anda dapat mendukung AWS aplikasi yang sangat tersedia AWS dengan menjalankan beberapa (biasanya tiga) replika redundan di seluruh Wilayah. Kemudian Anda dapat menggunakan kontrol perutean Amazon Route 53 untuk merutekan lalu lintas ke replika yang sesuai.

Misalnya, Anda dapat mengatur satu replika aplikasi agar aktif dan melayani lalu lintas aplikasi, sementara yang lain adalah replika siaga. Ketika replika aktif Anda mengalami kegagalan, Anda dapat mengubah rute lalu lintas pengguna di sana untuk memulihkan ketersediaan aplikasi Anda. Anda harus memutuskan apakah akan gagal dari atau ke replika berdasarkan informasi dari sistem pemantauan dan pemeriksaan kesehatan Anda.

Jika Anda ingin mengaktifkan pemulihan yang lebih cepat, opsi lain yang dapat Anda pilih untuk arsitektur Anda adalah implementasi aktif-aktif. Dengan pendekatan ini, replika Anda aktif pada saat yang sama. Ini berarti Anda dapat pulih dari kegagalan dengan memindahkan pengguna dari replika aplikasi yang rusak hanya dengan mengalihkan lalu lintas ke replika aktif lainnya.

AWS Ketersediaan wilayah untuk kontrol perutean

Untuk informasi terperinci tentang dukungan Regional dan titik akhir layanan untuk Amazon Application Recovery Controller (ARC), lihat titik akhir dan kuota Amazon Application Recovery Controller (ARC) di Referensi Umum Amazon Web Services.

> **Note**
>
> Kontrol perutean di Amazon Application Recovery Controller (ARC) adalah fitur global. Namun, Anda harus menentukan Wilayah AS Barat (Oregon) (tentukan parameternya`--region us-west-2`) dalam AWS CLI perintah ARC Regional. Artinya, saat Anda membuat sumber daya seperti cluster, panel kontrol, atau kontrol perutean.

Kontrol perutean ARC adalah on/off sakelar yang mengubah status pemeriksaan kesehatan ARC, yang kemudian dapat dikaitkan dengan catatan DNS yang mengalihkan lalu lintas, misalnya, dari replika penyebaran primer ke siaga.

Jika ada kegagalan aplikasi atau masalah latensi, Anda dapat memperbarui status kontrol perutean untuk mengalihkan lalu lintas dari replika utama Anda ke, misalnya, replika siaga. Dengan menggunakan operasi API bidang data ARC yang sangat andal untuk membuat kueri kontrol perutean dan pembaruan status kontrol perutean, Anda dapat mengandalkan ARC untuk failover selama skenario pemulihan bencana. Untuk informasi selengkapnya, lihat Mendapatkan dan memperbarui status kontrol perutean menggunakan ARC API (disarankan).

ARC mempertahankan status kontrol routing dalam sebuah cluster, yang merupakan satu set dari lima endpoint Regional redundan. ARC menyebarkan perubahan status kontrol perutean di seluruh cluster, yang terletak di EC2 armada Amazon, untuk mendapatkan kuorum di lima Wilayah. AWS Setelah propagasi, saat Anda menanyakan ARC untuk status kontrol perutean, menggunakan API dan bidang data yang sangat andal, ARC akan menampilkan tampilan konsensus.

Anda dapat berinteraksi dengan salah satu dari lima titik akhir cluster untuk memperbarui status kontrol perutean dari, misalnya, `Off` ke. On Kemudian ARC menyebarkan pembaruan di lima Wilayah cluster.

Konsistensi data di kelima titik akhir cluster dicapai dalam waktu rata-rata 5 detik, dan setelah maksimum tidak lebih dari 15 detik.

ARC menawarkan keandalan ekstrim dengan bidang datanya agar Anda gagal secara manual atas aplikasi Anda di seluruh sel. ARC memastikan bahwa setidaknya tiga dari lima titik akhir cluster selalu

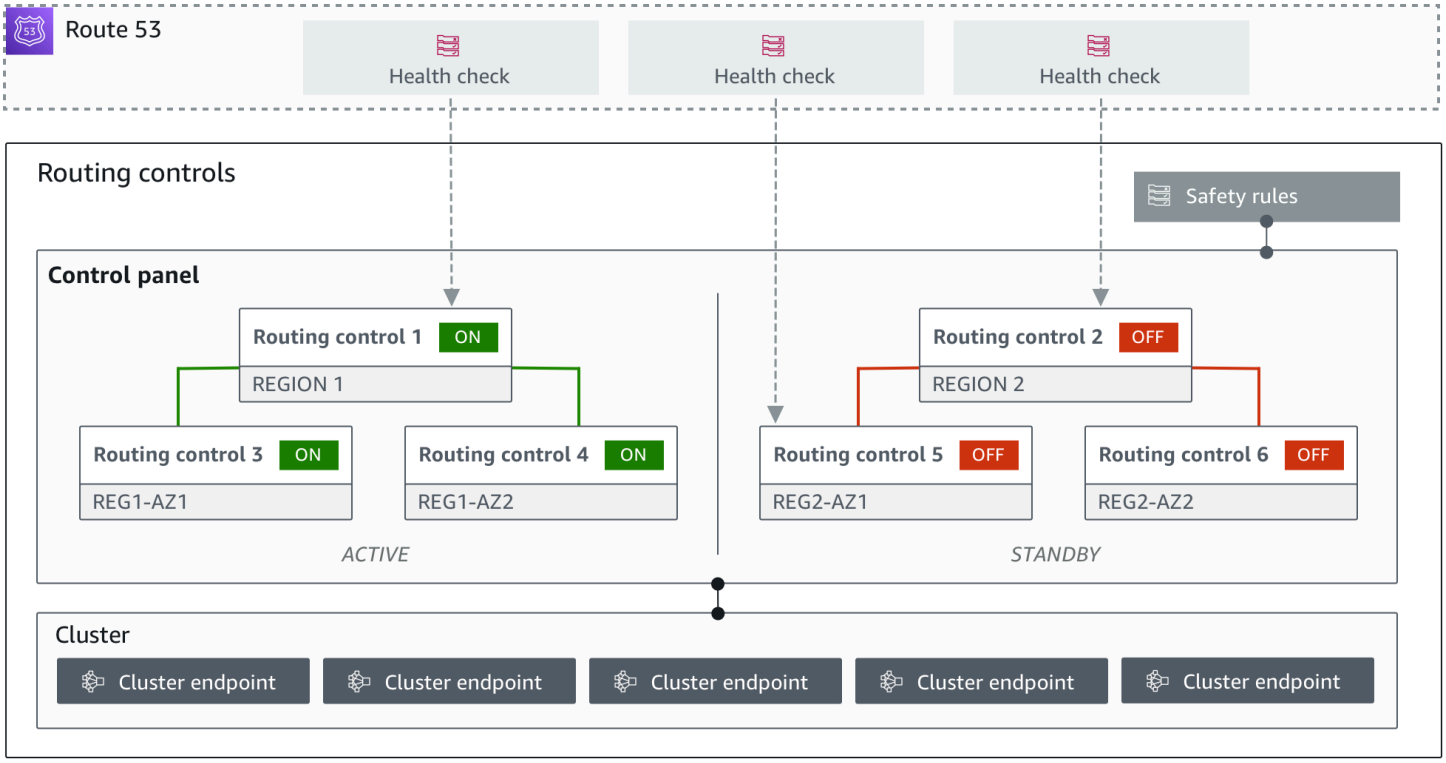
dapat diakses oleh Anda untuk melakukan perubahan status kontrol perutean. Perhatikan bahwa setiap cluster ARC adalah penyewa tunggal, untuk memastikan bahwa Anda tidak terpengaruh oleh “tetangga yang berisik” yang mungkin memperlambat pola akses Anda.

Saat Anda membuat perubahan pada status kontrol perutean, Anda mengandalkan tiga kriteria berikut, yang sangat tidak mungkin gagal:

- Setidaknya tiga dari lima titik akhir Anda tersedia dan ambil bagian dalam kuorum.
- Anda memiliki kredensil IAM yang berfungsi dan dapat mengautentikasi terhadap titik akhir cluster Regional yang berfungsi.
- Pesawat data Route 53 sehat (pesawat data ini dirancang untuk memenuhi SLA ketersediaan 100%).

Komponen kontrol perutean

Diagram berikut menggambarkan contoh komponen yang mendukung fitur kontrol routing di ARC. Kontrol perutean yang ditampilkan di sini (dikelompokkan ke dalam satu panel kontrol) memungkinkan Anda mengelola lalu lintas ke dua Availability Zone di masing-masing dua Wilayah. Saat Anda memperbarui status kontrol perutean, ARC mengubah pemeriksaan kesehatan di Amazon Route 53, yang mengarahkan lalu lintas DNS ke sel yang berbeda. Aturan keselamatan yang Anda konfigurasikan untuk kontrol perutean membantu menghindari skenario kegagalan terbuka dan konsekuensi tidak disengaja lainnya.



Berikut ini adalah komponen fitur kontrol routing di ARC.

Klaster

Cluster adalah sekumpulan lima titik akhir Regional redundan tempat Anda memulai panggilan API untuk memperbarui atau mendapatkan status kontrol perutean. Cluster menyertakan panel kontrol default, dan Anda dapat meng-host beberapa panel kontrol dan kontrol perutean pada satu cluster.

Kontrol perutean

Kontrol perutean adalah on/off sakelar sederhana, yang dihosting di cluster, yang Anda gunakan untuk mengontrol perutean lalu lintas klien masuk dan keluar sel. Saat Anda membuat kontrol perutean, Anda menambahkan pemeriksaan kesehatan ARC di Route 53. Ini memungkinkan Anda untuk mengubah rute lalu lintas (menggunakan pemeriksaan kesehatan, dikonfigurasi dengan catatan DNS untuk aplikasi Anda) saat Anda memperbarui status kontrol perutean di ARC.

Pemeriksaan kesehatan kontrol perutean

Kontrol perutean terintegrasi dengan pemeriksaan kesehatan di Route 53. Pemeriksaan kesehatan dikaitkan dengan catatan DNS yang di depan setiap replika aplikasi, misalnya,

catatan failover. Saat Anda mengubah status kontrol perutean, ARC memperbarui pemeriksaan kesehatan terkait, yang mengarahkan lalu lintas—misalnya, ke failover ke replika siaga Anda.

Panel kontrol

Panel kontrol mengelompokkan satu set kontrol routing terkait. Anda dapat mengaitkan beberapa kontrol perutean dengan satu panel kontrol, dan kemudian membuat aturan keselamatan untuk panel kontrol untuk memastikan bahwa pembaruan pengalihan lalu lintas yang Anda buat aman. Misalnya, Anda dapat mengonfigurasi kontrol perutean untuk setiap penyeimbang beban di setiap Availability Zone, lalu mengelompokkannya di panel kontrol yang sama. Kemudian Anda dapat menambahkan aturan keamanan (“aturan pernyataan”) yang memastikan bahwa setidaknya satu zona (diwakili oleh kontrol perutean) aktif pada satu waktu, untuk menghindari skenario “fail-open” yang tidak diinginkan.

Panel kontrol default

Saat Anda membuat cluster, ARC membuat panel kontrol default. Secara default, semua kontrol routing yang Anda buat di cluster ditambahkan ke panel kontrol default. Atau, Anda dapat membuat panel kontrol Anda sendiri untuk mengelompokkan kontrol perutean terkait.

Aturan keamanan

Aturan keselamatan adalah aturan yang Anda tambahkan ke kontrol perutean untuk memastikan bahwa tindakan pemulihan tidak secara tidak sengaja mengganggu ketersediaan aplikasi Anda. Misalnya, Anda dapat membuat aturan keamanan yang membuat kontrol perutean yang bertindak sebagai sakelar “on/off” secara keseluruhan sehingga Anda dapat mengaktifkan atau menonaktifkan serangkaian kontrol perutean lainnya.

Titik akhir (titik akhir cluster)

Setiap cluster di ARC memiliki lima titik akhir Regional yang dapat Anda gunakan untuk mengatur dan mengambil status kontrol perutean. Proses Anda untuk mengakses titik akhir harus mengasumsikan bahwa ARC secara teratur membawa titik akhir ke atas dan ke bawah untuk pemeliharaan, jadi Anda harus mencoba setiap titik akhir secara berurutan sampai Anda terhubung ke satu. Anda mengakses titik akhir untuk mendapatkan status kontrol perutean saat ini (Aktif atau Mati) dan untuk memicu failover untuk aplikasi Anda dengan mengubah status kontrol perutean.

Bidang data dan kontrol untuk kontrol perutean

Saat Anda merencanakan kegagalan dan pemulihan bencana, pertimbangkan seberapa tangguh mekanisme failover Anda. Kami menyarankan Anda memastikan bahwa mekanisme yang Anda andalkan selama failover sangat tersedia, sehingga Anda dapat menggunakannya saat Anda membutuhkannya dalam skenario bencana. Biasanya, Anda harus menggunakan fungsi bidang data untuk mekanisme Anda kapan pun Anda bisa, untuk keandalan dan toleransi kesalahan terbesar. Dengan mengingat hal itu, penting untuk memahami bagaimana fungsionalitas layanan dibagi antara bidang kontrol dan pesawat data, dan kapan Anda dapat mengandalkan ekspektasi keandalan ekstrim dengan bidang data layanan.

Seperti kebanyakan AWS layanan, fungsionalitas untuk kemampuan kontrol routing didukung oleh pesawat kontrol dan pesawat data. Meskipun keduanya dibangun agar dapat diandalkan, bidang kontrol dioptimalkan untuk konsistensi data, sementara bidang data dioptimalkan untuk ketersediaan. Pesawat data dirancang untuk ketahanan sehingga dapat mempertahankan ketersediaan bahkan selama peristiwa yang mengganggu, ketika pesawat kontrol mungkin menjadi tidak tersedia.

Secara umum, bidang kontrol memungkinkan Anda melakukan fungsi manajemen dasar, seperti membuat, memperbarui, dan menghapus sumber daya dalam layanan. Pesawat data menyediakan fungsionalitas inti layanan. Karena itu, kami menyarankan Anda menggunakan operasi pesawat data ketika ketersediaan penting, misalnya, ketika Anda perlu mengalihkan lalu lintas ke replika siaga selama pemadaman.

Untuk kontrol perutean, bidang kontrol dan pesawat data dibagi sebagai berikut:

- API bidang kontrol untuk kontrol perutean adalah API Konfigurasi Kontrol Pemulihan, yang didukung di Wilayah AS Barat (Oregon) (us-west-2). Anda menggunakan operasi API ini atau AWS Management Console untuk membuat atau menghapus klaster, panel kontrol, dan kontrol perutean, untuk membantu mempersiapkan peristiwa pemulihan bencana ketika Anda mungkin perlu mengubah rute lalu lintas untuk aplikasi Anda. Bidang kontrol konfigurasi kontrol perutean tidak terlalu tersedia.
- Bidang data kontrol perutean adalah cluster khusus di lima Wilayah yang terisolasi secara geografis AWS . Setiap pelanggan membuat satu atau lebih cluster menggunakan bidang kontrol kontrol routing. Cluster menghosting panel kontrol dan kontrol perutean. Kemudian Anda menggunakan Routing Control (Recovery Cluster) API untuk mendapatkan, membuat daftar, dan memperbarui status kontrol perutean saat Anda ingin mengubah rute lalu lintas untuk aplikasi Anda. Bidang data kontrol perutean sangat tersedia.

Karena bidang data kontrol perutean sangat tersedia, sebaiknya Anda berencana menggunakan panggilan API AWS Command Line Interface untuk bekerja dengan status kontrol perutean saat Anda ingin gagal memulihkan dari suatu peristiwa. Untuk informasi selengkapnya tentang pertimbangan utama saat Anda mempersiapkan dan menyelesaikan operasi pemulihan dengan kontrol perutean, lihat. Praktik terbaik untuk kontrol perutean di ARC

Untuk informasi selengkapnya tentang bidang data, pesawat kontrol, dan cara AWS membangun layanan untuk memenuhi target ketersediaan tinggi, lihat paper Stabilitas statis menggunakan Availability Zones di Amazon Builders' Library.

Penandaan untuk kontrol perutean di Amazon Application Recovery Controller (ARC)

Tag adalah kata atau frasa (meta data) yang Anda gunakan untuk mengidentifikasi dan mengatur AWS sumber daya Anda. Anda dapat menambahkan beberapa tag ke setiap sumber daya, dan setiap tag mencakup kunci dan nilai yang Anda tentukan. Misalnya, kuncinya mungkin lingkungan dan nilainya mungkin produksi. Anda dapat mencari dan memfilter sumber daya Anda berdasarkan tanda yang Anda tambahkan.

Anda dapat menandai sumber daya berikut dalam kontrol perutean di ARC:

- Klaster
- Panel kontrol
- Aturan keamanan

Penandaan di ARC hanya tersedia melalui API, misalnya, dengan menggunakan file. AWS CLI

Berikut ini adalah contoh penandaan dalam kontrol routing dengan menggunakan. AWS CLI

```
aws route53-recovery-control-config --region us-west-2 create-cluster --
cluster-name example1-cluster --tags Region=PDX,Stage=Prod
```

```
aws route53-recovery-control-config --region us-west-2 create-control-panel
--control-panel-name example1-control-panel --cluster-arn arn:aws:route53-
recovery-control::111122223333:cluster/5678abcd-abcd-5678-abcd-5678abcdefgh
--tags Region=PDX,Stage=Prod
```

Untuk informasi selengkapnya, lihat TagResourcedi Panduan Referensi API Konfigurasi Kontrol Pemulihan Pemulihan untuk Amazon Application Recovery Controller (ARC).

Harga untuk kontrol perutean di ARC

Untuk kontrol perutean di ARC, Anda membayar biaya per jam per cluster yang Anda buat. Setiap cluster dapat meng-host beberapa kontrol routing, yang Anda gunakan untuk memicu failover aplikasi.

Untuk membantu mengelola biaya dan meningkatkan efisiensi, Anda dapat mengatur berbagi lintas akun untuk sebuah klaster, untuk berbagi satu klaster dengan beberapa AWS akun. Untuk informasi selengkapnya, lihat Support cross-account untuk cluster di ARC.

Untuk informasi harga terperinci untuk ARC dan contoh harga, lihat Harga ARC.

Memulai pemulihan Multi-wilayah di Amazon Application Recovery Controller (ARC)

Untuk gagal atas aplikasi Anda dengan menggunakan kontrol routing di Amazon Application Recovery Controller (ARC), Anda harus memiliki AWS aplikasi yang dalam beberapa Wilayah AWS. Untuk memulai, pertama, pastikan bahwa aplikasi Anda diatur dalam replika silo di setiap Wilayah, sehingga Anda dapat gagal dari satu ke yang lain selama acara. Kemudian, Anda dapat membuat kontrol perutean untuk mengalihkan lalu lintas aplikasi agar gagal dari aplikasi utama ke aplikasi sekunder, menjaga kontinuitas bagi pengguna Anda.

> **Note**
>
> Jika Anda memiliki aplikasi yang dibungkam oleh Availability Zones, pertimbangkan untuk menggunakan zonal shift atau zonal autoshift untuk pemulihan failover. Tidak diperlukan pengaturan untuk menggunakan pergeseran zona atau pergeseran otomatis zona untuk memulihkan aplikasi dengan andal dari gangguan Availability Zone. Untuk informasi selengkapnya, lihat Gunakan zonal shift dan zonal autoshift untuk memulihkan aplikasi di ARC.

Agar Anda dapat menggunakan kontrol perutean ARC untuk memulihkan aplikasi selama acara, kami sarankan Anda mengatur setidaknya dua aplikasi yang merupakan replika satu sama lain. Setiap replika, atau sel, mewakili Wilayah AWS. Setelah menyiapkan sumber daya aplikasi agar selaras dengan Wilayah, pastikan aplikasi Anda disiapkan untuk pemulihan yang berhasil dengan mengambil langkah-langkah berikut.

Tip: Untuk membantu menyederhanakan penyiapan, kami menyediakan AWS CloudFormation dan templat HashiCorp Terraform yang membuat aplikasi dengan replika redundan yang gagal secara independen satu sama lain. Untuk mempelajari lebih lanjut dan mengunduh templat, lihatMenyiapkan aplikasi contoh.

Untuk mempersiapkan penggunaan kontrol perutean, pastikan aplikasi Anda diatur agar tangguh dengan melakukan hal berikut:

1. Buat salinan independen dari tumpukan aplikasi Anda (jaringan dan lapisan komputasi) yang merupakan replika satu sama lain di setiap Wilayah sehingga Anda dapat gagal melewati lalu lintas dari satu ke yang lain ketika ada acara. Pastikan Anda tidak memiliki dependensi lintas wilayah dalam kode aplikasi Anda yang akan menyebabkan kegagalan satu replika memengaruhi yang lain. Agar berhasil gagal di antaranya Wilayah AWS, batas tumpukan Anda harus berada dalam Wilayah.
2. Gandakan semua data stateful yang diperlukan untuk aplikasi Anda di seluruh replika. Anda dapat menggunakan layanan AWS database untuk membantu mereplikasi data Anda.

Memulai dengan kontrol perutean untuk failover lalu lintas

Kontrol perutean di Amazon Application Recovery Controller (ARC) memungkinkan Anda memicu failover agar lalu lintas Anda gagal di antara salinan aplikasi yang berlebihan, atau replika, yang berjalan secara terpisah. Wilayah AWS Failover dilakukan dengan DNS, menggunakan bidang data Amazon Route 53.

Setelah Anda mengatur replika Anda di setiap Wilayah, seperti yang dijelaskan di bagian berikutnya, Anda dapat mengaitkan masing-masing dengan kontrol perutean. Pertama, Anda mengaitkan kontrol perutean dengan nama domain tingkat atas replika Anda di setiap Wilayah. Kemudian, Anda menambahkan pemeriksaan kesehatan kontrol perutean ke kontrol perutean sehingga dapat mengaktifkan dan mematikan arus lalu lintas. Ini memungkinkan Anda untuk mengontrol perutean lalu lintas di seluruh replika aplikasi Anda.

Anda dapat memperbarui status kontrol perutean di dalam AWS Management Console agar gagal melewati lalu lintas, tetapi sebaiknya Anda menggunakan tindakan ARC, menggunakan API atau AWS CLI, untuk mengubahnya. Tindakan API tidak bergantung pada konsol, jadi tindakan tersebut lebih tangguh.

Misalnya, untuk gagal di antara Wilayah, dari us-west-1 ke us-east-1, Anda dapat `update-routing-control-state` menggunakan tindakan API untuk menyetel status ke dan ke. `us-west-1` `Off` `us-east-1` `On`

Sebelum Anda membuat komponen kontrol routing untuk mengatur failover untuk aplikasi Anda, pastikan bahwa aplikasi Anda di-siloed ke replika Regional, sehingga Anda dapat gagal dari satu ke yang lain. Untuk mempelajari lebih lanjut dan mulai membungkam aplikasi baru atau membuat tumpukan contoh, lihat bagian berikutnya.

Menyiapkan aplikasi contoh

Untuk membantu Anda memahami cara kerja kontrol perutean, kami menyediakan contoh aplikasi yang disebut`TicTacToe`. Contoh menggunakan AWS CloudFormation template untuk menyederhanakan proses, serta AWS CloudFormation template yang dapat diunduh sehingga Anda dapat dengan cepat menjelajahi pengaturan dan menggunakan ARC sendiri.

Setelah menerapkan aplikasi sampel, Anda dapat menggunakan templat untuk membuat komponen ARC, lalu menjelajah menggunakan kontrol perutean untuk mengelola arus lalu lintas ke aplikasi. Anda dapat menyesuaikan template dan proses untuk skenario dan aplikasi Anda sendiri.

Untuk memulai dengan contoh aplikasi dan AWS CloudFormation template, lihat instruksi README di GitHubrepo ARC. Anda dapat mempelajari lebih lanjut tentang menggunakan AWS CloudFormation templat dengan membaca AWS CloudFormation konsep di Panduan AWS CloudFormation Pengguna.

Praktik terbaik untuk kontrol perutean di ARC

Kami merekomendasikan praktik terbaik berikut untuk pemulihan dan kesiapan failover untuk kontrol perutean di ARC.

Topik

- Jaga agar AWS kredensil yang dibangun khusus dan berumur panjang tetap aman dan selalu dapat diakses
- Pilih nilai TTL yang lebih rendah untuk catatan DNS yang terlibat dalam failover
- Batasi waktu klien tetap terhubung ke titik akhir Anda
- Tandai atau kode keras lima titik akhir cluster Regional dan kontrol perutean Anda ARNs
- Pilih salah satu titik akhir Anda secara acak untuk memperbarui status kontrol perutean Anda

- Gunakan API bidang data yang sangat andal untuk membuat daftar dan memperbarui status kontrol perutean, bukan konsol

Jaga agar AWS kredensil yang dibangun khusus dan berumur panjang tetap aman dan selalu dapat diakses

Dalam skenario pemulihan bencana (DR), pertahankan ketergantungan sistem seminimal mungkin dengan menggunakan pendekatan sederhana untuk mengakses AWS dan melakukan tugas pemulihan. Buat kredensil IAM yang berumur panjang khusus untuk tugas DR, dan simpan kredensialnya dengan aman di brankas fisik lokal atau brankas virtual, untuk diakses bila diperlukan. Dengan IAM, Anda dapat mengelola kredenal keamanan secara terpusat, seperti kunci akses, dan izin untuk akses ke sumber daya. AWS Untuk tugas non-DR, kami menyarankan Anda untuk terus menggunakan akses federasi, menggunakan AWS layanan seperti AWS Single Sign-On.

Untuk melakukan tugas failover di ARC dengan API bidang data cluster pemulihan, Anda dapat melampirkan kebijakan ARC IAM ke pengguna Anda. Untuk mempelajari selengkapnya, lihat Contoh kebijakan berbasis identitas di Amazon Application Recovery Controller (ARC).

Pilih nilai TTL yang lebih rendah untuk catatan DNS yang terlibat dalam failover

Untuk catatan DNS yang mungkin perlu Anda ubah sebagai bagian dari mekanisme failover Anda, terutama catatan yang diperiksa kesehatan, menggunakan nilai TTL yang lebih rendah adalah tepat. Mengatur TTL 60 atau 120 detik adalah pilihan umum untuk skenario ini.

Pengaturan DNS TTL (time to live) memberi tahu resolver DNS berapa lama untuk menyimpan rekaman sebelum meminta yang baru. Ketika Anda memilih TTL, Anda membuat trade-off antara latensi dan keandalan, dan responsif terhadap perubahan. Dengan TTL yang lebih pendek pada catatan, penyelesai DNS melihat pembaruan ke catatan lebih cepat karena TTL menentukan bahwa mereka harus melakukan kueri lebih sering.

Untuk informasi selengkapnya, lihat Memilih nilai TTL untuk catatan DNS dalam Praktik terbaik untuk Amazon Route 53 DNS.

Batasi waktu klien tetap terhubung ke titik akhir Anda

Saat Anda menggunakan kontrol perutean untuk berpindah dari satu Wilayah AWS ke yang lain, mekanisme yang digunakan Amazon Application Recovery Controller (ARC) untuk memindahkan lalu lintas aplikasi Anda adalah pembaruan DNS. Pembaruan ini menyebabkan semua koneksi baru diarahkan menjauh dari lokasi yang rusak.

Namun, klien dengan koneksi terbuka yang sudah ada sebelumnya mungkin terus membuat permintaan terhadap lokasi yang rusak sampai klien terhubung kembali. Untuk memastikan pemulihan yang cepat, kami sarankan Anda membatasi jumlah waktu klien tetap terhubung ke titik akhir Anda.

Jika Anda menggunakan Application Load Balancer, Anda dapat menggunakan `keepalive` opsi untuk mengonfigurasi berapa lama koneksi berlanjut. Untuk informasi selengkapnya, lihat durasi keepalive klien HTTP di Panduan Pengguna Application Load Balancer.

Secara default, Application Load Balancers menetapkan nilai durasi keepalive klien HTTP menjadi 3600 detik, atau 1 jam. Kami menyarankan agar Anda menurunkan nilai agar sesuai dengan sasaran waktu pemulihan untuk aplikasi Anda, misalnya, 300 detik. Saat Anda memilih waktu durasi keepalive klien HTTP, pertimbangkan bahwa nilai ini adalah pertukaran antara menghubungkan kembali lebih sering secara umum, yang dapat memengaruhi latensi, dan lebih cepat memindahkan semua klien dari AZ atau Wilayah yang terganggu.

Tandai atau kode keras lima titik akhir cluster Regional dan kontrol perutean Anda ARNs

Kami menyarankan Anda menyimpan salinan lokal titik akhir cluster ARC Regional Anda, di bookmark atau disimpan dalam kode otomatisasi yang Anda gunakan untuk mencoba kembali titik akhir Anda. Selama peristiwa kegagalan, Anda mungkin tidak dapat mengakses beberapa operasi API, termasuk operasi ARC API yang tidak dihosting di cluster bidang data yang sangat andal. Anda dapat membuat daftar titik akhir untuk kluster ARC Anda dengan menggunakan operasi DescribeClusterAPI.

Pilih salah satu titik akhir Anda secara acak untuk memperbarui status kontrol perutean Anda

Kontrol perutean menyediakan lima titik akhir Regional untuk memastikan ketersediaan tinggi, bahkan ketika menghadapi kegagalan. Untuk mencapai ketahanan penuh mereka, penting untuk memiliki logika coba lagi yang dapat menggunakan kelima titik akhir yang diperlukan. Untuk informasi tentang menggunakan contoh kode dengan AWS SDK, termasuk contoh untuk mencoba titik akhir klaster, lihat. Contoh kode untuk Application Recovery Controller menggunakan AWS SDKs

Gunakan API bidang data yang sangat andal untuk membuat daftar dan memperbarui status kontrol perutean, bukan konsol

Menggunakan API bidang data ARC, lihat kontrol dan status perutean Anda dengan ListRoutingControlsoperasi dan perbarui status kontrol perutean untuk mengarahkan lalu lintas untuk failover dengan operasi. UpdateRoutingControlState Anda dapat menggunakan AWS CLI (seperti dalam contoh ini) atau kode yang Anda tulis menggunakan salah satu AWS SDKs.

ARC menawarkan keandalan ekstrim dengan API di bidang data untuk gagal selama lalu lintas. Sebaiknya gunakan API alih-alih mengubah status kontrol perutean di. AWS Management Console

Connect ke salah satu endpoint kluster Regional Anda agar ARC dapat menggunakan API bidang data. Jika titik akhir tidak tersedia, coba sambungkan ke titik akhir cluster lain.

Jika aturan keselamatan memblokir pembaruan status kontrol perutean, Anda dapat memotongnya untuk membuat pembaruan dan gagal atas lalu lintas. Untuk informasi selengkapnya, lihat Mengesampingkan aturan keselamatan untuk mengubah rute lalu lintas.

Uji failover dengan ARC

Uji failover secara teratur dengan kontrol perutean ARC, untuk gagal dari tumpukan aplikasi utama Anda ke tumpukan aplikasi sekunder. Penting untuk memastikan bahwa struktur ARC yang telah Anda tambahkan selaras dengan sumber daya yang benar di tumpukan Anda, dan semuanya berfungsi seperti yang Anda harapkan. Anda harus menguji ini setelah Anda mengatur ARC untuk lingkungan Anda, dan terus menguji secara berkala, sehingga lingkungan failover Anda siap, sebelum Anda mengalami situasi kegagalan di mana Anda memerlukan sistem sekunder Anda untuk aktif dan berjalan cepat untuk menghindari downtime bagi pengguna Anda.

Operasi API kontrol perutean

Bagian ini mencakup tabel dengan daftar operasi API yang dapat Anda gunakan untuk menyiapkan dan menggunakan kontrol perutean di Amazon Application Recovery Controller (ARC), dengan tautan ke dokumentasi yang relevan.

Untuk contoh cara menggunakan operasi API konfigurasi kontrol perutean umum dengan AWS Command Line Interface, lihat Contoh penggunaan operasi API kontrol perutean ARC dengan AWS CLI.

Tabel berikut mencantumkan operasi ARC API yang dapat Anda gunakan untuk konfigurasi kontrol perutean, dengan tautan ke dokumentasi yang relevan.

Tindakan	Menggunakan konsol ARC	Menggunakan ARC API
Membuat klaster	Lihat Membuat komponen kontrol perutean di ARC	Lihat CreateCluster

Tindakan	Menggunakan konsol ARC	Menggunakan ARC API
Jelaskan sebuah cluster	Lihat Membuat komponen kontrol perutean di ARC	Lihat DescribeCluster
Hapus klaster	Lihat Membuat komponen kontrol perutean di ARC	Lihat DeleteCluster
Daftar klaster untuk akun	Lihat Membuat komponen kontrol perutean di ARC	Lihat ListClusters
Buat kontrol perutean	Lihat Membuat komponen kontrol perutean di ARC	Lihat CreateRoutingControl
Jelaskan kontrol perutean	Lihat Membuat komponen kontrol perutean di ARC	Lihat DescribeRoutingControl
Perbarui kontrol perutean	Lihat Membuat komponen kontrol perutean di ARC	Lihat UpdateRoutingControl
Hapus kontrol perutean	Lihat Membuat komponen kontrol perutean di ARC	Lihat DeleteRoutingControl
Daftar kontrol perutean	Lihat Membuat komponen kontrol perutean di ARC	Lihat ListRoutingControls
Buat panel kontrol	Lihat Membuat komponen kontrol perutean di ARC	Lihat CreateControlPanel
Jelaskan panel kontrol	Lihat Membuat komponen kontrol perutean di ARC	Lihat DescribeControlPanel
Perbarui panel kontrol	Lihat Membuat komponen kontrol perutean di ARC	Lihat UpdateControlPanel
Hapus panel kontrol	Lihat Membuat komponen kontrol perutean di ARC	Lihat DeleteControlPanel
Daftar panel kontrol	Lihat Membuat komponen kontrol perutean di ARC	Lihat ListControlPanels

Tindakan	Menggunakan konsol ARC	Menggunakan ARC API
Buat aturan keamanan	Lihat Membuat aturan keselamatan untuk kontrol perutean	Lihat CreateSafetyRule
Jelaskan aturan keamanan	Lihat Membuat aturan keselamatan untuk kontrol perutean	Lihat DescribeSafetyRule
Perbarui aturan keamanan	Lihat Membuat aturan keselamatan untuk kontrol perutean	Lihat UpdateSafetyRule
Hapus aturan keamanan	Lihat Membuat aturan keselamatan untuk kontrol perutean	Lihat DeleteSafetyRule
Daftar aturan keselamatan	Lihat Membuat aturan keselamatan untuk kontrol perutean	Lihat ListSafetyRules
Daftar terkait pemeriksaan kesehatan Route 53	Lihat Membuat pemeriksaan kesehatan kontrol perutean di ARC	Lihat ListAssociatedRoute53HealthChecks
Buat daftar kebijakan AWS RAM sumber daya untuk berbagi klaster	Lihat Support cross-account untuk cluster di ARC	Lihat GetResourcePolicy

Tabel berikut mencantumkan operasi ARC API umum yang dapat Anda gunakan untuk mengelola failover lalu lintas dengan bidang data kontrol perutean, dengan tautan ke dokumentasi yang relevan.

Tindakan	Menggunakan konsol ARC	Menggunakan ARC API
Dapatkan status kontrol perutean	Lihat Mendapatkan dan memperbarui status kontrol	Lihat GetRoutingControlState

Tindakan	Menggunakan konsol ARC	Menggunakan ARC API
	perutean di AWS Management Console	
Daftar kontrol perutean	N/A	Lihat ListRoutingControls
Perbarui status kontrol perutean	Lihat Mendapatkan dan memperbarui status kontrol perutean di AWS Management Console	Lihat UpdateRoutingControlState
Perbarui beberapa status kontrol perutean	Lihat Mendapatkan dan memperbarui status kontrol perutean di AWS Management Console	Lihat UpdateRoutingControlStates

Menggunakan layanan ini dengan AWS SDK

AWS kit pengembangan perangkat lunak (SDKs) tersedia untuk banyak bahasa pemrograman populer. Setiap SDK menyediakan API, contoh kode, dan dokumentasi yang memudahkan developer untuk membangun aplikasi dalam bahasa pilihan mereka.

Dokumentasi SDK	Contoh kode
AWS SDK untuk C++	AWS SDK untuk C++ contoh kode
AWS CLI	AWS CLI contoh kode
AWS SDK untuk Go	AWS SDK untuk Go contoh kode
AWS SDK untuk Java	AWS SDK untuk Java contoh kode
AWS SDK untuk JavaScript	AWS SDK untuk JavaScript contoh kode
AWS SDK untuk Kotlin	AWS SDK untuk Kotlin contoh kode
AWS SDK untuk .NET	AWS SDK untuk .NET contoh kode

Dokumentasi SDK	Contoh kode
AWS SDK untuk PHP	AWS SDK untuk PHP contoh kode
Alat AWS untuk PowerShell	Alat AWS untuk PowerShell contoh kode
AWS SDK untuk Python (Boto3)	AWS SDK untuk Python (Boto3) contoh kode
AWS SDK untuk Ruby	AWS SDK untuk Ruby contoh kode
AWS SDK for Rust	AWS SDK for Rust contoh kode
AWS SDK untuk SAP ABAP	AWS SDK untuk SAP ABAP contoh kode
AWS SDK for Swift	AWS SDK for Swift contoh kode

Untuk contoh khusus untuk layanan ini, lihat Contoh kode untuk Application Recovery Controller menggunakan AWS SDKs.

Ketersediaan contoh

Tidak dapat menemukan apa yang Anda butuhkan? Minta contoh kode menggunakan tautan Berikan umpan balik di bagian bawah halaman ini.

Contoh penggunaan operasi API kontrol perutean ARC dengan AWS CLI

Bagian ini berjalan melalui contoh aplikasi sederhana bekerja dengan kontrol routing, menggunakan AWS Command Line Interface untuk bekerja dengan kemampuan kontrol routing di Amazon Application Recovery Controller (ARC) menggunakan operasi API. Contohnya dimaksudkan untuk membantu Anda mengembangkan pemahaman dasar tentang cara bekerja dengan kontrol perutean menggunakan CLI.

Dengan kontrol perutean di Amazon Application Recovery Controller (ARC), Anda dapat memicu kegagalan lalu lintas antara salinan aplikasi redundan, atau replika, yang berjalan di zona terpisah atau Availability Zones. Wilayah AWS

Anda mengatur kontrol perutean ke dalam grup yang disebut panel kontrol yang disediakan di klaster. Cluster ARC adalah seperangkat titik akhir Regional yang digunakan secara global. Titik akhir klaster

menyediakan API yang sangat tersedia yang dapat Anda gunakan untuk mengatur dan mengambil status kontrol perutean. Untuk informasi selengkapnya tentang komponen fitur kontrol perutean, lihatKomponen kontrol perutean .

> **Note**
>
> ARC adalah layanan global yang mendukung titik akhir dalam beberapa Wilayah AWS. Namun, Anda harus menentukan wilayah AS Barat (Oregon) — yaitu, tentukan `--region us-west-2` parameternya — di sebagian besar perintah ARC CLI. Misalnya, gunakan `region` parameter saat Anda membuat grup pemulihan, panel kontrol, dan cluster.
> Saat Anda membuat klaster, ARC memberi Anda satu set titik akhir Regional. Untuk mendapatkan atau memperbarui status kontrol perutean, Anda harus menentukan titik akhir Regional (URL Wilayah AWS dan titik akhir) dalam perintah CLI Anda.

Untuk informasi selengkapnya tentang penggunaan AWS CLI, lihat Referensi AWS CLI Perintah. Untuk daftar tindakan API kontrol perutean, lihat Operasi API kontrol perutean dan Operasi API kontrol perutean.

Kita akan mulai dengan membuat komponen yang Anda butuhkan untuk mengelola failover dengan menggunakan kontrol routing, dimulai dengan membuat cluster.

Siapkan komponen kontrol perutean

Langkah pertama kami adalah membuat cluster. Cluster ARC adalah satu set lima titik akhir, satu di masing-masing dari lima titik berbeda Wilayah AWS. Infrastruktur ARC mendukung titik akhir ini untuk bekerja dalam koordinasi sehingga mereka menjamin ketersediaan tinggi dan konsistensi berurutan operasi failover.

1. Membuat klaster

1a. Buat sebuah klaster. `network-type`Ini opsional, dan bisa jadi `IPV4` atau`DUALSTACK`. Nilai default-nya `IPV4`.

```
aws route53-recovery-control-config create-cluster --cluster-name test --network-type
 DUALSTACK
```

```
"Cluster": {
```

```
    "ClusterArn": "arn:aws:route53-recovery-
control::123456789123:cluster/12341234-1234-1234-1234-123412341234",
    "Name": "test",
    "Status": "PENDING",
    "Owner": "123456789123",
    "NetworkType": "DUALSTACK"
}
```

Saat pertama kali membuat sumber daya ARC, ia memiliki status PENDING saat cluster dibuat. Anda dapat memeriksa kemajuannya dengan menelepondescribe-cluster.

1b. Jelaskan sebuah cluster.

```
aws route53-recovery-control-config --region us-west-2 \
    describe-cluster --cluster-arn arn:aws:route53-recovery-
control::111122223333:cluster/5678abcd-abcd-5678-abcd-5678abcdefgh
```

```
"Cluster": {
    "ClusterArn": "arn:aws:route53-recovery-
control::123456789123:cluster/12341234-1234-1234-1234-123412341234",
    "Name": "test",
    "Status": "DEPLOYED",
    "Owner": "123456789123",
    "NetworkType": "DUALSTACK"
}
```

Saat status DEPLOYED, ARC telah berhasil membuat klaster dengan kumpulan titik akhir untuk berinteraksi dengan Anda. Anda dapat membuat daftar semua cluster Anda dengan meneleponlist-clusters.

1c. Buat daftar cluster Anda.

```
aws route53-recovery-control-config --region us-west-2 list-clusters
```

```
"Cluster": {
    "ClusterArn": "arn:aws:route53-recovery-
control::123456789123:cluster/12341234-1234-1234-1234-123412341234",
    "Name": "test",
    "Status": "DEPLOYED",
    "Owner": "123456789123",
    "NetworkType": "DUALSTACK"
```

```
}
```

1d. Perbarui jenis jaringan untuk cluster Anda. Pilihannya adalah `IPV4` atau`DUALSTACK`.

```
aws route53-recovery-control-config update-cluster \
--cluster-arn arn:aws:route53-recovery-
control::123456789123:cluster/12341234-1234-1234-1234-123412341234 \
--network-type DUALSTACK
```

```
"Cluster": {
    "ClusterArn": "arn:aws:route53-recovery-
control::123456789123:cluster/12341234-1234-1234-1234-123412341234",
    "Name": "test",
    "Status": "PENDING",
    "Owner": "123456789123",
   "NetworkType": "DUALSTACK"
}
```

2. Buat panel kontrol

Panel kontrol adalah pengelompokan logis untuk mengatur kontrol routing ARC Anda.
Saat Anda membuat cluster, ARC secara otomatis menyediakan panel kontrol untuk Anda dipanggil`DefaultControlPanel`. Anda dapat menggunakan panel kontrol ini segera.

Panel kontrol hanya bisa ada dalam satu cluster. Jika Anda ingin memindahkan panel kontrol ke cluster lain, Anda harus menghapusnya dan kemudian membuatnya di cluster kedua. Anda dapat melihat semua panel kontrol di akun Anda dengan meneleponlist-control-panels. Untuk melihat hanya panel kontrol di cluster tertentu, tambahkan `--cluster-arn` bidang.

2a. Daftar panel kontrol.

```
aws route53-recovery-control-config --region us-west-2 \
    list-control-panels --cluster-arn arn:aws:route53-recovery-
control::111122223333:cluster/eba23304-1a51-4674-ae32-b4cf06070bdd
```

```
{
    "ControlPanels": [
        {
            "ControlPanelArn": "arn:aws:route53-recovery-
control::111122223333:controlpanel/1234567dddddd1234567dddddd1234567",
```

```
            "ClusterArn": "arn:aws:route53-recovery-
control::111122223333:cluster/5678abcd-abcd-5678-abcd-5678abcdefgh",
            "DefaultControlPanel": true,
            "Name": "DefaultControlPanel",
            "RoutingControlCount": 0,
            "Status": "DEPLOYED"
        }
    ]
}
```

Secara opsional, buat panel kontrol Anda sendiri dengan meneleponcreate-control-panel.

2b. Buat panel kontrol.

```
aws route53-recovery-control-config --region us-west-2 create-control-panel \
        --control-panel-name NewControlPanel2 \
        --cluster-arn arn:aws:route53-recovery-control::111122223333:cluster/5678abcd-
abcd-5678-abcd-5678abcdefgh
```

```
{
    "ControlPanel": {
        "ControlPanelArn": "arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456",
        "ClusterArn": "arn:aws:route53-recovery-control::111122223333:cluster/5678abcd-
abcd-5678-abcd-5678abcdefgh",
        "DefaultControlPanel": false,
        "Name": "NewControlPanel2",
        "RoutingControlCount": 0,
        "Status": "PENDING"
    }
}
```

Saat pertama kali membuat sumber daya ARC, ia memiliki status PENDING saat sedang dibuat. Anda dapat memeriksa kemajuan dengan menelepondescribe-control-panel.

2c. Jelaskan panel kontrol.

```
aws route53-recovery-control-config --region us-west-2 describe-control-panel \
    --control-panel-arn arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456
```

```
{
```

```
    "ControlPanel": {
        "ControlPanelArn": "arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456",
        "ClusterArn": "arn:aws:route53-recovery-control::111122223333:cluster/5678abcd-
abcd-5678-abcd-5678abcdefgh",
        "DefaultControlPanel": true,
        "Name": "DefaultControlPanel",
        "RoutingControlCount": 0,
        "Status": "DEPLOYED"
    }
}
```

3. Buat kontrol perutean

Sekarang setelah Anda mengatur cluster dan melihat panel kontrol, Anda dapat mulai membuat kontrol perutean. Saat Anda membuat kontrol perutean, Anda setidaknya harus menentukan Nama Sumber Daya Amazon (ARN) dari cluster tempat Anda ingin kontrol perutean berada. Anda juga dapat menentukan ARN dari panel kontrol untuk kontrol routing. Anda juga harus menentukan cluster tempat panel kontrol berada.

Jika Anda tidak menentukan panel kontrol, kontrol perutean Anda ditambahkan ke panel kontrol yang dibuat secara otomatis. `DefaultControlPanel`

Buat kontrol perutean dengan meneleponcreate-routing-control.

3a. Buat kontrol routing.

```
aws route53-recovery-control-config --region us-west-2 create-routing-control \
    --routing-control-name NewRc1 \
    --cluster-arn arn:aws:route53-recovery-control::111122223333:cluster/5678abcd-
abcd-5678-abcd-5678abcdefgh
```

```
{
    "RoutingControl": {
        "ControlPanelArn": " arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456",
        "Name": "NewRc1",
        "RoutingControlArn": "arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/routingcontrol/
abcdefg1234567",
        "Status": "PENDING"
```

```
    }
}
```

Kontrol perutean mengikuti pola pembuatan yang sama dengan sumber daya ARC lainnya, sehingga Anda dapat melacak kemajuannya dengan memanggil operasi describe.

3b. Jelaskan kontrol perutean.

```
aws route53-recovery-control-config --region us-west-2 describe-routing-control \
       --routing-control-arn arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/routingcontrol/
abcdefg1234567
```

```
{
    "RoutingControl": {
        "ControlPanelArn": "arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456",
        "Name": "NewRc1",
        "RoutingControlArn": "arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/routingcontrol/
abcdefg1234567",
        "Status": "DEPLOYED"
    }
}
```

Anda dapat membuat daftar kontrol perutean di panel kontrol dengan meneleponlist-routing-controls. Panel kontrol ARN diperlukan.

3c. Daftar kontrol perutean.

```
aws route53-recovery-control-config --region us-west-2 list-routing-controls \
        --control-panel-arn arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456
```

```
{
    "RoutingControls": [
        {
            "ControlPanelArn": "arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456",
            "Name": "Rc1",
```

```
            "RoutingControlArn": "arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/routingcontrol/
abcdefg1234567",
            "Status": "DEPLOYED"
        },
        {
            "ControlPanelArn": "arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456",
            "Name": "Rc2",
            "RoutingControlArn": "arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/routingcontrol/
hijklmnop987654321",
            "Status": "DEPLOYED"
        }
    ]
}
```

Dalam contoh berikut, di mana kami bekerja dengan status kontrol perutean, kami berasumsi bahwa Anda memiliki dua kontrol perutean yang tercantum di bagian ini (Rc1 dan Rc2). Dalam contoh ini, setiap kontrol perutean mewakili Availability Zone tempat aplikasi Anda digunakan.

4. Buat aturan keselamatan

Ketika Anda bekerja dengan beberapa kontrol routing pada saat yang sama, Anda mungkin memutuskan bahwa Anda ingin beberapa perlindungan di tempat ketika Anda mengaktifkan dan menonaktifkannya, untuk menghindari konsekuensi yang tidak disengaja, seperti mematikan kedua kontrol routing dan menghentikan semua arus lalu lintas. Untuk membuat perlindungan ini, Anda membuat aturan keselamatan kontrol perutean.

Ada dua jenis aturan keselamatan: aturan asertion dan aturan gating. Untuk mempelajari lebih lanjut tentang aturan keselamatan, lihatMembuat aturan keselamatan untuk kontrol perutean .

Panggilan berikut memberikan contoh pembuatan aturan pernyataan yang memastikan bahwa setidaknya satu dari dua kontrol routing diatur 0n pada waktu tertentu. Untuk membuat aturan, Anda menjalankan `create-safety-rule` dengan `assertion-rule` parameter.

Untuk informasi terperinci tentang operasi API aturan pernyataan, lihat AssertionRuledi Panduan Referensi API Kontrol Perutean untuk Pengontrol Pemulihan Aplikasi Amazon.

4a. Buat aturan pernyataan.

```
aws route53-recovery-control-config --region us-west-2 create-safety-rule \
```

```
        --assertion-rule '{"Name": "TestAssertionRule",
        "ControlPanelArn": "arn:aws:route53-recovery-
control::888888888888:controlpanel/zzz123yyy456xxx789zzz123yyy456xxx",
        "WaitPeriodMs": 5000,
        "AssertedControls":
        ["arn:aws:route53-recovery-control::888888888888:controlpanel/
zzz123yyy456xxx789zzz123yyy456xxx/routingcontrol/def123def123def"
        "arn:aws:route53-recovery-control::888888888888:controlpanel/
zzz123yyy456xxx789zzz123yyy456xxx/routingcontrol/ghi456ghi456ghi"],
        "RuleConfig": {"Threshold": 1, "Type": "ATLEAST", "Inverted": false}}'
```

```
{
    "Rule": {
        "ASSERTION": {
            "Arn": "arn:aws:route53-recovery-control::888888888888:controlpanel/
zzz123yyy456xxx789zzz123yyy456xxx/safetyrule/333333444444",
            "AssertedControls": [
                "arn:aws:route53-recovery-control::888888888888:controlpanel/
zzz123yyy456xxx789zzz123yyy456xxx/routingcontrol/def123def123def"
                "arn:aws:route53-recovery-control::888888888888:controlpanel/
zzz123yyy456xxx789zzz123yyy456xxx/routingcontrol/ghi456ghi456ghi"],
            "ControlPanelArn": "arn:aws:route53-recovery-
control::888888888888:controlpanel/zzz123yyy456xxx789zzz123yyy456xxx",
            "Name": "TestAssertionRule",
            "RuleConfig": {
                "Inverted": false,
                "Threshold": 1,
                "Type": "ATLEAST"
            },
            "Status": "PENDING",
            "WaitPeriodMs": 5000
        }
    }
}
```

Panggilan berikut memberikan contoh pembuatan aturan gating yang menyediakan sakelar “on/off” atau “gating” keseluruhan untuk satu set kontrol perutean target di panel kontrol. Ini memungkinkan Anda melarang memperbarui kontrol perutean target sehingga, misalnya, otomatisasi tidak dapat membuat pembaruan yang tidak sah. Dalam contoh ini, sakelar gating adalah kontrol perutean yang ditentukan oleh `GatingControls` parameter dan dua kontrol perutean yang dikendalikan atau “terjaga” ditentukan oleh parameter. `TargetControls`

> **Note**
>
> Sebelum Anda membuat aturan gating, Anda harus membuat kontrol routing gating, yang tidak menyertakan catatan failover DNS, dan kontrol perutean target, yang Anda konfigurasikan dengan catatan failover DNS.

Untuk membuat aturan, Anda menjalankan `create-safety-rule` dengan `gating-rule` parameter.

Untuk informasi terperinci tentang operasi API aturan pernyataan, lihat GatingRuledi Panduan Referensi API Kontrol Perutean untuk Pengontrol Pemulihan Aplikasi Amazon.

4b. Buat aturan gating.

```
aws route53-recovery-control-config --region us-west-2 create-safety-rule \
        --gating-rule '{"Name": "TestGatingRule",
        "ControlPanelArn": "arn:aws:route53-recovery-
control::888888888888:controlpanel/zzz123yyy456xxx789zzz123yyy456xxx",
        "WaitPeriodMs": 5000,
        "GatingControls": ["arn:aws:route53-recovery-
control::888888888888:controlpanel/zzz123yyy456xxx789zzz123yyy456xxx/routingcontrol/
def123def123def"]
        "TargetControls": ["arn:aws:route53-recovery-
control::888888888888:controlpanel/zzz123yyy456xxx789zzz123yyy456xxx/routingcontrol/
ghi456ghi456ghi",
        "arn:aws:route53-recovery-control::888888888888:controlpanel/
zzz123yyy456xxx789zzz123yyy456xxx/routingcontrol/lmn789lmn789lmn"],
        "RuleConfig": {"Threshold": 0, "Type": "OR", "Inverted": false}}'
```

```
{
    "Rule": {
        "GATING": {
            "Arn": "arn:aws:route53-recovery-control::888888888888:controlpanel/
zzz123yyy456xxx789zzz123yyy456xxx/safetyrule/444444444444",
            "GatingControls": [
                "arn:aws:route53-recovery-control::888888888888:controlpanel/
zzz123yyy456xxx789zzz123yyy456xxx/routingcontrol/def123def123def"
            ],
            "TargetControls": [
                "arn:aws:route53-recovery-control::888888888888:controlpanel/
zzz123yyy456xxx789zzz123yyy456xxx/routingcontrol/ghi456ghi456ghi"
```

```
                "arn:aws:route53-recovery-control::888888888888:controlpanel/
zzz123yyy456xxx789zzz123yyy456xxx/routingcontrol/lmn789lmn789lmn"
            ],
            "ControlPanelArn": "arn:aws:route53-recovery-
control::888888888888:controlpanel/zzz123yyy456xxx789zzz123yyy456xxx",
            "Name": "TestGatingRule",
            "RuleConfig": {
                "Inverted": false,
                "Threshold": 0,
                "Type": "OR"
            },
            "Status": "PENDING",
            "WaitPeriodMs": 5000
        }
    }
}
```

Seperti sumber daya kontrol perutean lainnya, Anda dapat menjelaskan, membuat daftar, atau menghapus aturan keselamatan setelah disebarkan ke bidang data.

Setelah menyiapkan satu atau beberapa aturan keselamatan, Anda dapat terus berinteraksi dengan klaster, mengatur, atau mengambil status untuk kontrol perutean. Jika `set-routing-control-state` operasi melanggar aturan yang Anda buat, Anda akan menerima pengecualian yang mirip dengan berikut ini:

```
Cannot modify control state for [0123456bbbbbbb0123456bbbbbb01234560123
abcdefg1234567] due to failed rule evaluation
0123456bbbbbbb0123456bbbbbb01234563333333444444
```

Identifier pertama adalah panel kontrol ARN digabungkan dengan kontrol routing ARN. Pengenal kedua adalah panel kontrol ARN digabungkan dengan aturan keselamatan ARN.

5. Buat pemeriksaan kesehatan

Untuk menggunakan kontrol perutean agar gagal dalam lalu lintas, Anda membuat pemeriksaan kesehatan di Amazon Route 53, lalu mengaitkan pemeriksaan kesehatan dengan catatan DNS Anda. Untuk gagal melewati lalu lintas, kontrol perutean ARC menetapkan pemeriksaan kesehatan untuk gagal, sehingga Route 53 mengubah rute lalu lintas. (Pemeriksaan kesehatan tidak memvalidasi kesehatan aplikasi Anda; itu hanya digunakan sebagai metode untuk mengalihkan lalu lintas.)

Sebagai contoh, katakanlah Anda memiliki dua sel (Wilayah atau Zona Ketersediaan). Anda mengonfigurasi satu sebagai sel utama untuk aplikasi Anda, dan yang lainnya sebagai sel sekunder, untuk gagal.

Untuk mengatur pemeriksaan kesehatan untuk failover, Anda dapat melakukan hal berikut, misalnya:

1. Gunakan ARC CLI untuk membuat kontrol perutean untuk setiap sel.
2. Gunakan Route 53 CLI untuk membuat pemeriksaan kesehatan ARC di Route 53 untuk setiap kontrol perutean.
3. Gunakan CLI Route 53 untuk membuat dua catatan DNS failover di Route 53, dan kaitkan pemeriksaan kesehatan dengan masing-masing file.

5a. Buat kontrol perutean untuk setiap sel.

```
aws route53-recovery-control-config --region us-west-2 create-routing-control \
        --routing-control-name RoutingControlCell1 \
        --cluster-arn arn:aws:route53-recovery-control::111122223333:cluster/5678abcd-
abcd-5678-abcd-5678abcdefgh
```

```
aws route53-recovery-control-config --region us-west-2 create-routing-control \
        --routing-control-name RoutingControlCell2 \
        --cluster-arn arn:aws:route53-recovery-control::111122223333:cluster/5678abcd-
abcd-5678-abcd-5678abcdefgh
```

5b. Buat pemeriksaan kesehatan untuk setiap kontrol perutean.

> **Note**
>
> Anda membuat pemeriksaan kesehatan ARC dengan menggunakan Amazon Route 53 CLI.

```
aws route53 create-health-check --caller-reference RoutingControlCell1 \
        --health-check-config \
        Type=RECOVERY_CONTROL,RoutingControlArn=arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/routingcontrol/
abcdefg1234567
```

```
{
```

```
    "Location": "https://route53.amazonaws.com/2015-01-01/healthcheck/11111aaaa-bbbb-
cccc-dddd-ffffff22222",
    "HealthCheck": {
        "Id": "xxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
        "CallerReference": "RoutingControlCell1",
        "HealthCheckConfig": {
            "Type": "RECOVERY_CONTROL",
            "Inverted": false,
            "Disabled": false,
            "RoutingControlArn": "arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/routingcontrol/
abcdefg1234567"
        },
        "HealthCheckVersion": 1
    }
}
```

```
aws route53 create-health-check --caller-reference RoutingControlCell2 \
    --health-check-config \
    Type=RECOVERY_CONTROL,RoutingControlArn=arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/routingcontrol/
abcdefg1234567
```

```
{
    "Location": "https://route53.amazonaws.com/2015-01-01/healthcheck/11111aaaa-bbbb-
cccc-dddd-ffffff22222",
    "HealthCheck": {
        "Id": "xxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
        "CallerReference": "RoutingControlCell2",
        "HealthCheckConfig": {
            "Type": "RECOVERY_CONTROL",
            "Inverted": false,
            "Disabled": false,
            "RoutingControlArn": "arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/routingcontrol/
abcdefg1234567"
        },
        "HealthCheckVersion": 1
    }
}
```

5c. Buat dua catatan DNS failover, dan kaitkan pemeriksaan kesehatan dengan masing-masing file.

Anda membuat catatan DNS failover di Route 53 menggunakan CLI Route 53. Untuk membuat catatan, ikuti petunjuk di Referensi AWS CLI Perintah Amazon Route 53 untuk change-resource-record-setsperintah tersebut. Dalam catatan, tentukan nilai DNS untuk setiap sel bersama dengan `HealthCheckID` nilai yang sesuai yang dibuat Route 53 untuk pemeriksaan kesehatan (lihat 6b).

Untuk sel primer:

```
{
    "Name": "myapp.yourdomain.com",
    "Type": "CNAME",
    "SetIdentifier": "primary",
    "Failover": "PRIMARY",
    "TTL": 0,
    "ResourceRecords": [
        {
            "Value": "cell1.yourdomain.com"
        }
    ],
    "HealthCheckId": "xxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
}
```

Untuk sel sekunder:

```
{
    "Name": "myapp.yourdomain.com",
    "Type": "CNAME",
    "SetIdentifier": "secondary",
    "Failover": "SECONDARY",
    "TTL": 0,
    "ResourceRecords": [
        {
            "Value": "cell2.yourdomain.com"
        }
    ],
    "HealthCheckId": "yyyyyy-yyyy-yyyy-yyyy-yyyyyyyyyyyy"
}
```

Sekarang, untuk gagal dari sel primer Anda ke sel sekunder Anda, Anda dapat mengikuti contoh CLI di langkah 4b untuk memperbarui status ke dan `RoutingControlCell1` ke`OFF`. `RoutingControlCell2` `ON`

Daftar dan perbarui kontrol dan status perutean dengan AWS CLI

Setelah membuat resource Amazon Application Recovery Controller (ARC), seperti cluster, kontrol routing, dan panel kontrol, Anda dapat berinteraksi dengan cluster untuk mencantumkan dan memperbarui status kontrol perutean untuk failover.

Untuk setiap cluster yang Anda buat, ARC memberi Anda satu set titik akhir cluster, satu dari masing-masing lima Wilayah AWS. Anda harus menentukan salah satu titik akhir Regional ini (URL Wilayah AWS dan titik akhir) saat Anda melakukan panggilan ke klaster untuk mengambil atau mengatur status kontrol perutean ke atau. `On` `Off` Saat Anda menggunakan AWS CLI, untuk mendapatkan atau memperbarui status kontrol perutean, selain titik akhir Regional, Anda juga harus menentukan titik akhir Regional, seperti yang ditunjukkan pada contoh di bagian ini. `--region`

Anda dapat menggunakan salah satu titik akhir cluster Regional. Kami menyarankan agar sistem Anda berputar melalui titik akhir regional, dan bersiaplah untuk mencoba lagi dengan masing-masing titik akhir yang tersedia. Untuk contoh kode yang menggambarkan mencoba titik akhir cluster secara berurutan, lihat. Tindakan untuk Application Recovery Controller menggunakan AWS SDKs

Untuk informasi selengkapnya tentang penggunaan AWS CLI, lihat Referensi AWS CLI Perintah. Untuk daftar tindakan API kontrol perutean dan tautan ke informasi selengkapnya, lihat Operasi API kontrol perutean.

> **Important**
>
> Meskipun Anda dapat memperbarui status kontrol perutean di konsol Amazon Route 53, sebaiknya Anda memperbarui status kontrol perutean dengan menggunakan AWS CLI atau SDK. AWS ARC menawarkan keandalan ekstrim dengan bidang data kontrol perutean ARC untuk mengalihkan lalu lintas dan gagal di seluruh sel. Untuk rekomendasi selengkapnya tentang penggunaan ARC untuk failover, lihatPraktik terbaik untuk kontrol perutean di ARC.

Saat Anda membuat kontrol perutean, status diatur ke`Off`. Ini berarti bahwa lalu lintas tidak diarahkan ke sel target untuk kontrol perutean itu. Anda dapat memverifikasi status kontrol perutean dengan menjalankan perintah`get-routing-control-state`.

Untuk menentukan Wilayah dan titik akhir yang akan ditentukan, jalankan `describe-clusters` perintah untuk melihat. `ClusterEndpoints` Masing-masing `ClusterEndpoint` menyertakan Wilayah dan titik akhir terkait yang dapat Anda gunakan untuk mendapatkan atau memperbarui status kontrol perutean. DescribeClusteradalah operasi API konfigurasi kontrol pemulihan. Kami

menyarankan Anda menyimpan salinan lokal titik akhir cluster ARC Regional Anda, di bookmark atau di-hardcode dalam kode otomatisasi yang Anda gunakan untuk mencoba lagi titik akhir Anda.

1. Daftar kontrol perutean

Anda dapat melihat kontrol perutean dan status kontrol perutean menggunakan titik akhir bidang data ARC yang sangat andal.

1. Daftar kontrol routing untuk panel kontrol tertentu. Jika Anda tidak menentukan panel kontrol, `list-routing-controls` mengembalikan semua kontrol routing di cluster.

```
aws route53-recovery-cluster list-routing-controls --control-panel-arn \
        arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456 \
        --region us-west-2 \
        --endpoint-url https://host-dddddd.us-west-2.example.com/v1
```

```
{
    "RoutingControls": [{
        "ControlPanelArn": "arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456",
        "ControlPanelName": "ExampleControlPanel",
        "RoutingControlArn": "arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/routingcontrol/
abcdefg1234567",
        "RoutingControlName": "RCOne",
        "RoutingControlState": "On"
    },
    {
        "ControlPanelArn": "arn:aws:route53-recovery-
control::023759465626:controlpanel/0123456bbbbbbb0123456bbbbbb0123456",
        "ControlPanelName": "ExampleControlPanel",
        "RoutingControlArn": "arn:aws:route53-recovery-
control::023759465626:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/routingcontrol/
zzzzxxxxyyyy123456",
        "RoutingControlName": "RCTwo",
        "RoutingControlState": "Off"
    }
]
```

2. Dapatkan kontrol perutean

2. Dapatkan status kontrol perutean.

```
aws route53-recovery-cluster get-routing-control-state --routing-control-arn \
        arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/routingcontrol/
abcdefg1234567 \
        --region us-west-2 \
        --endpoint-url https://host-dddddd.us-west-2.example.com/v1
```

```
{"RoutingControlArn": "arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/routingcontrol/
abcdefg1234567",
    "RoutingControlName": "RCOne",
    "RoutingControlState": "On"
}
```

2. Perbarui kontrol perutean

Untuk merutekan lalu lintas ke titik akhir target yang dikendalikan oleh kontrol perutean, Anda memperbarui status kontrol perutean ke. 0n Perbarui status kontrol perutean dengan menjalankan perintah`update-routing-control-state`. (Ketika permintaan berhasil, responsnya kosong.)

2a. Perbarui status kontrol perutean.

```
aws route53-recovery-cluster update-routing-control-state \
        --routing-control-arn \
        arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/routingcontrol/
abcdefg1234567 \
        --routing-control-state On \
        --region us-west-2 \
        --endpoint-url https://host-dddddd.us-west-2.example.com/v1
```

```
{}
```

Anda dapat memperbarui beberapa kontrol perutean secara bersamaan dengan satu panggilan API:`update-routing-control-states`. (Ketika permintaan berhasil, responsnya kosong.)

2b. Perbarui beberapa status kontrol perutean sekaligus (pembaruan batch).

```
aws route53-recovery-cluster update-routing-control-states \
        --update-routing-control-state-entries \
```

```
        '[{"RoutingControlArn": "arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/routingcontrol/
abcdefg1234567",
        "RoutingControlState": "Off"}, \
        {"RoutingControlArn": "arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/routingcontrol/
hijklmnop987654321",
        "RoutingControlState": "On"}]' \
        --region us-west-2 \
        --endpoint-url https://host-dddddd.us-west-2.example.com/v1
```

```
{}
```

Bekerja dengan komponen kontrol perutean di ARC

Topik

- Membuat komponen kontrol perutean di ARC
- Melihat dan memperbarui status kontrol perutean di ARC
- Membuat aturan keselamatan untuk kontrol perutean
- Support cross-account untuk cluster di ARC

Membuat komponen kontrol perutean di ARC

Bagian ini menjelaskan cara membuat cluster, kontrol routing, pemeriksaan kesehatan, dan panel kontrol untuk bekerja dengan kontrol routing di Amazon Application Recovery Controller (ARC).

Mulailah dengan membuat cluster, untuk meng-host kontrol routing Anda dan panel kontrol yang Anda gunakan untuk mengelompokkannya. Kemudian buat kontrol perutean dan pemeriksaan kesehatan sehingga Anda dapat mengubah rute lalu lintas untuk gagal dari satu sel ke sel lainnya, sehingga lalu lintas masuk ke replika cadangan Anda, misalnya.

Perhatikan bahwa Anda dikenakan biaya per jam untuk setiap cluster yang Anda buat. Anda biasanya hanya memerlukan satu cluster untuk meng-host kontrol routing dan panel kontrol untuk manajemen kontrol pemulihan untuk aplikasi. Selain itu, Anda dapat mengatur berbagi sumber daya dengan menggunakan AWS Resource Access Manager, sehingga satu cluster dapat meng-host kontrol perutean dan sumber daya ARC lainnya yang dimiliki oleh beberapa Akun AWS. Untuk mempelajari tentang berbagi sumber daya di ARC, Support cross-account untuk cluster di ARC

Untuk informasi harga, lihat Harga Amazon Application Recovery Controller (ARC) dan gulir ke bawah ke Amazon Route 53.

Untuk menggunakan kontrol perutean agar gagal atas lalu lintas, Anda membuat pemeriksaan kesehatan kontrol perutean yang Anda kaitkan dengan data DNS Amazon Route 53 untuk sumber daya dalam aplikasi Anda. Sebagai contoh, katakanlah Anda memiliki dua sel, satu yang telah Anda konfigurasikan sebagai sel utama untuk aplikasi Anda, dan yang lainnya yang telah Anda konfigurasikan sebagai sel sekunder, untuk gagal.

Untuk mengatur pemeriksaan kesehatan untuk failover, lakukan hal berikut:

1. Buat kontrol perutean untuk setiap sel.
2. Buat pemeriksaan kesehatan untuk setiap kontrol perutean.
3. Buat dua catatan DNS, misalnya, dua catatan failover DNS, dan kaitkan pemeriksaan kesehatan dengan masing-masing file.

Skenario lain ketika Anda mungkin membuat kontrol routing adalah ketika Anda membuat aturan keamanan yang merupakan aturan gating. Dalam hal ini, Anda tidak mengaitkan pemeriksaan kesehatan dan catatan DNS dengan kontrol perutean karena Anda akan menggunakannya sebagai kontrol perutean gating. Untuk informasi selengkapnya, lihat Membuat aturan keselamatan untuk kontrol perutean .

Langkah-langkah untuk membuat komponen untuk kontrol perutean pada konsol ARC disertakan dalam bagian ini. Untuk mempelajari tentang menggunakan operasi API konfigurasi kontrol pemulihan dengan ARC, lihat Operasi API kontrol perutean.

Membuat cluster di ARC

Anda harus membuat cluster untuk meng-host kontrol routing dan panel kontrol di ARC.

Cluster adalah sekumpulan titik akhir Regional redundan yang dengannya Anda dapat menjalankan panggilan API untuk memperbarui atau mendapatkan status dari satu atau beberapa kontrol perutean. Sebuah cluster tunggal dapat meng-host sejumlah kontrol routing.

> **Important**
>
> Ketahuilah bahwa Anda dikenakan biaya per jam untuk setiap cluster yang Anda buat. Satu cluster dapat menampung sejumlah kontrol routing dan panel kontrol untuk manajemen kontrol pemulihan, biasanya cukup untuk sebuah aplikasi.

Untuk membuat klaster DB

1. Buka konsol ARC dihttps://console.aws.amazon.com/route53recovery/home#/dashboard.
2. Pilih Klaster.
3. Pilih Buat, lalu masukkan nama untuk klaster Anda.
4. Pilih Buat klaster.

Membuat kontrol perutean di ARC

Buat kontrol perutean untuk setiap sel yang ingin Anda rutekan lalu lintas. Misalnya, ketika Anda memiliki aplikasi dengan sumber daya yang telah Anda silokan untuk pemulihan, Anda mungkin memiliki sel untuk masing-masing Wilayah AWS, dan sel bersarang untuk setiap Availability Zone dalam setiap Region. Dalam skenario ini, Anda akan membuat kontrol perutean untuk setiap sel dan setiap sel bersarang.

Ketika Anda membuat kontrol routing, perlu diingat bahwa nama kontrol routing harus unik dalam setiap panel kontrol.

Setelah Anda membuat kontrol perutean yang akan digunakan untuk mengalihkan lalu lintas, Anda mengaitkan masing-masing dengan pemeriksaan kesehatan, yang memungkinkan Anda merutekan lalu lintas ke sel, berdasarkan catatan DNS yang Anda kaitkan dengan masing-masing. Jika Anda menyiapkan aturan gating sebagai aturan keselamatan dan membuat kontrol routing gating, Anda tidak menambahkan pemeriksaan kesehatan ke kontrol perutean.

Untuk membuat kontrol routing

1. Buka konsol ARC dihttps://console.aws.amazon.com/route53recovery/home#/dashboard.
2. Pilih kontrol Routing.
3. Pada halaman kontrol Routing, pilih Buat, lalu pilih kontrol Routing.
4. Masukkan nama untuk kontrol perutean Anda, pilih klaster untuk menambahkan kontrol, dan pilih untuk menambahkannya ke panel kontrol yang ada, termasuk menggunakan panel kontrol default. Atau, buat panel kontrol baru.
5. Jika Anda memilih untuk membuat panel kontrol baru, pilih cluster untuk membuat panel kontrol, dan kemudian masukkan nama untuk panel.
6. Pilih Buat kontrol perutean.
7. Ikuti langkah-langkah untuk memberi nama dan membuat kontrol perutean.

Membuat pemeriksaan kesehatan kontrol perutean di ARC

Anda mengaitkan pemeriksaan kesehatan kontrol perutean dengan setiap kontrol perutean yang ingin Anda gunakan untuk mengalihkan lalu lintas. Kemudian Anda mengonfigurasi setiap pemeriksaan kesehatan dengan catatan DNS Amazon Route 53, misalnya, catatan DNS failover. Kemudian Anda dapat mengubah rute lalu lintas di Amazon Application Recovery Controller (ARC) hanya dengan memperbarui status kontrol routing terkait, untuk mengaturnya ke atau. `On` `Off`

> **Note**
>
> Anda tidak dapat mengedit pemeriksaan kesehatan kontrol perutean yang ada untuk mengaitkannya dengan kontrol perutean yang berbeda.

Untuk membuat pemeriksaan kesehatan kontrol perutean

1. Buka konsol ARC dihttps://console.aws.amazon.com/route53recovery/home#/dashboard.
2. Pilih kontrol Routing.
3. Pada halaman kontrol Routing, pilih kontrol routing.
4. Pada halaman Routing control detail, pilih Create health check.
5. Masukkan nama untuk pemeriksaan kesehatan, lalu pilih Buat.

Selanjutnya, Anda membuat catatan DNS Route 53, dan mengaitkan pemeriksaan kesehatan kontrol perutean Anda dengan masing-masing data. Misalnya, mari kita asumsikan bahwa Anda ingin menggunakan dua catatan failover DNS untuk mengaitkan pemeriksaan kesehatan kontrol perutean Anda. Agar ARC gagal dengan benar melalui lalu lintas dengan menggunakan kontrol perutean, mulailah dengan membuat dua catatan failover di Route 53: primer dan sekunder. Untuk informasi selengkapnya tentang mengonfigurasi catatan failover DNS, lihat Konsep pemeriksaan Kesehatan.

Saat Anda membuat catatan failover utama, nilainya harus seperti berikut:

```
Name: myapp.yourdomain.com
Type: CNAME
Set Identifier: Primary
Failover: Primary
TTL: 0
Resource Records:
```

```
Value: cell1.yourdomain.com
Health Check ID: xxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
```

Nilai catatan failover sekunder harus seperti berikut:

```
Name: myapp.yourdomain.com
Type: CNAME
Set Identifier: Secondary
Failover: Secondary
TTL: 0
Resource Records:
Value: cell2.yourdomain.com
Health Check ID: xxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
```

Sekarang, katakan bahwa Anda ingin mengubah rute lalu lintas karena ada kegagalan. Untuk melakukan ini, Anda memperbarui status kontrol perutean terkait untuk mengubah status kontrol perutean utama ke OFF dan status kontrol perutean sekunder ke. ON Ketika Anda melakukan ini, pemeriksaan kesehatan terkait menghentikan lalu lintas dari pergi ke replika utama dan mengarahkannya ke replika sekunder. Untuk informasi selengkapnya tentang kegagalan lalu lintas dengan kontrol perutean, lihat. Mendapatkan dan memperbarui status kontrol perutean menggunakan ARC API (disarankan)

Untuk melihat contoh AWS CLI perintah untuk membuat kontrol perutean dan pemeriksaan kesehatan terkait menggunakan operasi ARC API, lihat Contoh penggunaan operasi API kontrol perutean ARC dengan AWS CLI.

Membuat panel kontrol di ARC

Panel kontrol di Amazon Application Recovery Controller (ARC) memungkinkan Anda mengelompokkan kontrol perutean terkait. Panel kontrol dapat memiliki kontrol routing yang mewakili layanan mikro dalam aplikasi, seluruh aplikasi itu sendiri, atau sekelompok aplikasi, tergantung pada ruang lingkup failover Anda. Manfaat pengelompokan kontrol perutean ke dalam panel kontrol adalah Anda dapat menggunakan aturan keselamatan dengan panel kontrol untuk membantu melindungi perubahan perutean lalu lintas.

Saat Anda membuat cluster, ARC membuat panel kontrol default. Anda dapat menggunakan panel kontrol default untuk kontrol routing Anda, atau Anda dapat membuat satu atau beberapa panel kontrol untuk mengelompokkan kontrol routing Anda. Perhatikan bahwa hanya karakter ASCII yang didukung untuk nama panel kontrol.

Langkah-langkah untuk membuat panel kontrol pada konsol ARC disertakan dalam bagian ini. Untuk informasi tentang penggunaan operasi API konfigurasi kontrol pemulihan dengan ARC, lihat Operasi API kontrol perutean.

Untuk membuat panel kontrol

1. Buka konsol ARC dihttps://console.aws.amazon.com/route53recovery/home#/dashboard.
2. Pilih kontrol Routing.
3. Pada halaman kontrol Routing, pilih Create, dan kemudian pilih Control panel.
4. Pilih cluster untuk membuat panel kontrol, dan kemudian masukkan nama untuk panel.
5. Pilih Buat panel kontrol.

Melihat dan memperbarui status kontrol perutean di ARC

Bagian ini menjelaskan cara melihat dan memperbarui status kontrol perutean di Amazon Application Recovery Controller (ARC). Kontrol perutean adalah sakelar on-off sederhana yang mengelola arus lalu lintas ke sel dalam grup pemulihan Anda. Sel biasanya Wilayah AWS, atau terkadang Availability Zone, yang mencakup sumber daya Anda. Ketika status kontrol routing0n, lalu lintas mengalir ke sel yang dikendalikan oleh kontrol routing itu.

Anda mengelompokkan kontrol perutean ke dalam panel kontrol, yang merupakan pengelompokan failover logis. Saat Anda membuka panel kontrol di konsol, misalnya, Anda dapat melihat semua kontrol perutean untuk pengelompokan sekaligus, untuk melihat di mana lalu lintas mengalir.

Anda dapat memperbarui status kontrol perutean di konsol ARC atau dengan menggunakan ARC API. Kami menyarankan Anda memperbarui status kontrol perutean dengan menggunakan API. Pertama, ARC menawarkan keandalan ekstrim dengan API di bidang data untuk melakukan tindakan ini. Itu penting saat Anda mengubah status ini karena perubahan status perutean gagal di seluruh sel dengan mengalihkan lalu lintas aplikasi. Selain itu, dengan menggunakan API, Anda dapat mencoba menghubungkan ke titik akhir cluster yang berbeda dalam rotasi, sesuai kebutuhan, jika titik akhir cluster yang Anda coba sambungkan tidak tersedia.

Anda dapat memperbarui satu status kontrol perutean, atau Anda dapat memperbarui beberapa status kontrol perutean sekaligus. Misalnya, Anda mungkin ingin menyetel satu status kontrol perutean `Off` untuk menghentikan lalu lintas mengalir ke satu sel, seperti Availability Zone tempat aplikasi mengalami peningkatan latensi. Pada saat yang sama, Anda mungkin ingin mengatur status kontrol perutean lain untuk memulai lalu lintas yang mengalir 0n ke sel lain atau Availability Zone.

Dalam skenario ini, Anda dapat memperbarui kedua status kontrol perutean secara bersamaan, sehingga lalu lintas terus mengalir.

Topik

- Mendapatkan dan memperbarui status kontrol perutean menggunakan ARC API (disarankan)
- Mendapatkan dan memperbarui status kontrol perutean di AWS Management Console

Mendapatkan dan memperbarui status kontrol perutean menggunakan ARC API (disarankan)

Kami menyarankan Anda menggunakan operasi API Amazon Application Recovery Controller (ARC) untuk mendapatkan atau memperbarui status kontrol perutean, dengan menggunakan AWS CLI perintah atau dengan menggunakan kode yang telah Anda kembangkan untuk menggunakan operasi ARC API dengan salah satu. AWS SDKs Kami merekomendasikan penggunaan operasi API, dengan CLI atau dalam kode, untuk bekerja dengan status kontrol perutean, daripada menggunakan. AWS Management Console

ARC menawarkan keandalan ekstrim untuk kegagalan di seluruh sel (Wilayah AWS) dengan memperbarui status kontrol perutean menggunakan API karena kontrol perutean disimpan dalam cluster yang sangat tersedia. ARC memastikan bahwa setidaknya tiga dari lima titik akhir cluster Regional selalu dapat diakses oleh Anda untuk membuat perubahan status kontrol perutean. Untuk mendapatkan atau mengubah status kontrol perutean menggunakan API, Anda terhubung ke salah satu titik akhir klaster Regional Anda. Jika titik akhir tidak tersedia, Anda dapat mencoba menghubungkan ke salah satu titik akhir cluster Anda yang lain.

Anda dapat melihat daftar titik akhir klaster Regional untuk klaster Anda di konsol Route 53, atau dengan menggunakan tindakan API, DescribeCluster. Proses Anda untuk mendapatkan dan mengubah status kontrol perutean harus mencoba setiap titik akhir dalam rotasi, sesuai kebutuhan, karena titik akhir cluster didaur ulang melalui status yang tersedia dan tidak tersedia untuk pemeliharaan dan pembaruan rutin.

Kami memberikan informasi terperinci dan contoh kode untuk menggunakan operasi ARC API untuk mendapatkan dan memperbarui status kontrol perutean, dan bekerja dengan titik akhir cluster Regional. Untuk informasi selengkapnya, lihat berikut ini:

- Untuk contoh kode yang menjelaskan cara memutar melalui titik akhir cluster Regional untuk mendapatkan dan mengatur status kontrol perutean, lihat. Tindakan untuk Application Recovery Controller menggunakan AWS SDKs

- Untuk informasi tentang menggunakan status kontrol perutean AWS CLI untuk mendapatkan dan memperbarui, lihat Daftar dan perbarui kontrol dan status perutean dengan AWS CLI.

Mendapatkan dan memperbarui status kontrol perutean di AWS Management Console

Anda bisa mendapatkan dan memperbarui status kontrol perutean di file. AWS Management Console Namun, ketahuilah bahwa Anda tidak dapat memilih titik akhir cluster Regional yang berbeda di konsol. Artinya, tidak ada proses untuk memilih dan memutar melalui titik akhir cluster di konsol seperti yang dapat Anda lakukan dengan menggunakan Amazon Application Recovery Controller (ARC) API. Selain itu, konsol ini tidak terlalu tersedia sementara bidang data ARC menawarkan keandalan yang ekstrim. Untuk alasan ini, kami menyarankan Anda menggunakan ARC API untuk mendapatkan dan memperbarui status kontrol perutean untuk operasi produksi.

Untuk rekomendasi selengkapnya tentang penggunaan ARC untuk failover, lihatPraktik terbaik untuk kontrol perutean di ARC.

Untuk melihat dan memperbarui kontrol perutean di konsol, ikuti langkah-langkah dalam prosedur berikut.

Untuk mendapatkan status kontrol perutean

1. Buka konsol ARC dihttps://console.aws.amazon.com/route53recovery/home#/dashboard.
2. Pilih kontrol Routing.
3. Dari daftar, pilih panel kontrol dan lihat kontrol perutean.

Untuk memperbarui satu atau beberapa status kontrol perutean

1. Buka konsol Amazon Route 53 di https://console.aws.amazon.com/route53/rumah.
2. Di bawah Application Recovery Controller, pilih Routing control.
3. Pilih Tindakan, lalu pilih Ubah perutean lalu lintas.
4. Perbarui status satu atau lebih kontrol perutean menjadi `Off` atau`On`, tergantung di mana Anda ingin lalu lintas mengalir atau berhenti mengalir untuk aplikasi Anda.
5. Masukkan `confirm` di kotak teks.
6. Pilih Perbarui perutean lalu lintas.

Membuat aturan keselamatan untuk kontrol perutean

Ketika Anda bekerja dengan beberapa kontrol routing pada saat yang sama, Anda mungkin memutuskan bahwa Anda ingin perlindungan di tempat untuk menghindari konsekuensi yang tidak diinginkan. Misalnya, Anda mungkin ingin mencegah secara tidak sengaja mematikan semua kontrol perutean untuk aplikasi, yang akan menghasilkan skenario fail-open. Atau Anda mungkin ingin menerapkan sakelar on-off master untuk menonaktifkan serangkaian kontrol perutean, mungkin untuk mencegah otomatisasi mengalihkan lalu lintas. Untuk menetapkan perlindungan seperti ini untuk kontrol perutean di ARC, Anda membuat aturan keselamatan.

Anda mengonfigurasi aturan keselamatan untuk kontrol perutean dengan kombinasi kontrol perutean, aturan, dan opsi lain yang Anda tentukan. Setiap aturan keselamatan dikaitkan dengan panel kontrol tunggal, tetapi panel kontrol dapat memiliki lebih dari satu aturan keselamatan. Saat Anda membuat aturan keselamatan, ingatlah bahwa nama aturan keselamatan harus unik di setiap panel kontrol.

Topik

Jenis aturan keselamatan

Ada dua jenis aturan keselamatan, aturan assertion dan aturan gating, yang dapat Anda gunakan untuk melindungi failover dengan cara yang berbeda.

Aturan penegasan

Dengan aturan pernyataan, saat Anda mengubah satu atau satu set status kontrol perutean, ARC memberlakukan bahwa kriteria yang Anda tetapkan saat Anda mengonfigurasi aturan terpenuhi, atau status kontrol perutean tidak berubah.

Contoh kapan ini berguna adalah untuk mencegah skenario fail-open, seperti skenario di mana Anda menghentikan lalu lintas dari pergi ke satu sel tetapi tidak memulai lalu lintas mengalir ke sel lain. Untuk menghindari hal ini, aturan assertion memastikan bahwa setidaknya satu kontrol routing dalam satu set kontrol routing di panel kontrol adalah 0n pada waktu tertentu. Ini memastikan bahwa lalu lintas mengalir ke setidaknya satu Wilayah atau Zona Ketersediaan untuk suatu aplikasi.

Untuk melihat AWS CLI perintah contoh yang membuat aturan pernyataan untuk menerapkan kriteria ini, lihat Membuat aturan keselamatan di. Contoh penggunaan operasi API kontrol perutean ARC dengan AWS CLI

Untuk informasi terperinci tentang properti operasi API aturan pernyataan, lihat AssertionRuledi Panduan Referensi API Kontrol Perutean untuk Pengontrol Pemulihan Aplikasi Amazon.

Aturan gerbang

Dengan aturan gating, Anda dapat menerapkan sakelar on-off secara keseluruhan melalui serangkaian kontrol perutean sehingga apakah status kontrol perutean tersebut dapat diubah diberlakukan berdasarkan serangkaian kriteria yang Anda tentukan dalam aturan. Kriteria paling sederhana adalah apakah kontrol perutean tunggal yang Anda tentukan sebagai sakelar diatur ke `ON` atau`OFF`.

Untuk mengimplementasikannya, Anda membuat kontrol perutean gating, untuk digunakan sebagai sakelar keseluruhan, dan kontrol perutean target, untuk mengontrol arus lalu lintas ke Wilayah atau Zona Ketersediaan yang berbeda. Kemudian, untuk mencegah pembaruan status manual atau otomatis ke kontrol perutean target yang telah Anda konfigurasikan untuk aturan gating, Anda mengatur status kontrol perutean gating ke. `Off` Untuk mengizinkan pembaruan, Anda mengaturnya ke`On`.

Untuk melihat AWS CLI perintah contoh yang membuat aturan gating yang mengimplementasikan sakelar keseluruhan semacam ini, lihat Membuat aturan keselamatan di. Contoh penggunaan operasi API kontrol perutean ARC dengan AWS CLI

Untuk informasi terperinci tentang properti operasi API aturan gating, lihat GatingRuledi Panduan Referensi API Kontrol Perutean untuk Pengontrol Pemulihan Aplikasi Amazon.

Membuat aturan keamanan di konsol

Langkah-langkah di bagian ini menjelaskan cara membuat aturan keamanan di konsol ARC. Langkah-langkahnya serupa apakah Anda membuat aturan pernyataan atau aturan gating. Perbedaannya dicatat dalam prosedur.

Untuk mempelajari cara menggunakan operasi API kontrol pemulihan dan perutean dengan Amazon Application Recovery Controller (ARC), lihat Operasi API kontrol perutean.

Untuk membuat aturan keamanan

1. Buka konsol ARC dihttps://console.aws.amazon.com/route53recovery/home#/dashboard.

2. Pilih kontrol Routing.
3. Pada halaman kontrol Routing, pilih panel kontrol.
4. Pada halaman detail panel kontrol, pilih Tindakan, lalu pilih Tambahkan aturan keamanan.
5. Pilih jenis aturan untuk ditambahkan: Aturan pernyataan atau aturan Gating.
6. Pilih nama dan, secara opsional, ubah periode tunggu.
7. Tentukan opsi konfigurasi untuk aturan keamanan.

   - Untuk aturan pernyataan, tentukan kontrol perutean yang ditegaskan.
   - Untuk aturan gating, tentukan kontrol perutean gating dan kontrol perutean target.

   Untuk kedua aturan, tentukan konfigurasi aturan dengan memilih jenis dan ambang batas, dan apakah aturan terbalik.

   > **Note**
   >
   > Untuk mempelajari lebih lanjut tentang menentukan aturan pernyataan, lihat informasi yang disediakan untuk AssertionRulepengoperasian di Panduan Referensi API Kontrol Perutean untuk Pengontrol Pemulihan Aplikasi Amazon. Untuk mempelajari lebih lanjut tentang menentukan aturan gating, lihat informasi yang disediakan untuk GatingRulepengoperasian di Panduan Referensi API Kontrol Perutean untuk Pengontrol Pemulihan Aplikasi Amazon.

8. Pilih Buat.

Mengedit atau menghapus aturan keamanan di konsol

Langkah-langkah di bagian ini menjelaskan cara mengedit atau menghapus aturan keamanan di konsol ARC. Anda hanya dapat melakukan pengeditan terbatas pada aturan keamanan, untuk mengubah nama atau memperbarui periode tunggu. Untuk membuat perubahan lain, hapus dan buat ulang aturan keamanan.

Untuk mempelajari cara menggunakan operasi API dengan Amazon Application Recovery Controller (ARC), lihat Operasi API kontrol perutean.

Untuk menghapus aturan keamanan

1. Buka konsol ARC dihttps://console.aws.amazon.com/route53recovery/home#/dashboard.

2. Pilih kontrol Routing.
3. Pada halaman kontrol Routing, pilih panel kontrol.
4. Pada halaman detail panel kontrol, pilih aturan keamanan, lalu pilih Hapus atau Edit.

Mengesampingkan aturan keselamatan untuk mengubah rute lalu lintas

Ada skenario ketika Anda mungkin ingin melewati perlindungan kontrol perutean yang diberlakukan dengan aturan keselamatan yang telah Anda konfigurasi. Misalnya, Anda mungkin ingin gagal dengan cepat untuk pemulihan bencana, dan satu atau lebih aturan keselamatan mungkin secara tak terduga mencegah Anda memperbarui status kontrol perutean untuk mengalihkan lalu lintas. Dalam skenario “break glass” seperti ini, Anda dapat mengganti satu atau beberapa aturan keselamatan untuk mengubah status kontrol perutean dan gagal atas aplikasi Anda.

Anda dapat melewati aturan keselamatan saat memperbarui status kontrol perutean (atau beberapa status kontrol perutean) dengan menggunakan `update-routing-control-states` AWS CLI perintah `update-routing-control-state` or dengan parameter. `safety-rules-to-override` Tentukan parameter dengan Nama Sumber Daya Amazon (ARN) dari aturan keselamatan yang ingin Anda ganti, atau tentukan daftar terpisah koma ARNs untuk mengganti dua atau beberapa aturan keselamatan.

Ketika aturan keamanan memblokir pembaruan status kontrol perutean, pesan kesalahan menyertakan ARN dari aturan yang memblokir pembaruan. Jadi Anda dapat membuat catatan ARN, dan kemudian menentukannya dalam perintah CLI status kontrol perutean dengan parameter penggantian aturan keselamatan.

> **Note**
>
> Karena lebih dari satu aturan keamanan mungkin ada untuk kontrol perutean yang Anda perbarui, Anda dapat menjalankan perintah CLI untuk memperbarui status kontrol perutean Anda dengan satu penggantian aturan keselamatan tetapi mendapatkan kesalahan bahwa aturan keselamatan lain memblokir pembaruan. Terus tambahkan aturan keamanan ARNs ke daftar aturan yang akan diganti dalam perintah pembaruan, dipisahkan dengan koma, hingga perintah pembaruan berhasil diselesaikan.

Untuk mempelajari lebih lanjut tentang menggunakan `SafetyRulesToOverride` properti dengan API dan SDKs, lihat UpdateRoutingControlState.

Berikut ini adalah dua contoh perintah CLI untuk mengganti aturan keselamatan untuk memperbarui status kontrol perutean.

Ganti satu aturan keamanan

```
aws route53-recovery-cluster --region us-west-2 update-routing-control-state \
    --routing-control-arn \
    arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/
routingcontrol/abcdefg1234567 \
    --routing-control-state On \
    --safety-rules-to-override arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/safetyrule/
yyyyyyy8888888 \
    --endpoint-url https://host-dddddd.us-west-2.example.com/v1
```

Ganti dua aturan keselamatan

```
aws route53-recovery-cluster --region us-west-2 update-routing-control-state \
    --routing-control-arn \
    arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/
routingcontrol/abcdefg1234567 \
    --routing-control-state On \
    --safety-rules-to-override "arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/safetyrule/
yyyyyyy8888888" \
    "arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/safetyrule/
qqqqqqq7777777"
    --endpoint-url https://host-dddddd.us-west-2.example.com/v1
```

Support cross-account untuk cluster di ARC

Amazon Application Recovery Controller (ARC) terintegrasi dengan AWS Resource Access Manager untuk mengaktifkan berbagi sumber daya. AWS RAM adalah layanan yang memungkinkan Anda untuk berbagi sumber daya dengan orang lain Akun AWS atau melalui AWS Organizations. Untuk kontrol perutean ARC, Anda dapat berbagi sumber daya cluster.

Dengan AWS RAM, Anda berbagi sumber daya yang Anda miliki dengan membuat pembagian sumber daya. Pembagian sumber daya menentukan sumber daya untuk dibagikan, dan peserta untuk membagikannya. Peserta dapat mencakup:

- Khusus Akun AWS di dalam atau di luar organisasi pemilik di AWS Organizations
- Unit organisasi di dalam organisasinya di AWS Organizations
- Seluruh organisasinya di AWS Organizations

Untuk informasi selengkapnya AWS RAM, lihat Panduan AWS RAM Pengguna.

Dengan menggunakan AWS Resource Access Manager untuk berbagi sumber daya klaster di seluruh akun di ARC, Anda dapat menggunakan satu cluster untuk meng-host panel kontrol dan kontrol perutean yang dimiliki oleh beberapa yang berbeda Akun AWS. Saat Anda memilih untuk berbagi klaster, cluster lain Akun AWS yang Anda tentukan dapat menggunakan klaster untuk meng-host panel kontrol dan kontrol perutean mereka sendiri, memungkinkan lebih banyak kontrol dan fleksibilitas atas kemampuan perutean di berbagai tim.

AWS RAM adalah layanan yang membantu AWS pelanggan berbagi sumber daya dengan aman. Akun AWS Dengan AWS RAM, Anda dapat berbagi sumber daya dalam organisasi atau unit organisasi (OUs) di AWS Organizations, dengan menggunakan peran dan pengguna IAM. AWS RAM adalah cara terpusat dan terkontrol untuk berbagi cluster.

Saat berbagi klaster, Anda dapat mengurangi jumlah total cluster yang dibutuhkan organisasi Anda. Dengan cluster bersama, Anda dapat mengalokasikan total biaya menjalankan cluster di berbagai tim, untuk memaksimalkan manfaat ARC dengan biaya lebih rendah. (Membuat sumber daya yang di-host di cluster tidak memiliki biaya tambahan, untuk pemilik atau untuk peserta.) Berbagi cluster di seluruh akun juga dapat memudahkan proses orientasi beberapa aplikasi ke ARC, terutama jika Anda memiliki sejumlah besar aplikasi yang didistribusikan di beberapa akun dan tim operasi.

Untuk memulai berbagi lintas akun di ARC, Anda membuat pembagian sumber daya. AWS RAM Pembagian sumber daya menentukan peserta yang berwenang untuk berbagi klaster yang dimiliki akun Anda. Kemudian, peserta dapat membuat sumber daya, seperti panel kontrol dan kontrol perutean, di cluster, dengan menggunakan AWS Management Console atau dengan menjalankan operasi ARC API menggunakan AWS Command Line Interface or AWS SDKs.

Topik ini menjelaskan cara berbagi sumber daya yang Anda miliki, dan cara menggunakan sumber daya yang dibagikan dengan Anda.

Daftar Isi

Prasyarat untuk berbagi cluster

- Untuk berbagi cluster, Anda harus memilikinya di Akun AWS. Ini berarti bahwa sumber daya harus dialokasikan atau disediakan di akun Anda. Anda tidak dapat berbagi cluster yang telah dibagikan dengan Anda.
- Untuk berbagi klaster dengan organisasi atau unit organisasi AWS Organizations, Anda harus mengaktifkan berbagi dengan AWS Organizations. Untuk informasi lebih lanjut, lihat Aktifkan pembagian dengan AWS Organizations dalam Panduan Pengguna AWS RAM .

Berbagi klaster

Saat Anda berbagi cluster yang Anda miliki, peserta yang Anda tentukan untuk berbagi cluster dapat membuat dan meng-host sumber daya ARC mereka sendiri di cluster.

Untuk berbagi cluster, Anda harus menambahkannya ke berbagi sumber daya. Pembagian sumber daya adalah AWS RAM sumber daya yang memungkinkan Anda berbagi sumber daya Anda Akun AWS. Pembagian sumber daya menentukan sumber daya untuk dibagikan, dan peserta yang mereka bagikan. Untuk berbagi klaster, Anda dapat membuat pembagian sumber daya baru atau menambahkan sumber daya ke pembagian sumber daya yang ada. Untuk membuat pembagian sumber daya baru, Anda dapat menggunakan AWS RAM konsol, atau menggunakan operasi AWS RAM API dengan AWS Command Line Interface atau AWS SDKs.

Jika Anda adalah bagian dari organisasi AWS Organizations dan berbagi dalam organisasi Anda diaktifkan, peserta dalam organisasi Anda secara otomatis diberikan akses ke cluster bersama. Jika tidak, peserta menerima undangan untuk bergabung dengan pembagian sumber daya dan diberikan akses ke cluster bersama setelah menerima undangan.

Anda dapat membagikan klaster yang Anda miliki dengan menggunakan AWS RAM konsol, atau dengan menggunakan operasi AWS RAM API dengan AWS CLI atau SDKs.

Untuk berbagi cluster yang Anda miliki dengan menggunakan AWS RAM konsol

Lihat Membuat berbagi sumber daya di Panduan AWS RAM Pengguna.

Untuk berbagi cluster yang Anda miliki dengan menggunakan AWS CLI

Gunakan perintah create-resource-share.

Memberikan izin untuk berbagi kluster

Berbagi kluster di seluruh akun memerlukan izin untuk prinsipal IAM yang berbagi kluster melalui. AWS RAM

Sebaiknya gunakan kebijakan IAM `AmazonRoute53RecoveryControlConfigFullAccess` terkelola untuk memastikan bahwa kepala sekolah IAM Anda memiliki izin yang diperlukan untuk berbagi dan menggunakan kluster bersama.

Berbagi klaster menggunakan kebijakan IAM khusus memerlukan`route53-recovery-control-config:PutResourcePolicy`,`route53-recovery-control-config:GetResourcePolicy`, dan `route53-recovery-control-config:DeleteResourcePolicy` izin untuk klaster tersebut. `PutResourcePolicy`dan `DeleteResourcePolicy` merupakan tindakan IAM khusus izin. Mencoba berbagi cluster AWS RAM tanpa izin ini akan menghasilkan kesalahan.

Untuk informasi selengkapnya tentang cara AWS Resource Access Manager menggunakan IAM, lihat Cara AWS Resource Access Manager menggunakan IAM di AWS RAM Panduan Pengguna.

Membatalkan berbagi kluster bersama

Saat Anda membatalkan pembagian klaster, berikut ini berlaku untuk peserta dan pemilik:

- Sumber daya peserta saat ini terus ada di cluster yang tidak dibagikan.
- Peserta dapat terus memperbarui status kontrol perutean di cluster yang tidak dibagikan, untuk mengelola perutean untuk failover aplikasi.
- Peserta tidak dapat lagi membuat sumber daya baru di cluster yang tidak dibagikan.
- Jika peserta masih memiliki sumber daya dalam klaster yang tidak dibagikan, pemilik tidak dapat menghapus kluster bersama.

Untuk membatalkan berbagi kluster bersama yang Anda miliki, hapus dari pembagian sumber daya. Anda dapat melakukan ini dengan menggunakan AWS RAM konsol atau dengan menggunakan operasi AWS RAM API dengan AWS CLI atau SDKs.

Untuk membatalkan berbagi cluster bersama yang Anda miliki menggunakan konsol AWS RAM

Lihat Memperbarui Pembagian Sumber Daya di Panduan Pengguna.AWS RAM

Untuk membatalkan berbagi cluster bersama yang Anda miliki menggunakan AWS CLI

Gunakan perintah disassociate-resource-share.

Mengidentifikasi cluster bersama

Pemilik dan peserta dapat mengidentifikasi cluster bersama dengan melihat informasi di AWS RAM. Mereka juga bisa mendapatkan informasi tentang sumber daya bersama dengan menggunakan konsol ARC dan AWS CLI.

Secara umum, untuk mempelajari lebih lanjut tentang sumber daya yang telah Anda bagikan atau yang telah dibagikan dengan Anda, lihat informasi di Panduan AWS Resource Access Manager Pengguna:

- Sebagai pemilik, Anda dapat melihat semua sumber daya yang Anda bagikan dengan orang lain dengan menggunakan AWS RAM. Untuk informasi selengkapnya, lihat Melihat sumber daya bersama Anda di AWS RAM.
- Sebagai peserta, Anda dapat melihat semua sumber daya yang dibagikan dengan Anda menggunakan AWS RAM. Untuk informasi selengkapnya, lihat Melihat sumber daya bersama Anda di AWS RAM.

Sebagai pemilik, Anda dapat menentukan apakah Anda berbagi cluster dengan melihat informasi di AWS Management Console atau dengan menggunakan operasi AWS Command Line Interface With ARC API.

Untuk mengidentifikasi apakah cluster yang Anda miliki dibagikan dengan menggunakan konsol

Di AWS Management Console, pada halaman detail untuk klaster, lihat status berbagi Cluster.

Untuk mengidentifikasi apakah klaster yang Anda miliki dibagikan dengan menggunakan AWS CLI

Gunakan perintah get-resource-policy. Jika ada kebijakan sumber daya untuk klaster, perintah akan menampilkan informasi tentang kebijakan tersebut.

Sebagai peserta, ketika sebuah cluster dibagikan dengan Anda, Anda biasanya harus menerima bagian tersebut. Selain itu, bidang Pemilik untuk klaster berisi akun pemilik cluster.

Tanggung jawab dan izin untuk kluster bersama

Izin untuk pemilik

Ketika Anda berbagi cluster yang Anda miliki dengan orang lain Akun AWS, peserta yang diizinkan untuk menggunakan klaster dapat membuat panel kontrol, kontrol routing, dan sumber daya lainnya di cluster.

Sebagai pemilik cluster, Anda bertanggung jawab untuk membuat, mengelola, dan menghapus cluster. Anda tidak dapat mengubah atau menghapus sumber daya yang dibuat oleh peserta, seperti kontrol perutean dan aturan keselamatan. Misalnya, Anda tidak dapat memperbarui kontrol perutean yang dibuat oleh peserta untuk mengubah status kontrol perutean.

Namun, Anda dapat melihat detail untuk kontrol perutean yang dibuat oleh peserta dalam klaster yang Anda miliki. Misalnya, Anda dapat melihat status kontrol perutean dengan memanggil operasi API kontrol perutean ARC, menggunakan atau. AWS Command Line Interface AWS SDKs

Jika Anda perlu mengubah sumber daya yang dibuat oleh peserta, mereka dapat mengatur peran di IAM dengan izin untuk mengakses sumber daya, dan menambahkan akun Anda ke peran tersebut.

Izin untuk peserta

Secara umum, peserta dapat membuat dan menggunakan panel kontrol, kontrol perutean, aturan keselamatan, dan pemeriksaan kesehatan yang mereka buat dalam cluster yang dibagikan dengan mereka. Mereka hanya dapat melihat, memodifikasi, atau menghapus sumber daya cluster di cluster bersama jika mereka memiliki sumber daya. Misalnya, peserta dapat membuat dan menghapus aturan keselamatan untuk panel kontrol yang telah mereka buat.

Pembatasan berikut berlaku untuk peserta:

- Peserta tidak dapat melihat, memodifikasi, atau menghapus panel kontrol yang dibuat oleh akun lain menggunakan cluster bersama.
- Peserta tidak dapat melihat, membuat, atau memodifikasi kontrol perutean, termasuk status kontrol perutean, untuk sumber daya yang dibuat dalam klaster bersama oleh akun lain.
- Peserta tidak dapat membuat, memodifikasi, atau melihat aturan keselamatan yang dibuat oleh akun lain dalam klaster bersama.
- Peserta tidak dapat menambahkan sumber daya di panel kontrol default di cluster bersama karena milik pemilik klaster.

Sebagaimana dicatat, peserta tidak dapat membuat kontrol perutean di panel kontrol default untuk cluster bersama, karena pemilik cluster memiliki panel kontrol default. Namun, pemilik klaster dapat membuat peran IAM lintas akun yang memberikan izin untuk mengakses panel kontrol default untuk cluster. Kemudian, pemilik dapat memberikan izin peserta untuk mengambil peran, sehingga peserta dapat mengakses panel kontrol default untuk menggunakannya namun pemilik telah menentukan melalui izin peran.

Biaya penagihan

Pemilik cluster di ARC ditagih untuk biaya yang terkait dengan cluster. Tidak ada biaya tambahan, untuk pemilik klaster atau peserta, untuk membuat sumber daya yang dihosting di cluster.

Untuk informasi dan contoh harga terperinci, lihat Harga Amazon Application Recovery Controller (ARC) dan gulir ke bawah ke Amazon Application Recovery Controller (ARC).

Kuota

Semua sumber daya yang dibuat dalam cluster bersama—termasuk sumber daya yang dibuat oleh semua peserta dengan akses ke cluster bersama—dihitung terhadap kuota yang berlaku untuk cluster dan sumber daya lainnya, seperti kontrol perutean. Jika akun yang berbagi sumber daya klaster memiliki kuota yang lebih tinggi daripada kuota pemilik klaster, kuota pemilik klaster lebih diutamakan daripada kuota untuk akun yang berbagi.

Untuk lebih memahami cara kerjanya, lihat contoh berikut. Untuk mengilustrasikan bagaimana kuota bekerja dengan berbagi sumber daya, untuk contoh ini, katakanlah pemilik klaster adalah Pemilik dan akun yang telah dibagikan oleh cluster adalah Peserta.

Kuota panel kontrol

Kuota diberlakukan untuk panel kontrol total Pemilik per cluster.

Misalnya, Owner memiliki kuota 50 untuk jumlah panel kontrol per cluster, dan memiliki 13 panel kontrol di cluster. Sekarang, katakan bahwa Peserta memiliki kuota yang ditetapkan menjadi 150. Dalam skenario ini, Peserta hanya dapat membuat hingga 37 panel kontrol (yaitu, 50-13) di cluster bersama.

Selain itu, jika akun lain yang berbagi cluster juga membuat panel kontrol, semua itu juga dihitung terhadap kuota keseluruhan cluster 50 panel kontrol.

Kuota kontrol perutean

Kontrol perutean memiliki beberapa kuota: kuota per panel kontrol, kuota per cluster, dan kuota per aturan keamanan. Kuota pemilik diutamakan untuk semua kuota ini.

Misalnya, Owner memiliki kuota 300 untuk jumlah kontrol routing per cluster, dan sudah memiliki 300 kontrol routing di cluster. Sekarang, katakan bahwa Peserta memiliki kuota ini ditetapkan menjadi 500. Dalam skenario ini, Peserta tidak dapat membuat kontrol perutean baru di cluster bersama.

Kuota aturan keamanan

Kuota diberlakukan untuk aturan keselamatan Pemilik per kuota panel kontrol.

Misalnya, pemilik memiliki kuota 20 untuk jumlah aturan keselamatan per panel kontrol dan Peserta memiliki kuota ini ditetapkan menjadi 80. Dalam skenario ini, karena batas bawah Pemilik diutamakan, Peserta hanya dapat membuat hingga 20 aturan keselamatan di panel kontrol di cluster bersama.

Untuk daftar kuota kontrol perutean, lihat. Kuota untuk kontrol perutean

Pencatatan dan pemantauan untuk kontrol perutean di Amazon Application Recovery Controller (ARC)

Anda dapat menggunakan AWS CloudTrail untuk memantau kontrol perutean di Amazon Application Recovery Controller (ARC), untuk menganalisis pola dan membantu memecahkan masalah.

Topik

- Mencatat panggilan ARC API menggunakan AWS CloudTrail

Mencatat panggilan ARC API menggunakan AWS CloudTrail

terintegrasi dengan AWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan di ARC. CloudTrail menangkap semua panggilan API untuk ARC sebagai peristiwa. Panggilan yang diambil termasuk panggilan dari konsol ARC dan panggilan kode ke operasi ARC API.

Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail acara secara terus menerus ke bucket Amazon S3, termasuk acara untuk ARC. Jika Anda tidak mengonfigurasi jejak, Anda masih dapat melihat peristiwa terbaru di CloudTrail konsol dalam Riwayat acara.

Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat untuk ARC, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan.

Untuk mempelajari selengkapnya CloudTrail, lihat Panduan AWS CloudTrail Pengguna.

Informasi ARC di CloudTrail

CloudTrail diaktifkan pada Akun AWS saat Anda membuat akun. Ketika aktivitas terjadi di ARC, aktivitas tersebut dicatat dalam suatu CloudTrail peristiwa bersama dengan peristiwa AWS layanan lainnya dalam riwayat Acara. Anda dapat melihat, mencari, dan mengunduh acara terbaru di situs Anda Akun AWS. Untuk informasi selengkapnya, lihat Bekerja dengan riwayat CloudTrail Acara.

Untuk catatan acara yang sedang berlangsung di Anda Akun AWS, termasuk acara untuk ARC, buat jejak. Jejak memungkinkan CloudTrail untuk mengirimkan file log ke bucket Amazon S3. Secara default, saat Anda membuat jejak di konsol, jejak tersebut berlaku untuk semua Wilayah AWS. Jejak mencatat peristiwa dari semua Wilayah di AWS partisi dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat mengonfigurasi AWS layanan lain untuk menganalisis lebih lanjut dan menindaklanjuti data peristiwa yang dikumpulkan dalam CloudTrail log. Untuk informasi selengkapnya, lihat berikut:

- Gambaran umum untuk membuat jejak
- CloudTrail layanan dan integrasi yang didukung
- Mengonfigurasi notifikasi Amazon SNS untuk CloudTrail
- Menerima file CloudTrail log dari beberapa wilayah dan Menerima file CloudTrail log dari beberapa akun

Semua tindakan ARC dicatat oleh CloudTrail dan didokumentasikan dalam Panduan Referensi API Kesiapan Pemulihan untuk Pengontrol Pemulihan Aplikasi Amazon, Panduan Referensi API Konfigurasi Kontrol Pemulihan untuk Pengontrol Pemulihan Aplikasi Amazon, dan Panduan Referensi API Kontrol Perutean untuk Pengontrol Pemulihan Aplikasi Amazon. Misalnya, panggilan ke`CreateCluster`, `UpdateRoutingControlState` dan `CreateRecoveryGroup` tindakan menghasilkan entri dalam file CloudTrail log.

Setiap entri peristiwa atau log berisi informasi tentang siapa yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan berikut ini:

- Apakah permintaan itu dibuat dengan kredenal pengguna root atau AWS Identity and Access Management (IAM).

- Apakah permintaan tersebut dibuat dengan kredensial keamanan sementara untuk satu peran atau pengguna gabungan.
- Apakah permintaan itu dibuat oleh AWS layanan lain.

Untuk informasi selengkapnya, lihat Elemen userIdentity CloudTrail .

Melihat peristiwa ARC dalam sejarah acara

CloudTrail memungkinkan Anda melihat peristiwa terbaru dalam riwayat Acara. Untuk melihat peristiwa permintaan ARC API, Anda harus memilih US West (Oregon) di pemilih Wilayah di bagian atas konsol. Untuk informasi selengkapnya, lihat Bekerja dengan riwayat CloudTrail Acara di Panduan AWS CloudTrail Pengguna.

Memahami entri file log ARC

Trail adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai file log ke bucket Amazon S3 yang Anda tentukan. CloudTrail file log berisi satu atau lebih entri log. Peristiwa mewakili permintaan tunggal dari sumber manapun dan mencakup informasi tentang tindakan yang diminta, tanggal dan waktu tindakan, parameter permintaan, dan sebagainya. CloudTrail file log bukanlah jejak tumpukan yang diurutkan dari panggilan API publik, sehingga file tersebut tidak muncul dalam urutan tertentu.

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan `CreateCluster` tindakan untuk mengkonfigurasi kontrol routing.

```
{
  "eventVersion": "1.08",
   "userIdentity": {
     "type": "IAMUser",
     "principalId": "A1B2C3D4E5F6G7EXAMPLE",
     "arn": "arn:aws:iam::111122223333:user/smithj",
     "accountId": "111122223333",
     "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
     "sessionContext": {
          "sessionIssuer": {
              "type": "Role",
              "principalId": "A1B2C3D4E5F6G7EXAMPLE",
              "arn": "arn:aws:iam::111122223333:role/smithj",
              "accountId": "111122223333",
              "userName": "smithj"
          },
```

```
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2021-06-30T04:44:41Z"
            }
        }
    },
    "eventTime": "2021-06-30T04:45:46Z",
    "eventSource": "route53-recovery-control-config.amazonaws.com",
    "eventName": "CreateCluster",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.50",
    "userAgent": "aws-cli/2.0.0 Python/3.8.2 Darwin/19.6.0 botocore/2.0.0dev7",
    "requestParameters": {
        "ClientToken": "12345abcdef-1234-5678-abcd-12345abcdef",
        "ClusterName": "XYZCluster"
    },
    "responseElements": {
        "Cluster": {
            "Arn": "arn:aws:route53-recovery-control::012345678901:cluster/abc123456-
aa11-bb22-cc33-abc123456",
            "ClusterArn": "arn:aws:route53-recovery-control::012345678901:cluster/
abc123456-aa11-bb22-cc33-abc123456",
            "Name": "XYZCluster",
            "Status": "PENDING"
        }
    },
    "requestID": "6090509a-5a97-4be6-8e6a-7d73example",
    "eventID": "9cab44ef-0777-41e6-838f-f249example",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "111122223333"
}
```

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan `UpdateRoutingControlState` tindakan untuk kontrol routing.

```
{
    "eventVersion": "1.08",
     "userIdentity": {
       "type": "AssumedRole",
```

```
      "principalId": "A1B2C3D4E5F6G7EXAMPLE",
      "arn": "arn:aws:sts::111122223333:assumed-role/admin/smithj",
      "accountId": "111122223333",
      "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
      "sessionContext": {
          "sessionIssuer": {
              "type": "Role",
              "principalId": "A1B2C3D4E5F6G7EXAMPLE",
              "arn": "arn:aws:iam::111122223333:role/admin",
              "accountId": "111122223333",
              "userName": "admin"
          },
          "webIdFederationData": {},
          "attributes": {
              "mfaAuthenticated": "false",
              "creationDate": "2021-06-30T04:44:41Z"
          }
      }
  },
  "eventTime": "2021-06-30T04:45:46Z",
  "eventSource": "route53-recovery-control-config.amazonaws.com",
  "eventName": "UpdateRoutingControl",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.50",
  "userAgent": "aws-cli/2.0.0 Python/3.8.2 Darwin/19.6.0 botocore/2.0.0dev7",
  "requestParameters": {
      "RoutingControlName": "XYZRoutingControl3",
      "RoutingControlArn": "arn:aws:route53-recovery-
control::012345678:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/routingcontrol/
abcdefg1234567"
  },
  "responseElements": {
      "RoutingControl": {
          "ControlPanelArn": "arn:aws:route53-recovery-
control::012345678:controlpanel/0123456bbbbbbb0123456bbbbbb0123456",
          "Name": "XYZRoutingControl3",
          "Status": "DEPLOYED",
          "RoutingControlArn": "arn:aws:route53-recovery-
control::012345678:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/routingcontrol/
abcdefg1234567"
      }
  },
  "requestID": "6090509a-5a97-4be6-8e6a-7d73example",
  "eventID": "9cab44ef-0777-41e6-838f-f249example",
```

```
  "readOnly": false,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "eventCategory": "Management",
  "recipientAccountId": "111122223333"
}
```

Identity and Access Management untuk kontrol routing di

AWS Identity and Access Management (IAM) adalah Layanan AWS yang membantu administrator mengontrol akses ke AWS sumber daya dengan aman. Administrator IAM mengontrol siapa yang dapat diautentikasi (masuk) dan diberi wewenang (memiliki izin) untuk menggunakan sumber daya ARC. IAM adalah Layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

Daftar Isi

Bagaimana kontrol perutean di Amazon Application Recovery Controller (ARC) bekerja dengan IAM

Sebelum Anda menggunakan IAM untuk mengelola akses ke kontrol perutean di Amazon Application Recovery Controller (ARC), pelajari fitur IAM apa yang tersedia untuk digunakan dengan kontrol perutean.

Fitur IAM yang dapat Anda gunakan dengan kontrol perutean di Amazon Application Recovery Controller (ARC)

Fitur IAM	Dukungan kontrol perutean
Kebijakan berbasis identitas	Ya
Kebijakan berbasis sumber daya	Tidak
Tindakan kebijakan	Ya
Sumber daya kebijakan	Ya

Fitur IAM	Dukungan kontrol perutean
Kunci kondisi kebijakan	Ya
ACLs	Tidak
ABAC (tanda dalam kebijakan)	Parsial
Kredensial sementara	Ya
Izin principal	Ya
Peran layanan	Tidak
Peran terkait layanan	Tidak

Untuk mendapatkan tampilan tingkat tinggi dan keseluruhan tentang cara kerja AWS layanan dengan sebagian besar fitur IAM, lihat AWS layanan yang bekerja dengan IAM di Panduan Pengguna IAM.

Kebijakan berbasis identitas untuk ARC

Mendukung kebijakan berbasis identitas: Ya

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, seperti pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana, dan berdasarkan kondisi seperti apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Tentukan izin IAM kustom dengan kebijakan terkelola pelanggan dalam Panduan Pengguna IAM.

Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan secara spesifik apakah tindakan dan sumber daya diizinkan atau ditolak, serta kondisi yang menjadi dasar dikabulkan atau ditolaknya tindakan tersebut. Anda tidak dapat menentukan secara spesifik prinsipal dalam sebuah kebijakan berbasis identitas karena prinsipal berlaku bagi pengguna atau peran yang melekat kepadanya. Untuk mempelajari semua elemen yang dapat Anda gunakan dalam kebijakan JSON, lihat Referensi elemen kebijakan JSON IAM dalam Panduan Pengguna IAM.

Untuk melihat contoh kebijakan berbasis identitas ARC untuk kontrol perutean, lihat. Contoh kebijakan berbasis identitas untuk kontrol perutean di ARC

Kebijakan berbasis sumber daya dalam kontrol perutean

Mendukung kebijakan berbasis sumber daya: Tidak

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu.

Tindakan kebijakan untuk kontrol perutean

Mendukung tindakan kebijakan: Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen `Action` dari kebijakan JSON menjelaskan tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam sebuah kebijakan. Tindakan kebijakan biasanya memiliki nama yang sama dengan operasi AWS API terkait. Ada beberapa pengecualian, misalnya tindakan hanya izin yang tidak memiliki operasi API yang cocok. Ada juga beberapa operasi yang memerlukan beberapa tindakan dalam suatu kebijakan. Tindakan tambahan ini disebut tindakan dependen.

Sertakan tindakan dalam kebijakan untuk memberikan izin untuk melakukan operasi terkait.

Untuk melihat daftar tindakan ARC untuk kontrol perutean, lihat Tindakan yang ditentukan oleh Amazon Route 53 Recovery Controls and Actions yang ditentukan oleh Amazon Route 53 Recovery Cluster dalam Referensi Otorisasi Layanan.

Tindakan kebijakan di ARC untuk kontrol perutean menggunakan awalan berikut sebelum tindakan, bergantung pada API yang Anda kerjakan:

```
route53-recovery-control-config
route53-recovery-cluster
```

Untuk menetapkan secara spesifik beberapa tindakan dalam satu pernyataan, pisahkan tindakan tersebut dengan koma. Sebagai contoh, Anda dapat melakukan hal berikut:

```
"Action": [
      "route53-recovery-control-config:action1",
      "route53-recovery-control-config:action2"
```

```
        ]
```

Anda dapat menentukan beberapa tindakan menggunakan wildcard (*). Sebagai contoh, untuk menentukan semua tindakan yang dimulai dengan kata `Describe`, sertakan tindakan berikut:

```
"Action": "route53-recovery-control-config:Describe*"
```

Untuk melihat contoh kebijakan berbasis identitas ARC untuk kontrol perutean, lihat. Contoh kebijakan berbasis identitas untuk kontrol perutean di ARC

Sumber daya kebijakan untuk ARC

Mendukung sumber daya kebijakan: Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen kebijakan JSON `Resource` menentukan objek yang menjadi target penerapan tindakan. Pernyataan harus menyertakan elemen `Resource` atau `NotResource`. Praktik terbaiknya, tentukan sumber daya menggunakan Amazon Resource Name (ARN). Anda dapat melakukan ini untuk tindakan yang mendukung jenis sumber daya tertentu, yang dikenal sebagai izin tingkat sumber daya.

Untuk tindakan yang tidak mendukung izin di tingkat sumber daya, misalnya operasi pencantuman, gunakan wildcard (*) untuk menunjukkan bahwa pernyataan tersebut berlaku untuk semua sumber daya.

```
"Resource": "*"
```

Dalam Referensi Otorisasi Layanan, Anda dapat melihat informasi berikut yang terkait dengan ARC:

Untuk melihat daftar jenis sumber daya dan mereka ARNs, dan tindakan yang dapat Anda tentukan dengan ARN dari setiap sumber daya, lihat topik berikut di Referensi Otorisasi Layanan:

- Tindakan yang ditentukan oleh Amazon Route 53 Recovery Controls
- Tindakan yang ditentukan oleh Amazon Route 53 Recovery Cluster.

Untuk melihat contoh kebijakan berbasis identitas ARC untuk kontrol perutean, lihat. Contoh kebijakan berbasis identitas untuk kontrol perutean di ARC

Kunci kondisi kebijakan untuk ARC

Mendukung kunci kondisi kebijakan khusus layanan: Yes

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen `Condition` (atau blok `Condition`) akan memungkinkan Anda menentukan kondisi yang menjadi dasar suatu pernyataan berlaku. Elemen `Condition` bersifat opsional. Anda dapat membuat ekspresi bersyarat yang menggunakan operator kondisi, misalnya sama dengan atau kurang dari, untuk mencocokkan kondisi dalam kebijakan dengan nilai-nilai yang diminta.

Jika Anda menentukan beberapa elemen `Condition` dalam sebuah pernyataan, atau beberapa kunci dalam elemen `Condition` tunggal, maka AWS akan mengevaluasinya menggunakan operasi `AND` logis. Jika Anda menentukan beberapa nilai untuk satu kunci kondisi, AWS mengevaluasi kondisi menggunakan `OR` operasi logis. Semua kondisi harus dipenuhi sebelum izin pernyataan diberikan.

Anda juga dapat menggunakan variabel placeholder saat menentukan kondisi. Sebagai contoh, Anda dapat memberikan izin kepada pengguna IAM untuk mengakses sumber daya hanya jika izin tersebut mempunyai tanda yang sesuai dengan nama pengguna IAM mereka. Untuk informasi selengkapnya, lihat Elemen kebijakan IAM: variabel dan tanda dalam Panduan Pengguna IAM.

AWS mendukung kunci kondisi global dan kunci kondisi khusus layanan. Untuk melihat semua kunci kondisi AWS global, lihat kunci konteks kondisi AWS global di Panduan Pengguna IAM.

Untuk melihat daftar kunci kondisi ARC untuk kontrol perutean, lihat topik berikut di Referensi Otorisasi Layanan:

- Kunci kondisi untuk Amazon Route 53 Recovery Controls
- Kunci kondisi untuk Amazon Route 53 Recovery Cluster

Untuk melihat tindakan dan sumber daya yang dapat Anda gunakan dengan kunci kondisi, lihat topik berikut di Referensi Otorisasi Layanan:

- Untuk melihat daftar jenis sumber daya dan jenisnya ARNs, lihat Tindakan yang ditentukan oleh Amazon Route 53 Recovery Controls and Actions yang ditentukan oleh Amazon Route 53 Recovery Cluster.

- Untuk melihat daftar tindakan yang dapat Anda tentukan dengan ARN setiap sumber daya, lihat Sumber daya yang ditentukan oleh Amazon Route 53 Recovery Controls and Resources yang ditentukan oleh Amazon Route 53 Recovery Cluster.

Untuk melihat contoh kebijakan berbasis identitas ARC untuk kontrol perutean, lihat Contoh kebijakan berbasis identitas untuk kontrol perutean di ARC

Daftar kontrol akses (ACLs) di ARC

Mendukung ACLs: Tidak

Access control lists (ACLs) mengontrol prinsipal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACLs mirip dengan kebijakan berbasis sumber daya, meskipun mereka tidak menggunakan format dokumen kebijakan JSON.

Kontrol akses berbasis atribut (ABAC) dengan ARC

Mendukung ABAC (tag dalam kebijakan): Sebagian

Kontrol akses berbasis atribut (ABAC) adalah strategi otorisasi yang menentukan izin berdasarkan atribut. Dalam AWS, atribut ini disebut tag. Anda dapat melampirkan tag ke entitas IAM (pengguna atau peran) dan ke banyak AWS sumber daya. Penandaan ke entitas dan sumber daya adalah langkah pertama dari ABAC. Kemudian rancanglah kebijakan ABAC untuk mengizinkan operasi ketika tanda milik prinsipal cocok dengan tanda yang ada di sumber daya yang ingin diakses.

ABAC sangat berguna di lingkungan yang berkembang dengan cepat dan berguna di situasi saat manajemen kebijakan menjadi rumit.

Untuk mengendalikan akses berdasarkan tanda, berikan informasi tentang tanda di elemen kondisi dari kebijakan menggunakan kunci kondisi `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`*, atau `aws:TagKeys`.

Jika sebuah layanan mendukung ketiga kunci kondisi untuk setiap jenis sumber daya, nilainya adalah Ya untuk layanan tersebut. Jika suatu layanan mendukung ketiga kunci kondisi untuk hanya beberapa jenis sumber daya, nilainya adalah Parsial.

Untuk informasi selengkapnya tentang ABAC, lihat Tentukan izin dengan otorisasi ABAC dalam Panduan Pengguna IAM. Untuk melihat tutorial yang menguraikan langkah-langkah pengaturan ABAC, lihat Menggunakan kontrol akses berbasis atribut (ABAC) dalam Panduan Pengguna IAM.

Kontrol perutean ARC mencakup dukungan berikut untuk ABAC:

- Recovery Control Config mendukung ABAC.
- Cluster Pemulihan tidak mendukung ABAC.

Menggunakan kredensi sementara dengan ARC

Mendukung kredensial sementara: Ya

Beberapa Layanan AWS tidak berfungsi saat Anda masuk menggunakan kredensi sementara. Untuk informasi tambahan, termasuk yang Layanan AWS bekerja dengan kredensi sementara, lihat Layanan AWS yang bekerja dengan IAM di Panduan Pengguna IAM.

Anda menggunakan kredensi sementara jika Anda masuk AWS Management Console menggunakan metode apa pun kecuali nama pengguna dan kata sandi. Misalnya, ketika Anda mengakses AWS menggunakan tautan masuk tunggal (SSO) perusahaan Anda, proses tersebut secara otomatis membuat kredensi sementara. Anda juga akan secara otomatis membuat kredensial sementara ketika Anda masuk ke konsol sebagai seorang pengguna lalu beralih peran. Untuk informasi selengkapnya tentang peralihan peran, lihat Beralih dari pengguna ke peran IAM (konsol) dalam Panduan Pengguna IAM.

Anda dapat membuat kredenal sementara secara manual menggunakan API AWS CLI atau AWS . Anda kemudian dapat menggunakan kredensi sementara tersebut untuk mengakses. AWS AWS merekomendasikan agar Anda secara dinamis menghasilkan kredensi sementara alih-alih menggunakan kunci akses jangka panjang. Untuk informasi selengkapnya, lihat Kredensial keamanan sementara di IAM.

Izin utama lintas layanan untuk ARC

Mendukung sesi akses maju (FAS): Ya

Saat Anda menggunakan entitas IAM (pengguna atau peran) untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Kebijakan memberikan izin kepada principal. Saat Anda menggunakan beberapa layanan, Anda mungkin melakukan tindakan yang kemudian memicu tindakan lain di layanan yang berbeda. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut.

Untuk melihat apakah suatu tindakan memerlukan tindakan dependen tambahan dalam kebijakan, lihat topik berikut di Referensi Otorisasi Layanan:

- Cluster Pemulihan Amazon Route 53

- Amazon Route 53 Kontrol Pemulihan

Peran layanan untuk ARC

Mendukung peran layanan: Tidak

Peran layanan adalah peran IAM yang diambil oleh sebuah layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, mengubah, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat Buat sebuah peran untuk mendelegasikan izin ke Layanan AWS dalam Panduan pengguna IAM.

Peran terkait layanan untuk ARC

Mendukung peran terkait layanan:

Peran terkait layanan adalah jenis peran layanan yang ditautkan ke layanan. AWS Layanan tersebut dapat menjalankan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di AWS akun Anda dan dimiliki oleh layanan. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.

Kontrol perutean tidak menggunakan peran terkait layanan.

Contoh kebijakan berbasis identitas untuk kontrol perutean di ARC

Secara default, pengguna dan peran tidak memiliki izin untuk membuat atau memodifikasi sumber daya ARC. Mereka juga tidak dapat melakukan tugas dengan menggunakan AWS Management Console, AWS Command Line Interface (AWS CLI), atau AWS API. Untuk memberikan izin kepada pengguna untuk melakukan tindakan di sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM. Administrator kemudian dapat menambahkan kebijakan IAM ke peran, dan pengguna dapat mengambil peran.

Untuk mempelajari cara membuat kebijakan berbasis identitas IAM dengan menggunakan contoh dokumen kebijakan JSON ini, lihat Membuat kebijakan IAM (konsol) di Panduan Pengguna IAM.

Untuk detail tentang tindakan dan jenis sumber daya yang ditentukan oleh ARC, termasuk format ARNs untuk setiap jenis sumber daya, lihat Kunci tindakan, sumber daya, dan kondisi untuk Amazon Application Recovery Controller (ARC) di Referensi Otorisasi Layanan.

Topik

- Praktik terbaik kebijakan

- Contoh: Akses konsol ARC untuk kontrol perutean
- Contoh: Tindakan ARC API untuk konfigurasi kontrol routing

Praktik terbaik kebijakan

Kebijakan berbasis identitas menentukan apakah seseorang dapat membuat, mengakses, atau menghapus sumber daya ARC di akun Anda. Tindakan ini membuat Akun AWS Anda dikenai biaya. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

- Mulailah dengan kebijakan AWS terkelola dan beralih ke izin hak istimewa paling sedikit — Untuk mulai memberikan izin kepada pengguna dan beban kerja Anda, gunakan kebijakan AWS terkelola yang memberikan izin untuk banyak kasus penggunaan umum. Mereka tersedia di Anda Akun AWS. Kami menyarankan Anda mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola AWS pelanggan yang khusus untuk kasus penggunaan Anda. Untuk informasi selengkapnya, lihat Kebijakan yang dikelola AWS atau Kebijakan yang dikelola AWS untuk fungsi tugas dalam Panduan Pengguna IAM.
- Menerapkan izin dengan hak akses paling rendah – Ketika Anda menetapkan izin dengan kebijakan IAM, hanya berikan izin yang diperlukan untuk melakukan tugas. Anda melakukannya dengan mendefinisikan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi tertentu, yang juga dikenal sebagai izin dengan hak akses paling rendah. Untuk informasi selengkapnya tentang cara menggunakan IAM untuk mengajukan izin, lihat Kebijakan dan izin dalam IAM dalam Panduan Pengguna IAM.
- Gunakan kondisi dalam kebijakan IAM untuk membatasi akses lebih lanjut – Anda dapat menambahkan suatu kondisi ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. Sebagai contoh, Anda dapat menulis kondisi kebijakan untuk menentukan bahwa semua permintaan harus dikirim menggunakan SSL. Anda juga dapat menggunakan ketentuan untuk memberikan akses ke tindakan layanan jika digunakan melalui yang spesifik Layanan AWS, seperti AWS CloudFormation. Untuk informasi selengkapnya, lihat Elemen kebijakan JSON IAM: Kondisi dalam Panduan Pengguna IAM.
- Gunakan IAM Access Analyzer untuk memvalidasi kebijakan IAM Anda untuk memastikan izin yang aman dan fungsional – IAM Access Analyzer memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan tersebut mematuhi bahasa kebijakan IAM (JSON) dan praktik terbaik IAM. IAM Access Analyzer menyediakan lebih dari 100 pemeriksaan kebijakan dan rekomendasi yang dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang aman dan fungsional. Untuk informasi selengkapnya, lihat Validasi kebijakan dengan IAM Access Analyzer dalam Panduan Pengguna IAM.

- Memerlukan otentikasi multi-faktor (MFA) - Jika Anda memiliki skenario yang mengharuskan pengguna IAM atau pengguna root di Anda, Akun AWS aktifkan MFA untuk keamanan tambahan. Untuk meminta MFA ketika operasi API dipanggil, tambahkan kondisi MFA pada kebijakan Anda. Untuk informasi selengkapnya, lihat Amankan akses API dengan MFA dalam Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang praktik terbaik dalam IAM, lihat Praktik terbaik keamanan di IAM dalam Panduan Pengguna IAM.

Contoh: Akses konsol ARC untuk kontrol perutean

Untuk mengakses konsol Amazon Application Recovery Controller (ARC), Anda harus memiliki seperangkat izin minimum. Izin ini harus memungkinkan Anda untuk membuat daftar dan melihat detail tentang sumber daya ARC di Anda Akun AWS. Jika Anda membuat kebijakan berbasis identitas yang lebih ketat daripada izin minimum yang diperlukan, konsol tidak akan berfungsi sebagaimana mestinya untuk entitas (pengguna atau peran) dengan kebijakan tersebut.

Anda tidak perlu mengizinkan izin konsol minimum untuk pengguna yang melakukan panggilan hanya ke AWS CLI atau AWS API. Sebagai gantinya, izinkan akses hanya ke tindakan yang sesuai dengan operasi API yang coba mereka lakukan.

Untuk memastikan bahwa pengguna dan peran masih dapat menggunakan konsol ARC saat Anda mengizinkan akses hanya ke operasi API tertentu, lampirkan juga kebijakan `ReadOnly` AWS terkelola untuk ARC ke entitas. Untuk informasi selengkapnya, lihat halaman kebijakan terkelola ARC ARC atau Menambahkan izin ke pengguna di Panduan Pengguna IAM.

Untuk memberi pengguna akses penuh untuk menggunakan fitur kontrol perutean ARC melalui konsol, lampirkan kebijakan seperti berikut ini kepada pengguna, untuk memberikan izin penuh kepada pengguna untuk mengonfigurasi sumber daya dan operasi kontrol perutean ARC:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                   "route53-recovery-cluster:GetRoutingControlState",
                   "route53-recovery-cluster:UpdateRoutingControlState",
                   "route53-recovery-cluster:UpdateRoutingControlStates",
                   "route53-recovery-control-config:CreateCluster",
```

```
                    "route53-recovery-control-config:CreateControlPanel",
                    "route53-recovery-control-config:CreateRoutingControl",
                    "route53-recovery-control-config:CreateSafetyRule",
                    "route53-recovery-control-config:DeleteCluster",
                    "route53-recovery-control-config:DeleteControlPanel",
                    "route53-recovery-control-config:DeleteRoutingControl",
                    "route53-recovery-control-config:DeleteSafetyRule",
                    "route53-recovery-control-config:DescribeCluster",
                    "route53-recovery-control-config:DescribeControlPanel",
                    "route53-recovery-control-config:DescribeSafetyRule",
                    "route53-recovery-control-config:DescribeRoutingControl",
                    "route53-recovery-control-config:ListAssociatedRoute53HealthChecks",
                    "route53-recovery-control-config:ListClusters",
                    "route53-recovery-control-config:ListControlPanels",
                    "route53-recovery-control-config:ListRoutingControls",
                    "route53-recovery-control-config:ListSafetyRules",
                    "route53-recovery-control-config:UpdateControlPanel",
                    "route53-recovery-control-config:UpdateRoutingControl",
                    "route53-recovery-control-config:UpdateSafetyRule"
                ],
                "Resource": "*"
            },
                    {
                "Effect": "Allow",
                "Action": [
                        "route53:GetHealthCheck",
                        "route53:CreateHealthCheck",
                        "route53:DeleteHealthCheck",
                        "route53:ChangeTagsForResource"
                ],
                "Resource": "*"
            }
        ]
}
```

Contoh: Tindakan ARC API untuk konfigurasi kontrol routing

Untuk memastikan bahwa pengguna dapat menggunakan tindakan ARC API untuk bekerja dengan konfigurasi kontrol perutean ARC, lampirkan kebijakan yang sesuai dengan operasi API yang perlu dikerjakan pengguna, seperti yang dijelaskan di bawah ini.

Untuk bekerja dengan operasi API untuk konfigurasi kontrol pemulihan, lampirkan kebijakan seperti berikut ini kepada pengguna:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                   "route53-recovery-control-config:CreateCluster",
                   "route53-recovery-control-config:CreateControlPanel",
                   "route53-recovery-control-config:CreateRoutingControl",
                   "route53-recovery-control-config:CreateSafetyRule",
                   "route53-recovery-control-config:DeleteCluster",
                   "route53-recovery-control-config:DeleteControlPanel",
                   "route53-recovery-control-config:DeleteRoutingControl",
                   "route53-recovery-control-config:DeleteSafetyRule",
                   "route53-recovery-control-config:DescribeCluster",
                   "route53-recovery-control-config:DescribeControlPanel",
                   "route53-recovery-control-config:DescribeSafetyRule",
                   "route53-recovery-control-config:DescribeRoutingControl",
                   "route53-recovery-control-config:GetResourcePolicy",
                   "route53-recovery-control-config:ListAssociatedRoute53HealthChecks",
                   "route53-recovery-control-config:ListClusters",
                   "route53-recovery-control-config:ListControlPanels",
                   "route53-recovery-control-config:ListRoutingControls",
                   "route53-recovery-control-config:ListSafetyRules",
                   "route53-recovery-control-config:ListTagsForResource",
                   "route53-recovery-control-config:UpdateControlPanel",
                   "route53-recovery-control-config:UpdateRoutingControl",
                   "route53-recovery-control-config:UpdateSafetyRule",
                   "route53-recovery-control-config:TagResource",
                   "route53-recovery-control-config:UntagResource"
            ],
            "Resource": "*"
        }
    ]
}
```

Untuk melakukan tugas dalam kontrol perutean ARC dengan API bidang data cluster pemulihan, misalnya, memperbarui status kontrol perutean agar gagal selama peristiwa bencana, Anda dapat melampirkan kebijakan ARC IAM seperti berikut ini ke pengguna IAM Anda.

`AllowSafetyRuleOverride`Boolean memberikan izin untuk mengganti aturan keselamatan yang telah Anda konfigurasikan sebagai pengaman untuk kontrol perutean. Izin ini mungkin diperlukan

dalam skenario “pecah kaca” untuk melewati perlindungan dalam bencana atau skenario failover mendesak lainnya. Misalnya, operator mungkin perlu gagal dengan cepat untuk pemulihan bencana, dan satu atau lebih aturan keselamatan mungkin secara tak terduga mencegah pembaruan status kontrol perutean yang diperlukan untuk mengalihkan lalu lintas. Izin ini memungkinkan operator menentukan aturan keselamatan yang akan diganti saat melakukan panggilan API untuk memperbarui status kontrol perutean. Untuk informasi selengkapnya, lihat Mengesampingkan aturan keselamatan untuk mengubah rute lalu lintas.

Jika Anda ingin mengizinkan operator menggunakan API bidang data cluster pemulihan tetapi mencegah aturan keselamatan yang berlebihan, Anda dapat melampirkan kebijakan seperti berikut ini, dengan `AllowSafetyRuleOverrides` boolean ke. `false` Untuk memungkinkan operator mengganti aturan keselamatan, setel `AllowSafetyRuleOverrides` boolean ke. `true`

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "route53-recovery-cluster:GetRoutingControlState",
                "route53-recovery-cluster:ListRoutingControls"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "route53-recovery-cluster:UpdateRoutingControlStates",
                "route53-recovery-cluster:UpdateRoutingControlState"
            ],
            "Resource": "*",
            "Condition": {
                "Bool": {
                    "route53-recovery-cluster:AllowSafetyRulesOverrides": "false"
                }
            }
        }
    ]
}
```

AWS kebijakan terkelola untuk kontrol perutean di Amazon Application Recovery Controller (ARC)

Kebijakan AWS terkelola adalah kebijakan mandiri yang dibuat dan dikelola oleh AWS. AWS Kebijakan terkelola dirancang untuk memberikan izin bagi banyak kasus penggunaan umum sehingga Anda dapat mulai menetapkan izin kepada pengguna, grup, dan peran.

Perlu diingat bahwa kebijakan AWS terkelola mungkin tidak memberikan izin hak istimewa paling sedikit untuk kasus penggunaan spesifik Anda karena tersedia untuk digunakan semua pelanggan. AWS Kami menyarankan Anda untuk mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola pelanggan yang khusus untuk kasus penggunaan Anda.

Anda tidak dapat mengubah izin yang ditentukan dalam kebijakan AWS terkelola. Jika AWS memperbarui izin yang ditentukan dalam kebijakan AWS terkelola, pembaruan akan memengaruhi semua identitas utama (pengguna, grup, dan peran) yang dilampirkan kebijakan tersebut. AWS kemungkinan besar akan memperbarui kebijakan AWS terkelola saat baru Layanan AWS diluncurkan atau operasi API baru tersedia untuk layanan yang ada.

Untuk informasi selengkapnya, lihat Kebijakan terkelola AWS dalam Panduan Pengguna IAM.

AWS kebijakan terkelola: AmazonRoute 53 RecoveryControlConfigFullAccess

Anda dapat melampirkan `AmazonRoute53RecoveryControlConfigFullAccess` ke entitas IAM Anda. Kebijakan ini memberikan akses penuh ke tindakan untuk bekerja dengan konfigurasi kontrol pemulihan di ARC. Lampirkan ke pengguna IAM dan prinsipal lain yang membutuhkan akses penuh ke tindakan konfigurasi kontrol pemulihan.

Sesuai kebijaksanaan Anda, Anda dapat menambahkan akses ke tindakan Amazon Route 53 tambahan untuk memungkinkan pengguna membuat pemeriksaan kesehatan untuk kontrol perutean. Misalnya, Anda dapat mengizinkan izin untuk satu atau beberapa tindakan berikut:`route53:GetHealthCheck`,`route53:CreateHealthCheck`,`route53:DeleteHealthCheck`, dan`route53:ChangeTagsForResource`.

Untuk melihat izin kebijakan ini, lihat AmazonRoute53 RecoveryControlConfigFullAccess di Referensi Kebijakan AWS Terkelola.

AWS kebijakan terkelola: AmazonRoute 53 RecoveryControlConfigReadOnlyAccess

Anda dapat melampirkan `AmazonRoute53RecoveryControlConfigReadOnlyAccess` ke entitas IAM Anda. Ini berguna bagi pengguna yang perlu melihat kontrol perutean dan konfigurasi aturan keselamatan. Kebijakan ini memberikan akses hanya-baca ke tindakan untuk bekerja dengan

konfigurasi kontrol pemulihan di ARC. Pengguna ini tidak dapat membuat, memperbarui, atau menghapus sumber daya kontrol pemulihan.

Untuk melihat izin kebijakan ini, lihat AmazonRoute53 RecoveryControlConfigReadOnlyAccess di Referensi Kebijakan AWS Terkelola.

AWS kebijakan terkelola: AmazonRoute 53 RecoveryClusterFullAccess

Anda dapat melampirkan `AmazonRoute53RecoveryClusterFullAccess` ke entitas IAM Anda. Kebijakan ini memberikan akses penuh ke tindakan untuk bekerja dengan bidang data klaster di ARC. Lampirkan ke pengguna IAM dan prinsipal lain yang membutuhkan akses penuh untuk memperbarui dan mengambil status kontrol perutean.

Untuk melihat izin kebijakan ini, lihat AmazonRoute53 RecoveryClusterFullAccess di Referensi Kebijakan AWS Terkelola.

AWS kebijakan terkelola: AmazonRoute 53 RecoveryClusterReadOnlyAccess

Anda dapat melampirkan `AmazonRoute53RecoveryClusterReadOnlyAccess` ke entitas IAM Anda. Kebijakan ini memberikan akses hanya-baca ke bidang data klaster di ARC. Pengguna ini dapat mengambil status kontrol perutean tetapi tidak dapat memperbaruinya.

Untuk melihat izin kebijakan ini, lihat AmazonRoute53 RecoveryClusterReadOnlyAccess di Referensi Kebijakan AWS Terkelola.

Pembaruan untuk kebijakan AWS terkelola untuk kontrol perutean

Untuk detail tentang pembaruan kebijakan AWS terkelola untuk kontrol perutean di ARC sejak layanan ini mulai melacak perubahan ini, lihatPembaruan kebijakan AWS terkelola untuk Amazon Application Recovery Controller (ARC). Untuk peringatan otomatis tentang perubahan pada halaman ini, berlangganan umpan RSS di halaman riwayat Dokumen ARC.

Kuota untuk kontrol perutean

Kontrol perutean di Amazon Application Recovery Controller (ARC) tunduk pada kuota berikut (sebelumnya disebut sebagai batas).

Entitas	Kuota
Jumlah cluster per akun	2

Entitas	Kuota
Jumlah panel kontrol per cluster	50
Jumlah kontrol routing per panel kontrol	100
Jumlah total kontrol routing (di semua panel kontrol) per cluster	300
Jumlah aturan keselamatan per panel kontrol	20
Jumlah kontrol perutean per panggilan UpdateRoutingControlStatesoperasi	10
Jumlah panggilan API yang bermutasi ke titik akhir cluster, per detik	3

Pemeriksaan kesiapan di ARC

Dengan pemeriksaan kesiapan di Amazon Application Recovery Controller (ARC), Anda dapat memperoleh wawasan tentang apakah aplikasi dan sumber daya Anda siap untuk pemulihan. Setelah Anda memodelkan AWS aplikasi Anda di ARC dan membuat pemeriksaan kesiapan, pemeriksaan terus memantau informasi tentang aplikasi Anda, seperti kuota AWS sumber daya, kapasitas, dan kebijakan perutean jaringan. Kemudian, Anda dapat memilih untuk diberi tahu tentang perubahan yang akan memengaruhi kemampuan Anda untuk gagal ke replika aplikasi Anda, untuk pulih dari suatu peristiwa. Pemeriksaan kesiapan membantu memastikan, secara berkelanjutan, bahwa Anda dapat mempertahankan aplikasi Multi-wilayah Anda dalam keadaan yang diskalakan dan dikonfigurasi untuk menangani lalu lintas failover.

Bab ini menjelaskan cara memodelkan aplikasi Anda di ARC untuk menyiapkan struktur yang memungkinkan pemeriksaan kesiapan bekerja, dengan membuat grup pemulihan dan sel yang menjelaskan aplikasi Anda. Kemudian, Anda dapat mengikuti langkah-langkah untuk menambahkan pemeriksaan kesiapan dan cakupan kesiapan sehingga ARC dapat mengaudit kesiapan untuk aplikasi Anda.

Setelah Anda membuat pemeriksaan kesiapan, Anda dapat memantau status kesiapan sumber daya Anda. Pemeriksaan kesiapan membantu Anda memastikan bahwa replika aplikasi siaga dan sumber dayanya sesuai dengan replika produksi Anda secara berkelanjutan, yang mencerminkan kapasitas, kebijakan perutean, dan detail konfigurasi lain dari aplikasi produksi Anda. Jika replika tidak cocok, Anda dapat menambahkan kapasitas atau mengubah konfigurasi sehingga replika aplikasi Anda disejajarkan lagi.

> **Important**
>
> Pemeriksaan kesiapan paling berguna untuk memverifikasi, secara berkelanjutan, bahwa konfigurasi replika aplikasi dan status runtime diselaraskan. Pemeriksaan kesiapan tidak boleh digunakan untuk menunjukkan apakah replika produksi Anda sehat, Anda juga tidak boleh mengandalkan pemeriksaan kesiapan sebagai pemicu utama kegagalan selama peristiwa bencana.

Apa itu pemeriksaan kesiapan di Amazon Application Recovery Controller (ARC)?

Pemeriksaan kesiapan di ARC terus menerus (pada interval satu menit) mengaudit ketidakcocokan dalam kapasitas yang AWS disediakan, kuota layanan, batas throttle, dan perbedaan konfigurasi dan versi untuk sumber daya yang disertakan dalam pemeriksaan. Pemeriksaan kesiapan dapat memberi tahu Anda tentang perbedaan ini sehingga Anda dapat memastikan bahwa setiap replika memiliki pengaturan konfigurasi yang sama dan status runtime yang sama. Meskipun pemeriksaan kesiapan memastikan bahwa kapasitas yang dikonfigurasi di seluruh replika konsisten, Anda tidak boleh mengharapkan mereka memutuskan atas nama Anda berapa kapasitas replika Anda seharusnya. Misalnya, Anda harus memahami persyaratan aplikasi sehingga Anda mengukur grup Auto Scaling Anda dengan kapasitas buffer yang cukup di setiap replika untuk mengelola jika sel lain tidak tersedia.

Untuk kuota, ketika ARC mendeteksi ketidakcocokan dengan pemeriksaan kesiapan, ARC dapat mengambil langkah-langkah untuk menyelaraskan kuota replika dengan meningkatkan kuota yang lebih rendah agar sesuai dengan kuota yang lebih tinggi. Saat kuota cocok, status pemeriksaan kesiapan ditampilkan. `READY` (Perhatikan bahwa ini bukan proses pembaruan langsung, dan total waktu tergantung pada jenis sumber daya tertentu dan faktor lainnya.)

Langkah pertama adalah menyiapkan pemeriksaan kesiapan untuk membuat grup pemulihan yang mewakili aplikasi Anda. Setiap grup pemulihan menyertakan sel untuk setiap unit penahanan

kegagalan individu atau replika aplikasi Anda. Selanjutnya, Anda membuat kumpulan sumber daya untuk setiap jenis sumber daya dalam aplikasi Anda, dan mengaitkan pemeriksaan kesiapan dengan kumpulan sumber daya. Terakhir, Anda mengaitkan sumber daya dengan cakupan kesiapan, sehingga Anda bisa mendapatkan status kesiapan tentang sumber daya dalam grup pemulihan (aplikasi Anda) atau sel individual (replika, yaitu Regions atau Availability Zones ()). AZs

Kesiapan (yaitu, `READY` atau`NOT READY`) didasarkan pada sumber daya yang berada dalam lingkup pemeriksaan kesiapan dan seperangkat aturan untuk jenis sumber daya. Ada seperangkat aturan kesiapan untuk setiap jenis sumber daya, yang digunakan oleh ARC untuk mengaudit sumber daya untuk kesiapan. Apakah sumber daya `READY` atau tidak didasarkan pada bagaimana setiap aturan kesiapan didefinisikan. Semua aturan kesiapan mengevaluasi sumber daya, tetapi beberapa membandingkan sumber daya satu sama lain dan beberapa melihat informasi spesifik tentang setiap sumber daya dalam kumpulan sumber daya.

Dengan menambahkan pemeriksaan kesiapan, Anda dapat memantau status kesiapan, dengan salah satu dari beberapa cara: dengan EventBridge, di AWS Management Console, atau dengan menggunakan tindakan ARC API. Anda juga dapat memantau status kesiapan sumber daya dalam konteks yang berbeda, termasuk kesiapan sel dan kesiapan aplikasi Anda. Gunakan fitur otorisasi lintas akun di ARC untuk memudahkan penyiapan dan pemantauan sumber daya terdistribusi dari satu AWS akun.

Memantau replika aplikasi dengan pemeriksaan kesiapan

ARC mengaudit replika aplikasi Anda dengan menggunakan pemeriksaan kesiapan untuk memastikan bahwa masing-masing memiliki pengaturan konfigurasi yang sama dan status runtime yang sama. Pemeriksaan kesiapan terus mengaudit kapasitas AWS sumber daya, konfigurasi, AWS kuota, dan kebijakan perutean untuk aplikasi, informasi yang dapat Anda gunakan untuk membantu memastikan bahwa replika siap untuk failover. Pemeriksaan kesiapan membantu Anda memastikan bahwa lingkungan pemulihan Anda diskalakan dan dikonfigurasi agar gagal saat diperlukan.

Bagian berikut memberikan rincian lebih lanjut tentang cara kerja pemeriksaan kesiapan.

Pemeriksaan kesiapan dan replika aplikasi Anda

Agar siap untuk pemulihan, Anda harus mempertahankan kapasitas cadangan yang cukup dalam replika setiap saat, untuk menyerap lalu lintas failover dari Availability Zone atau Region lain. ARC terus menerus (sekali dalam satu menit) memeriksa aplikasi Anda untuk memastikan bahwa kapasitas yang Anda berikan cocok di semua Availability Zone atau Region.

Kapasitas yang diperiksa ARC meliputi, misalnya, jumlah EC2 instans Amazon, unit kapasitas baca dan tulis Aurora, dan ukuran volume Amazon EBS. Jika Anda meningkatkan kapasitas dalam replika utama Anda untuk nilai sumber daya tetapi lupa juga meningkatkan nilai yang sesuai dalam replika siaga Anda, ARC mendeteksi ketidakcocokan sehingga Anda dapat meningkatkan nilai dalam siaga.

> **Important**
>
> Pemeriksaan kesiapan paling berguna untuk memverifikasi, secara berkelanjutan, bahwa konfigurasi replika aplikasi dan status runtime diselaraskan. Pemeriksaan kesiapan tidak boleh digunakan untuk menunjukkan apakah replika produksi Anda sehat, Anda juga tidak boleh mengandalkan pemeriksaan kesiapan sebagai pemicu utama kegagalan selama peristiwa bencana.

Dalam konfigurasi siaga aktif, Anda harus membuat keputusan tentang apakah akan gagal dari atau ke sel berdasarkan pemantauan dan sistem pemeriksaan kesehatan Anda, dan mempertimbangkan pemeriksaan kesiapan sebagai layanan pelengkap untuk sistem tersebut. Pemeriksaan kesiapan ARC tidak terlalu tersedia, jadi Anda tidak boleh bergantung pada pemeriksaan yang dapat diakses selama pemadaman. Selain itu, sumber daya yang diperiksa mungkin juga tidak tersedia selama peristiwa bencana.

Anda dapat memantau status kesiapan untuk sumber daya aplikasi Anda di sel tertentu (AWS Wilayah atau Zona Ketersediaan) atau untuk keseluruhan aplikasi Anda. Anda dapat diberi tahu ketika status pemeriksaan kesiapan berubah, misalnya, menjadi`Not ready`, dengan membuat aturan di. EventBridge Untuk informasi selengkapnya, lihat Menggunakan pemeriksaan kesiapan di ARC dengan Amazon EventBridge. Anda juga dapat melihat status kesiapan di AWS Management Console, atau dengan menggunakan operasi API, seperti`get-recovery-readiness`. Untuk informasi selengkapnya, lihat Kesiapan memeriksa operasi API.

Cara kerja pemeriksaan kesiapan

ARC mengaudit replika aplikasi Anda dengan menggunakan pemeriksaan kesiapan untuk memastikan bahwa masing-masing memiliki pengaturan konfigurasi yang sama dan status runtime yang sama.

Untuk bersiap menghadapi pemulihan, misalnya, Anda harus mempertahankan kapasitas cadangan yang cukup setiap saat untuk menyerap lalu lintas failover dari Availability Zone atau Region lain. ARC terus menerus (sekali dalam satu menit) memeriksa aplikasi Anda untuk memastikan bahwa

kapasitas yang Anda berikan cocok di semua Availability Zone atau Region. Kapasitas yang diperiksa ARC meliputi, misalnya, jumlah EC2 instans Amazon, unit kapasitas baca dan tulis Aurora, dan ukuran volume Amazon EBS. Jika Anda meningkatkan kapasitas dalam replika utama Anda untuk nilai sumber daya tetapi lupa juga meningkatkan nilai yang sesuai dalam replika siaga Anda, ARC mendeteksi ketidakcocokan sehingga Anda dapat meningkatkan nilai dalam siaga.

> **Important**
>
> Pemeriksaan kesiapan paling berguna untuk memverifikasi, secara berkelanjutan, bahwa konfigurasi replika aplikasi dan status runtime diselaraskan. Pemeriksaan kesiapan tidak boleh digunakan untuk menunjukkan apakah replika produksi Anda sehat, Anda juga tidak boleh mengandalkan pemeriksaan kesiapan sebagai pemicu utama kegagalan selama peristiwa bencana.

Dalam konfigurasi siaga aktif, Anda harus membuat keputusan tentang apakah akan gagal dari atau ke sel berdasarkan pemantauan dan sistem pemeriksaan kesehatan Anda, dan mempertimbangkan pemeriksaan kesiapan sebagai layanan pelengkap untuk sistem tersebut. Pemeriksaan kesiapan ARC tidak terlalu tersedia, jadi Anda tidak boleh bergantung pada pemeriksaan yang dapat diakses selama pemadaman. Selain itu, sumber daya yang diperiksa mungkin juga tidak tersedia selama peristiwa bencana.

Anda dapat memantau status kesiapan untuk sumber daya aplikasi Anda di sel tertentu (AWS Wilayah atau Zona Ketersediaan) atau untuk keseluruhan aplikasi Anda. Anda dapat diberi tahu ketika status pemeriksaan kesiapan berubah, misalnya, menjadi`Not ready`, dengan membuat aturan di. EventBridge Untuk informasi selengkapnya, lihat Menggunakan pemeriksaan kesiapan di ARC dengan Amazon EventBridge. Anda juga dapat melihat status kesiapan di AWS Management Console, atau dengan menggunakan operasi API, seperti`get-recovery-readiness`. Untuk informasi selengkapnya, lihat Kesiapan memeriksa operasi API.

Bagaimana aturan kesiapan menentukan status kesiapan

Pemeriksaan kesiapan ARC menentukan status kesiapan berdasarkan aturan yang telah ditentukan untuk setiap jenis sumber daya dan cara aturan tersebut didefinisikan. ARC mencakup satu kelompok aturan untuk setiap jenis sumber daya yang didukungnya. Misalnya, ARC memiliki kelompok aturan kesiapan untuk cluster Amazon Aurora, grup Auto Scaling, dan sebagainya. Beberapa aturan kesiapan membandingkan sumber daya dalam satu set satu sama lain, dan beberapa melihat informasi spesifik tentang setiap sumber daya dalam kumpulan sumber daya.

Anda tidak dapat menambahkan, mengedit, atau menghapus aturan kesiapan, atau grup aturan. Namun, Anda dapat membuat CloudWatch alarm Amazon dan membuat pemeriksaan kesiapan untuk memantau keadaan alarm. Misalnya, Anda dapat membuat CloudWatch alarm khusus untuk memantau layanan kontainer Amazon EKS, dan membuat pemeriksaan kesiapan untuk mengaudit status kesiapan alarm.

Anda dapat melihat semua aturan kesiapan untuk setiap jenis sumber daya AWS Management Console saat Anda membuat kumpulan sumber daya, atau Anda dapat melihat aturan kesiapan nanti dengan menavigasi ke halaman detail untuk kumpulan sumber daya. Anda juga dapat melihat aturan kesiapan di bagian berikut:Aturan kesiapan di ARC.

Ketika pemeriksaan kesiapan mengaudit serangkaian sumber daya dengan seperangkat aturan, cara setiap aturan didefinisikan menentukan apakah hasilnya akan `READY` atau `NOT READY` untuk semua sumber daya atau jika hasilnya akan berbeda untuk sumber daya yang berbeda. Selain itu, Anda dapat melihat status kesiapan dalam berbagai cara. Misalnya, Anda dapat melihat status kesiapan grup sumber daya dalam kumpulan sumber daya atau melihat ringkasan status kesiapan untuk grup pemulihan atau sel (yaitu, AWS Wilayah atau Zona Ketersediaan, tergantung pada cara Anda mengatur grup pemulihan).

Kata-kata dalam setiap deskripsi aturan menjelaskan bagaimana mengevaluasi sumber daya untuk menentukan status kesiapan ketika aturan itu diterapkan. Aturan didefinisikan untuk memeriksa setiap sumber daya atau untuk memeriksa semua sumber daya dalam kumpulan sumber daya untuk menentukan kesiapan. Secara khusus, aturan berfungsi sebagai berikut:

- Aturan memeriksa setiap sumber daya dalam kumpulan sumber daya untuk memastikan suatu kondisi.
  - Jika semua sumber daya berhasil, semua sumber daya ditetapkan sebagai`READY`.
  - Jika satu sumber daya gagal, sumber daya itu ditetapkan sebagai`NOT READY`, dan sel lainnya tetap ada`READY`.

  Misalnya: MskClusterState:Memeriksa setiap cluster MSK Amazon untuk memastikan bahwa itu dalam keadaan. `ACTIVE`
- Aturan memeriksa semua sumber daya dalam kumpulan sumber daya untuk memastikan suatu kondisi.
  - Jika kondisinya dipastikan, semua sumber daya ditetapkan sebagai`READY`.
  - Jika ada yang gagal memenuhi kondisi, semua sumber daya ditetapkan sebagai`NOT READY`.

Misalnya: VpcSubnetCount:Memeriksa semua VPC subnet untuk memastikan bahwa mereka memiliki jumlah subnet yang sama.

- Aturan non-kritis: Aturan memeriksa semua sumber daya dalam kumpulan sumber daya untuk memastikan suatu kondisi.
  - Jika ada yang gagal, status kesiapan tidak berubah. Aturan dengan perilaku ini memiliki catatan dalam deskripsinya.

  Misalnya: ElbV2CheckAzCount:Memeriksa setiap Network Load Balancer untuk memastikan bahwa itu terpasang hanya pada satu Availability Zone. Catatan: Aturan ini tidak mempengaruhi status kesiapan.

Selain itu, ARC mengambil langkah ekstra untuk kuota. Jika pemeriksaan kesiapan mendeteksi ketidakcocokan di seluruh sel untuk kuota layanan (nilai maksimum untuk pembuatan dan operasi sumber daya) untuk sumber daya yang didukung, ARC secara otomatis menaikkan kuota untuk sumber daya dengan kuota yang lebih rendah. Ini hanya berlaku untuk kuota (batas). Untuk kapasitas, Anda harus menambahkan kapasitas tambahan sesuai kebutuhan aplikasi Anda.

Anda juga dapat mengatur EventBridge notifikasi Amazon untuk pemeriksaan kesiapan, misalnya, ketika status pemeriksaan kesiapan berubah menjadi. `NOT READY` Kemudian ketika ketidakcocokan konfigurasi terdeteksi, EventBridge mengirimkan pemberitahuan kepada Anda dan Anda dapat mengambil tindakan korektif untuk memastikan bahwa replika aplikasi Anda selaras dan disiapkan untuk pemulihan. Untuk informasi selengkapnya, lihat Menggunakan pemeriksaan kesiapan di ARC dengan Amazon EventBridge.

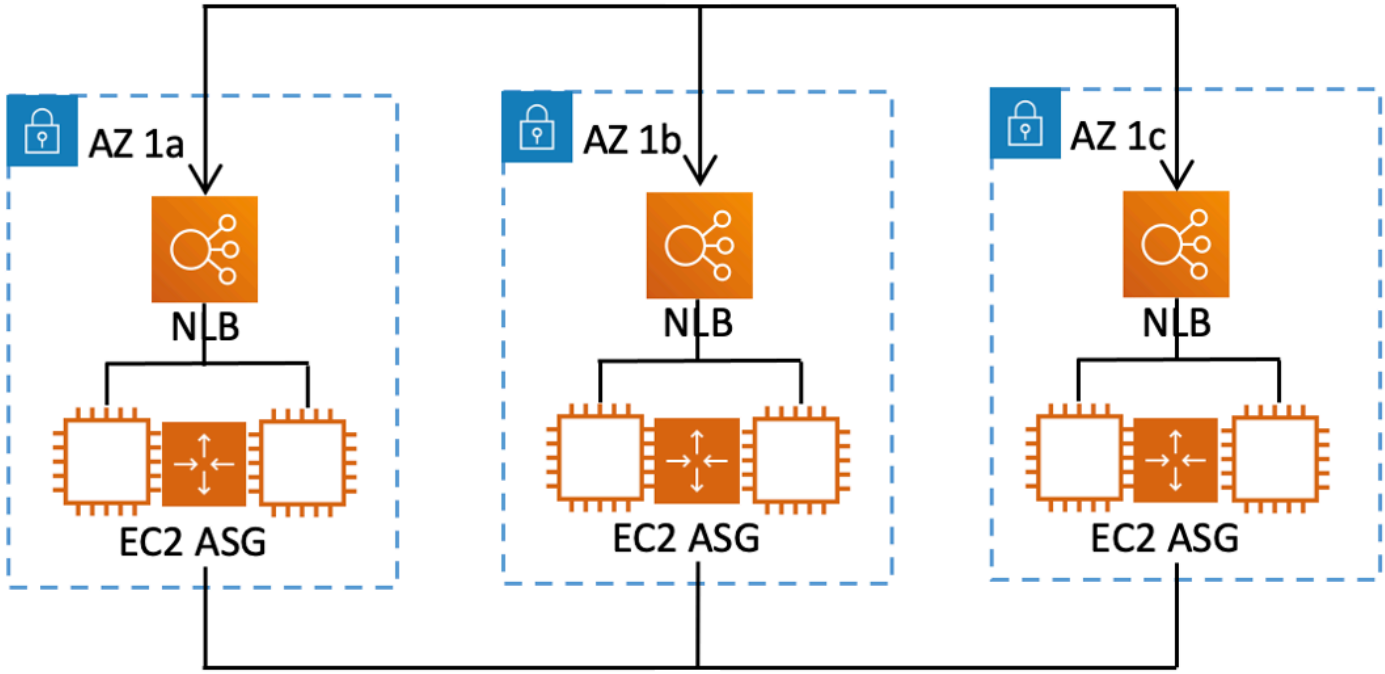
Bagaimana pemeriksaan kesiapan, kumpulan sumber daya, dan cakupan kesiapan bekerja sama

Pemeriksaan kesiapan selalu mengaudit kelompok sumber daya dalam kumpulan sumber daya. Anda membuat kumpulan sumber daya (secara terpisah, atau saat Anda membuat pemeriksaan kesiapan) untuk mengelompokkan sumber daya yang ada di sel (Availability Zones atau AWS Regions) di grup pemulihan ARC, sehingga Anda dapat menentukan pemeriksaan kesiapan. Kumpulan sumber daya biasanya merupakan kelompok dari jenis sumber daya yang sama (seperti Network Load Balancers) tetapi juga dapat menjadi sumber daya target DNS, untuk pemeriksaan kesiapan arsitektur.

Anda biasanya membuat satu set sumber daya dan pemeriksaan kesiapan untuk setiap jenis sumber daya dalam aplikasi Anda. Untuk pemeriksaan kesiapan arsitektur, Anda membuat sumber daya

target DNS tingkat atas dan sumber daya global (tingkat grup pemulihan) yang ditetapkan untuknya, lalu buat sumber daya target DNS tingkat sel, untuk kumpulan sumber daya terpisah.

Diagram berikut menunjukkan contoh grup pemulihan dengan tiga sel (Availability Zones), masing-masing dengan Network Load Balancer (NLB) dan grup Auto Scaling (ASG).



Dalam skenario ini, Anda akan membuat kumpulan sumber daya dan pemeriksaan kesiapan untuk tiga Network Load Balancer, dan satu set sumber daya dan pemeriksaan kesiapan untuk tiga grup Auto Scaling. Sekarang Anda memiliki pemeriksaan kesiapan untuk setiap set sumber daya untuk grup pemulihan Anda, berdasarkan jenis sumber daya.

Dengan membuat cakupan kesiapan untuk sumber daya, Anda dapat menambahkan ringkasan pemeriksaan kesiapan untuk sel atau grup pemulihan. Untuk menentukan lingkup kesiapan sumber daya, Anda mengaitkan ARN sel atau grup pemulihan dengan setiap sumber daya dalam kumpulan sumber daya. Anda dapat melakukan ini ketika Anda membuat pemeriksaan kesiapan untuk kumpulan sumber daya.

Misalnya, saat Anda menambahkan pemeriksaan kesiapan untuk kumpulan sumber daya untuk Network Load Balancers untuk grup pemulihan ini, Anda dapat menambahkan cakupan kesiapan ke setiap NLB secara bersamaan. Dalam hal ini, Anda akan mengaitkan ARN dari AZ 1a ke NLB di AZ 1a, ARN dari ke NLB, `AZ 1b` dan ARN ke NLB `AZ 1b` di. `AZ 1c` `AZ 1c` Saat Anda membuat pemeriksaan kesiapan untuk grup Auto Scaling, Anda akan melakukan hal yang sama, menetapkan

cakupan kesiapan untuk masing-masing grup saat Anda membuat pemeriksaan kesiapan untuk kumpulan sumber daya grup Auto Scaling.

Ini opsional untuk mengaitkan cakupan kesiapan saat Anda membuat pemeriksaan kesiapan, namun, kami sangat menyarankan Anda mengaturnya. Cakupan kesiapan memungkinkan ARC untuk menunjukkan status yang benar `READY` atau `NOT READY` kesiapan untuk pemeriksaan kesiapan ringkasan grup pemulihan dan pemeriksaan kesiapan ringkasan tingkat sel. Kecuali Anda menetapkan cakupan kesiapan, ARC tidak dapat memberikan ringkasan ini.

Perhatikan bahwa saat menambahkan tingkat aplikasi atau sumber daya global, seperti kebijakan perutean DNS, Anda tidak memilih grup atau sel pemulihan untuk cakupan kesiapan. Sebagai gantinya, Anda memilih sumber daya global (tanpa sel).

Pemeriksaan kesiapan sumber daya target DNS: Kesiapan ketahanan audit

Dengan pemeriksaan kesiapan sumber daya target DNS di ARC, Anda dapat mengaudit kesiapan arsitektur dan ketahanan aplikasi Anda. Jenis pemeriksaan kesiapan ini terus memindai arsitektur aplikasi Anda dan kebijakan perutean Amazon Route 53 untuk mengaudit dependensi lintas zona dan lintas wilayah.

Aplikasi berorientasi pemulihan memiliki beberapa replika yang dimasukkan ke dalam Availability Zone atau AWS Regions, sehingga replika dapat gagal secara independen satu sama lain. Jika aplikasi Anda perlu menyesuaikan agar di-siloed dengan benar, ARC akan menyarankan perubahan yang dapat Anda buat, jika perlu, untuk memperbarui arsitektur Anda guna membantu memastikan bahwa itu tangguh dan siap untuk failover.

ARC secara otomatis mendeteksi nomor dan cakupan sel (mewakili replika, atau unit penahanan kegagalan) dalam aplikasi Anda, dan apakah sel disilokan oleh Availability Zone atau Region. Kemudian, ARC mengidentifikasi dan memberikan informasi kepada Anda tentang sumber daya aplikasi dalam sel, untuk menentukan apakah mereka tersilo dengan benar ke zona atau Wilayah. Misalnya, jika Anda memiliki sel yang tercakup ke zona tertentu, pemeriksaan kesiapan dapat memantau apakah penyeimbang beban Anda dan target di belakangnya juga terdiam ke zona tersebut.

Dengan informasi ini, Anda dapat menentukan apakah ada perubahan yang perlu Anda lakukan untuk menyelaraskan sumber daya di sel Anda ke zona atau Wilayah yang benar.

Untuk memulai, Anda membuat sumber daya target DNS untuk aplikasi Anda, dan set sumber daya serta pemeriksaan kesiapan untuk mereka. Untuk informasi selengkapnya, lihat Mendapatkan rekomendasi arsitektur di ARC.

Pemeriksaan kesiapan dan skenario pemulihan bencana

Pemeriksaan kesiapan ARC memberi Anda wawasan tentang apakah aplikasi dan sumber daya Anda siap untuk pemulihan dengan membantu Anda memastikan bahwa aplikasi Anda diskalakan untuk menangani lalu lintas failover. Status pemeriksaan kesiapan tidak boleh digunakan sebagai sinyal untuk menunjukkan bahwa replika produksi sehat. Namun, Anda dapat menggunakan pemeriksaan kesiapan sebagai pelengkap aplikasi dan pemantauan infrastruktur atau sistem pemeriksa kesehatan Anda untuk menentukan apakah akan gagal dari atau ke replika.

Dalam situasi mendesak atau pemadaman, gunakan kombinasi pemeriksaan kesehatan dan informasi lainnya untuk menentukan bahwa siaga Anda ditingkatkan, sehat, dan siap bagi Anda untuk gagal melewati lalu lintas produksi. Misalnya, periksa untuk melihat apakah kenari yang berjalan melawan sel siaga Anda memenuhi kriteria keberhasilan Anda, selain memverifikasi bahwa status pemeriksaan kesiapan untuk siaga tersebut. `READY`

Ketahuilah bahwa pemeriksaan kesiapan ARC diselenggarakan di satu AWS Wilayah, AS Barat (Oregon), dan selama pemadaman atau bencana, informasi pemeriksaan kesiapan bisa menjadi basi atau cek bisa menjadi tidak tersedia. Untuk informasi selengkapnya, lihat Bidang data dan kontrol untuk kontrol perutean.

AWS Ketersediaan wilayah untuk pemeriksaan kesiapan

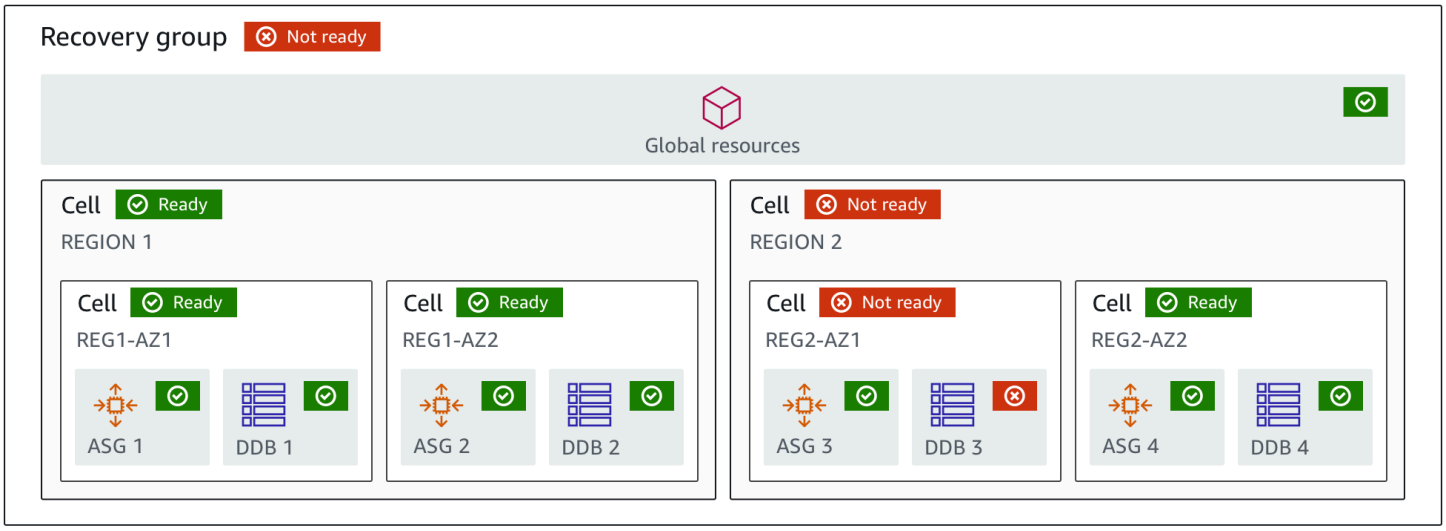
Untuk informasi terperinci tentang dukungan Regional dan titik akhir layanan untuk Amazon Application Recovery Controller (ARC), lihat titik akhir dan kuota Amazon Application Recovery Controller (ARC) di Referensi Umum Amazon Web Services.

> **Note**
>
> Pemeriksaan kesiapan di Amazon Application Recovery Controller (ARC) adalah fitur global. Namun, sumber daya pemeriksaan kesiapan ada di Wilayah AS Barat (Oregon), jadi Anda harus menentukan Wilayah Barat AS (Oregon) (tentukan parameternya`--region us-west-2`) dalam AWS CLI perintah ARC Regional, misalnya, saat Anda membuat sumber daya seperti set sumber daya dan pemeriksaan kesiapan.

Komponen pemeriksaan kesiapan

Diagram berikut menggambarkan kelompok pemulihan sampel yang dikonfigurasi untuk mendukung fitur pemeriksaan kesiapan. Sumber daya dalam contoh ini dikelompokkan ke dalam sel (oleh

Wilayah AWS) dan sel bersarang (menurut Availability Zones) dalam grup pemulihan. Ada status kesiapan keseluruhan untuk kelompok pemulihan (aplikasi), serta status kesiapan individu untuk setiap sel (Wilayah) dan sel bersarang (Availability Zone).



Berikut ini adalah komponen fitur pemeriksaan kesiapan di ARC.

Sel

Sel mendefinisikan replika aplikasi Anda atau unit failover independen. Ini mengelompokkan semua AWS sumber daya yang diperlukan agar aplikasi Anda berjalan secara independen di dalam replika. Misalnya, Anda mungkin memiliki satu set sumber daya di sel primer dan satu set lain di sel siaga. Anda menentukan batas dari apa yang disertakan sel, tetapi sel biasanya mewakili Availability Zone atau Region. Anda dapat memiliki beberapa sel (sel bersarang) di dalam sel, seperti AZs di dalam Wilayah. Setiap sel bersarang mewakili unit failover yang terisolasi.

Kelompok pemulihan

Sel dikumpulkan ke dalam kelompok pemulihan. Grup pemulihan mewakili aplikasi atau grup aplikasi yang ingin Anda periksa kesiapan failover. Ini terdiri dari dua atau lebih sel, atau replika, yang cocok satu sama lain dalam hal fungsionalitas. Misalnya, jika Anda memiliki aplikasi web yang direplikasi di us-east-1a dan us-east-1b, di mana us-east-1b adalah lingkungan failover Anda, Anda dapat mewakili aplikasi ini di ARC sebagai grup pemulihan dengan dua sel: satu di us-east-1a dan satu di us-east-1b. Kelompok pemulihan juga dapat mencakup sumber daya global, seperti pemeriksaan kesehatan Route 53.

Sumber daya dan pengidentifikasi sumber daya

Saat membuat komponen untuk pemeriksaan kesiapan di ARC, Anda menentukan sumber daya, seperti tabel Amazon DynamoDB, Network Load Balancer, atau sumber daya target DNS, dengan menggunakan pengenal sumber daya. Pengidentifikasi sumber daya adalah Amazon Resource Name (ARN) untuk sumber daya atau, untuk sumber daya target DNS, pengidentifikasi yang dihasilkan ARC saat membuat sumber daya.

Sumber daya target DNS

Sumber daya target DNS adalah kombinasi dari nama domain aplikasi Anda dan informasi DNS lainnya, seperti AWS sumber daya yang ditunjuk domain. Menyertakan AWS sumber daya bersifat opsional tetapi jika Anda menyediakannya, itu harus berupa catatan sumber daya Route 53 atau Network Load Balancer. Ketika Anda menyediakan AWS sumber daya, Anda bisa mendapatkan rekomendasi arsitektur yang lebih rinci yang dapat membantu Anda meningkatkan ketahanan pemulihan aplikasi Anda. Anda dapat membuat kumpulan sumber daya di ARC untuk sumber daya target DNS, dan kemudian membuat pemeriksaan kesiapan untuk kumpulan sumber daya sehingga Anda bisa mendapatkan rekomendasi arsitektur untuk aplikasi Anda. Pemeriksaan kesiapan juga memantau kebijakan perutean DNS untuk aplikasi Anda, berdasarkan aturan kesiapan untuk sumber daya target DNS.

Set sumber daya

Kumpulan sumber daya adalah sekumpulan sumber daya, termasuk sumber AWS daya atau sumber daya target DNS, yang menjangkau beberapa sel. Misalnya, Anda mungkin memiliki penyeimbang beban di us-east-1a dan satu lagi di us-east-1b. Untuk memantau kesiapan pemulihan penyeimbang beban, Anda dapat membuat kumpulan sumber daya yang mencakup kedua penyeimbang beban, dan kemudian membuat pemeriksaan kesiapan untuk kumpulan sumber daya. ARC akan terus memeriksa kesiapan sumber daya di set. Anda juga dapat menambahkan cakupan kesiapan untuk mengaitkan sumber daya dalam kumpulan sumber daya dengan grup pemulihan yang Anda buat untuk aplikasi Anda.

Aturan kesiapan

Aturan kesiapan adalah audit yang dilakukan ARC terhadap serangkaian sumber daya dalam kumpulan sumber daya. ARC memiliki seperangkat aturan kesiapan untuk setiap jenis sumber daya yang mendukung pemeriksaan kesiapan. Setiap aturan menyertakan ID dan deskripsi yang menjelaskan untuk apa ARC memeriksa sumber daya.

Pemeriksaan kesiapan

Pemeriksaan kesiapan memantau set sumber daya dalam aplikasi Anda, seperti sekumpulan instans Amazon Aurora, yang ARC mengaudit kesiapan pemulihan. Pemeriksaan kesiapan dapat mencakup audit, misalnya, konfigurasi kapasitas, AWS kuota, atau kebijakan perutean. Misalnya, jika Anda ingin mengaudit kesiapan untuk grup EC2 Auto Scaling Amazon di dua Availability Zone, Anda dapat membuat pemeriksaan kesiapan untuk kumpulan sumber daya dengan dua ARNs sumber daya, satu untuk setiap grup Auto Scaling. Kemudian, untuk memastikan bahwa setiap grup diskalakan secara merata, ARC terus memantau jenis instance dan jumlah dalam dua grup.

Ruang lingkup kesiapan

Lingkup kesiapan mengidentifikasi pengelompokan sumber daya yang mencakup pemeriksaan kesiapan tertentu. Ruang lingkup pemeriksaan kesiapan dapat berupa kelompok pemulihan (yaitu, global untuk seluruh aplikasi) atau sel (yaitu, Wilayah atau Zona Ketersediaan). Untuk sumber daya yang merupakan sumber daya global untuk ARC, tetapkan ruang lingkup kesiapan ke grup pemulihan atau tingkat sumber daya global. Misalnya, pemeriksaan kesehatan Route 53 adalah sumber daya global di ARC karena tidak spesifik untuk Wilayah atau Zona Ketersediaan.

Data dan pesawat kontrol untuk pemeriksaan kesiapan

Saat Anda merencanakan kegagalan dan pemulihan bencana, pertimbangkan seberapa tangguh mekanisme failover Anda. Kami menyarankan Anda memastikan bahwa mekanisme yang Anda andalkan selama failover sangat tersedia, sehingga Anda dapat menggunakannya saat Anda membutuhkannya dalam skenario bencana. Biasanya, Anda harus menggunakan fungsi bidang data untuk mekanisme Anda kapan pun Anda bisa, untuk keandalan dan toleransi kesalahan terbesar. Dengan mengingat hal itu, penting untuk memahami bagaimana fungsionalitas layanan dibagi antara bidang kontrol dan pesawat data, dan kapan Anda dapat mengandalkan harapan keandalan ekstrim dengan bidang data layanan.

Seperti kebanyakan AWS layanan, fungsionalitas untuk kemampuan pemeriksaan kesiapan didukung oleh pesawat kontrol dan pesawat data. Meskipun keduanya dibangun agar dapat diandalkan, bidang kontrol dioptimalkan untuk konsistensi data, sementara bidang data dioptimalkan untuk ketersediaan. Pesawat data dirancang untuk ketahanan sehingga dapat mempertahankan ketersediaan bahkan selama peristiwa yang mengganggu, ketika pesawat kontrol mungkin menjadi tidak tersedia.

Secara umum, bidang kontrol memungkinkan Anda melakukan fungsi manajemen dasar, seperti membuat, memperbarui, dan menghapus sumber daya dalam layanan. Pesawat data menyediakan fungsionalitas inti layanan.

Untuk pemeriksaan kesiapan, ada satu API, API Kesiapan Pemulihan, untuk bidang kontrol dan bidang data. Pemeriksaan kesiapan dan sumber daya kesiapan hanya ada di Wilayah Barat AS (Oregon) (us-west-2). Bidang kontrol pemeriksaan kesiapan dan bidang data dapat diandalkan tetapi tidak terlalu tersedia.

Untuk informasi selengkapnya tentang bidang data, pesawat kontrol, dan cara AWS membangun layanan untuk memenuhi target ketersediaan tinggi, lihat paper Stabilitas statis menggunakan Availability Zones di Amazon Builders' Library.

Menandai pemeriksaan kesiapan di Amazon Application Recovery Controller (ARC)

Tag adalah kata atau frasa (meta data) yang Anda gunakan untuk mengidentifikasi dan mengatur AWS sumber daya Anda. Anda dapat menambahkan beberapa tag ke setiap sumber daya, dan setiap tag mencakup kunci dan nilai yang Anda tentukan. Misalnya, kuncinya mungkin lingkungan dan nilainya mungkin produksi. Anda dapat mencari dan memfilter sumber daya Anda berdasarkan tanda yang Anda tambahkan.

Anda dapat menandai sumber daya berikut dalam pemeriksaan kesiapan di ARC:

- Set sumber daya
- Pemeriksaan kesiapan

Penandaan di ARC hanya tersedia melalui API, misalnya, dengan menggunakan file. AWS CLI

Berikut ini adalah contoh penandaan dalam pemeriksaan kesiapan dengan menggunakan. AWS CLI

```
aws route53-recovery-readiness --region us-west-2 create-resource-
set --resource-set-name dynamodb_resource_set --resource-set-type
AWS::DynamoDB::Table --resources ReadinessScopes=arn:aws:aws-recovery-
readiness::111122223333:cell/PDXCell,ResourceArn=arn:aws:dynamodb:us-
west-2:111122223333:table/PDX_Table ReadinessScopes=arn:aws:aws-recovery-
readiness::111122223333:cell/IADCell,ResourceArn=arn:aws:dynamodb:us-
east-1:111122223333:table/IAD_Table --tags Stage=Prod
```

```
aws route53-recovery-readiness --region us-west-2 create-readiness-
check --readiness-check-name dynamodb_readiness_check --resource-set-name
dynamodb_resource_set --tags Stage=Prod
```

Untuk informasi selengkapnya, lihat TagResourcedi Panduan Referensi API Kesiapan Pemulihan untuk Amazon Application Recovery Controller (ARC).

Harga untuk pemeriksaan kesiapan di ARC

Anda membayar biaya per jam per pemeriksaan kesiapan yang Anda konfigurasikan.

Untuk informasi harga terperinci untuk ARC dan contoh harga, lihat Harga ARC.

Siapkan proses pemulihan yang tangguh untuk aplikasi Anda

Untuk menggunakan Amazon Application Recovery Controller (ARC) dengan AWS aplikasi yang ada di beberapa AWS Wilayah, ada panduan yang harus diikuti untuk menyiapkan aplikasi Anda agar tahan, sehingga Anda dapat mendukung kesiapan pemulihan secara efektif. Kemudian, Anda dapat membuat pemeriksaan kesiapan untuk aplikasi Anda dan mengatur kontrol perutean untuk mengalihkan lalu lintas untuk failover. Anda juga dapat meninjau rekomendasi yang diberikan ARC tentang arsitektur aplikasi Anda yang dapat meningkatkan ketahanan.

> **Note**
>
> Jika Anda memiliki aplikasi yang dibungkam oleh Availability Zones, pertimbangkan untuk menggunakan zonal shift atau zonal autoshift untuk pemulihan failover. Tidak diperlukan pengaturan untuk menggunakan pergeseran zona atau pergeseran otomatis zona untuk memulihkan aplikasi dengan andal dari gangguan Availability Zone.
> Untuk memindahkan lalu lintas dari Availability Zone untuk sumber daya penyeimbang beban, mulailah pergeseran zona di konsol ARC atau di konsol Elastic Load Balancing. Atau, Anda dapat menggunakan AWS Command Line Interface atau AWS SDK dengan tindakan API pergeseran zona. Untuk informasi selengkapnya, lihat Pergeseran zona di ARC.

Untuk mempelajari lebih lanjut tentang memulai konfigurasi failover yang tangguh, lihat. Memulai pemulihan Multi-wilayah di Amazon Application Recovery Controller (ARC)

Praktik terbaik untuk pemeriksaan kesiapan di ARC

Kami merekomendasikan praktik terbaik berikut untuk pemeriksaan kesiapan di Amazon Application Recovery Controller (ARC).

Tambahkan notifikasi untuk perubahan status kesiapan

Tetapkan aturan di Amazon EventBridge untuk mengirim pemberitahuan setiap kali status pemeriksaan kesiapan berubah, misalnya, dari `READY` ke`NOT READY`. Ketika Anda menerima pemberitahuan, Anda dapat menyelidiki dan mengatasi masalah tersebut, untuk memastikan bahwa aplikasi dan sumber daya Anda siap untuk failover saat Anda mengharapkannya.

Anda dapat menetapkan EventBridge aturan untuk mengirim pemberitahuan untuk beberapa perubahan status pemeriksaan kesiapan, termasuk untuk grup pemulihan (untuk aplikasi Anda), untuk sel (seperti AWS Wilayah), atau untuk pemeriksaan kesiapan untuk kumpulan sumber daya.

Untuk informasi selengkapnya, lihat Menggunakan pemeriksaan kesiapan di ARC dengan Amazon EventBridge.

Kesiapan memeriksa operasi API

Tabel berikut mencantumkan operasi ARC yang dapat Anda gunakan untuk kesiapan pemulihan (pemeriksaan kesiapan), dengan tautan ke dokumentasi yang relevan.

Untuk contoh cara menggunakan operasi API kesiapan pemulihan umum dengan AWS Command Line Interface, lihat Contoh penggunaan operasi API pemeriksaan kesiapan ARC dengan AWS CLI.

Tindakan	Menggunakan konsol ARC	Menggunakan ARC API
Buat sel	Lihat Membuat, memperbarui, dan menghapus grup pemulihan di ARC	Lihat CreateCell
Dapatkan sel	Lihat Membuat, memperbarui, dan menghapus grup pemulihan di ARC	Lihat GetCell
Hapus sel	Lihat Membuat, memperbarui, dan menghapus grup pemulihan di ARC	Lihat DeleteCell

Tindakan	Menggunakan konsol ARC	Menggunakan ARC API
Perbarui sel	N/A	Lihat UpdateCell
Daftar sel untuk akun	Lihat Membuat, memperbarui, dan menghapus grup pemulihan di ARC	Lihat ListCells
Buat grup pemulihan	Lihat Membuat, memperbarui, dan menghapus grup pemulihan di ARC	Lihat CreateRecoveryGroup
Dapatkan grup pemulihan	Lihat Membuat, memperbarui, dan menghapus grup pemulihan di ARC	Lihat GetRecoveryGroup
Perbarui grup pemulihan	Lihat Membuat, memperbarui, dan menghapus grup pemulihan di ARC	Lihat UpdateRecoveryGroup
Hapus grup pemulihan	Lihat Membuat, memperbarui, dan menghapus grup pemulihan di ARC	Lihat DeleteRecoveryGroup
Daftar kelompok pemulihan	Lihat Membuat, memperbarui, dan menghapus grup pemulihan di ARC	Lihat ListRecoveryGroups
Buat kumpulan sumber daya	Lihat Membuat dan memperbarui pemeriksaan kesiapan di ARC	Lihat CreateResourceSet
Dapatkan satu set sumber daya	Lihat Membuat dan memperbarui pemeriksaan kesiapan di ARC	Lihat GetResourceSet
Perbarui kumpulan sumber daya	Lihat Membuat dan memperbarui pemeriksaan kesiapan di ARC	Lihat UpdateResourceSet

Tindakan	Menggunakan konsol ARC	Menggunakan ARC API
Hapus kumpulan sumber daya	Lihat Membuat dan memperbarui pemeriksaan kesiapan di ARC	Lihat DeleteResourceSet
Daftar kumpulan sumber daya	Lihat Membuat dan memperbarui pemeriksaan kesiapan di ARC	Lihat ListResourceSets
Buat pemeriksaan kesiapan	Lihat Membuat dan memperbarui pemeriksaan kesiapan di ARC	Lihat CreateReadinessCheck
Dapatkan cek kesiapan	Lihat Membuat dan memperbarui pemeriksaan kesiapan di ARC	Lihat GetReadinessCheck
Perbarui pemeriksaan kesiapan	Lihat Membuat dan memperbarui pemeriksaan kesiapan di ARC	Lihat UpdateReadinessCheck
Hapus pemeriksaan kesiapan	Lihat Membuat dan memperbarui pemeriksaan kesiapan di ARC	Lihat DeleteReadinessCheck
Daftar pemeriksaan kesiapan	Lihat Membuat dan memperbarui pemeriksaan kesiapan di ARC	Lihat ListReadinessChecks
Daftar aturan kesiapan	Lihat Deskripsi aturan kesiapan di ARC	Lihat ListRules
Periksa status seluruh pemeriksaan kesiapan	Lihat Memantau status kesiapan di ARC	Lihat GetReadinessCheckStatus
Periksa status sumber daya	Lihat Memantau status kesiapan di ARC	Lihat GetReadinessCheckResourceStatus

Tindakan	Menggunakan konsol ARC	Menggunakan ARC API
Periksa status sel	Lihat Memantau status kesiapan di ARC	Lihat GetCellReadinessSummary
Periksa status grup pemulihan	Lihat Memantau status kesiapan di ARC	Lihat GetRecoveryGroupReadinessSummary

Contoh penggunaan operasi API pemeriksaan kesiapan ARC dengan AWS CLI

Bagian ini berjalan melalui contoh aplikasi sederhana, menggunakan AWS Command Line Interface untuk bekerja dengan fitur pemeriksaan kesiapan di Amazon Application Recovery Controller (ARC) menggunakan operasi API. Contohnya dimaksudkan untuk membantu Anda mengembangkan pemahaman dasar tentang cara bekerja dengan kemampuan pemeriksaan kesiapan menggunakan CLI.

Pemeriksaan kesiapan dalam audit ARC untuk ketidakcocokan sumber daya dalam replika aplikasi Anda. Untuk menyiapkan pemeriksaan kesiapan untuk aplikasi Anda, Anda harus mengatur—atau memodelkan—sumber daya aplikasi Anda dalam sel ARC yang sejajar dengan replika yang telah Anda buat untuk aplikasi Anda. Anda kemudian mengatur pemeriksaan kesiapan yang mengaudit replika ini, untuk membantu Anda memastikan bahwa replika aplikasi siaga Anda dan sumber dayanya sesuai dengan replika produksi Anda, secara berkelanjutan

Mari kita lihat kasus sederhana di mana Anda memiliki aplikasi bernama Simple-Service yang saat ini berjalan di Wilayah AS Timur (Virginia N.) (us-east-1). Anda juga memiliki salinan siaga aplikasi di Wilayah AS Barat (Oregon) (us-west-2). Dalam contoh ini, kita akan mengkonfigurasi pemeriksaan kesiapan untuk membandingkan dua versi aplikasi ini. Ini memungkinkan kami memastikan bahwa siaga, Wilayah Barat AS (Oregon), siap menerima lalu lintas, jika perlu dalam skenario failover.

Untuk informasi selengkapnya tentang penggunaan AWS CLI, lihat Referensi AWS CLI Perintah. Untuk daftar tindakan API kesiapan dan tautan ke informasi selengkapnya, lihat Kesiapan memeriksa operasi API.

Sel di ARC mewakili batas kesalahan (seperti Availability Zone atau Regions) dan dikumpulkan ke dalam grup pemulihan. Grup pemulihan mewakili aplikasi yang ingin Anda periksa kesiapan failover. Untuk informasi lebih lanjut tentang komponen pemeriksaan kesiapan, lihatKomponen pemeriksaan kesiapan .

> **Note**
>
> ARC adalah layanan global yang mendukung titik akhir dalam beberapa Wilayah AWS tetapi Anda harus menentukan Wilayah AS Barat (Oregon) (yaitu, tentukan parameter`--region us-west-2`) di sebagian besar perintah ARC CLI. Misalnya, untuk membuat sumber daya seperti grup pemulihan atau pemeriksaan kesiapan.

Untuk contoh aplikasi kita, kita akan mulai dengan membuat satu sel untuk setiap Wilayah di mana kita memiliki sumber daya. Kemudian kita akan membuat grup pemulihan, dan kemudian menyelesaikan pengaturan untuk pemeriksaan kesiapan.

1. Buat sel

1a. Buat sel us-east-1.

```
aws route53-recovery-readiness --region us-west-2 create-cell \
    --cell-name east-cell
```

```
{
    "CellArn": "arn:aws:route53-recovery-readiness::111122223333:cell/east-cell",
    "CellName": "east-cell",
    "Cells": [],
    "ParentReadinessScopes": [],
    "Tags": {}
}
```

1b. Buat sel us-west-1.

```
aws route53-recovery-readiness --region us-west-2 create-cell \
    --cell-name west-cell
```

```
{
    "CellArn": "arn:aws:route53-recovery-readiness::111122223333:cell/west-cell",
    "CellName": "west-cell",
    "Cells": [],
    "ParentReadinessScopes": [],
    "Tags": {}
}
```

1c. Sekarang kita memiliki dua sel. Anda dapat memverifikasi bahwa mereka ada dengan memanggil `list-cells` API.

```
aws route53-recovery-readiness --region us-west-2 list-cells
```

```
{
    "Cells": [
        {
            "CellArn": "arn:aws:route53-recovery-readiness::111122223333:cell/east-
cell",
            "CellName": "east-cell",
            "Cells": [],
            "ParentReadinessScopes": [],
            "Tags": {}
        },
        {
            "CellArn": "arn:aws:route53-recovery-readiness::111122223333:cell/west-
cell",
            "CellName": "west-cell"
            "Cells": [],
            "ParentReadinessScopes": [],
            "Tags": {}
        }
    ]
}
```

2. Buat grup pemulihan

Grup pemulihan adalah sumber daya tingkat atas untuk kesiapan pemulihan di ARC. Kelompok pemulihan mewakili aplikasi secara keseluruhan. Pada langkah ini, kita akan membuat grup pemulihan untuk memodelkan aplikasi secara keseluruhan, dan kemudian menambahkan dua sel yang kita buat.

2a. Buat grup pemulihan.

```
aws route53-recovery-readiness --region us-west-2 create-recovery-group \
    --recovery-group-name simple-service-recovery-group \
       --cells "arn:aws:route53-recovery-readiness::111122223333:cell/east-cell"\
       "arn:aws:route53-recovery-readiness::111122223333:cell/west-cell"
```

```
{
```

```
    "Cells": [],
    "RecoveryGroupArn": "arn:aws:route53-recovery-readiness::111122223333:recovery-
group/simple-service-recovery-group",
    "RecoveryGroupName": "simple-service-recovery-group",
    "Tags": {}
}
```

2b. (Opsional) Anda dapat memverifikasi bahwa grup pemulihan Anda dibuat dengan benar dengan memanggil `list-recovery-groups` API.

```
aws route53-recovery-readiness --region us-west-2 list-recovery-groups
```

```
{
    "RecoveryGroups": [
        {
            "Cells": [
                "arn:aws:route53-recovery-readiness::111122223333:cell/east-cell",
                "arn:aws:route53-recovery-readiness::111122223333:cell/west-cell"
            ],
            "RecoveryGroupArn": "arn:aws:route53-recovery-
readiness::111122223333:recovery-group/simple-service-recovery-group",
            "RecoveryGroupName": "simple-service-recovery-group",
            "Tags": {}
        }
    ]
}
```

Sekarang kita memiliki model untuk aplikasi kita, mari tambahkan sumber daya yang akan dipantau. Di ARC, sekelompok sumber daya yang ingin Anda pantau disebut kumpulan sumber daya. Kumpulan sumber daya berisi sumber daya yang semuanya dari jenis yang sama. Kami membandingkan sumber daya dalam kumpulan sumber daya satu sama lain untuk membantu menentukan kesiapan sel untuk failover.

3. Buat kumpulan sumber daya

Mari kita asumsikan kita Simple-Service aplikasi ini memang sangat sederhana dan hanya menggunakan tabel DynamoDB. Ini memiliki tabel DynamoDB di us-east-1 dan satu lagi di us-west-2. Kumpulan sumber daya juga berisi ruang lingkup kesiapan, yang mengidentifikasi sel tempat setiap sumber daya terkandung di dalamnya.

3a. Buat kumpulan sumber daya yang mencerminkan Simple-Service sumber daya aplikasi.

```
aws route53-recovery-readiness --region us-west-2 create-resource-set \
    --resource-set-name ImportantInformationTables \
    --resource-set-type AWS::DynamoDB::Table \
    --resources
    ResourceArn="arn:aws:dynamodb:us-west-2:111122223333:table/
TableInUsWest2",ReadinessScopes="arn:aws:route53-recovery-readiness::111122223333:cell/
west-cell"
    ResourceArn="arn:aws:dynamodb:us-west-2:111122223333:table/
TableInUsEast1",ReadinessScopes="arn:aws:route53-recovery-readiness::111122223333:cell/
east-cell"
```

```
{
    "ResourceSetArn": "arn:aws:route53-recovery-readiness::111122223333:resource-set/
sample-resource-set",
    "ResourceSetName": "ImportantInformationTables",
    "Resources": [
        {
            "ReadinessScopes": [
                "arn:aws:route53-recovery-readiness::111122223333:cell/west-cell"
            ],
            "ResourceArn": "arn:aws:dynamodb:us-west-2:111122223333:table/
TableInUsWest2"
        },
        {
            "ReadinessScopes": [
                "arn:aws:route53-recovery-readiness::111122223333:cell/east-cell"
            ],
            "ResourceArn": "arn:aws:dynamodb:us-west-2:111122223333:table/
TableInUsEast1"
        }
    ],
    "Tags": {}
}
```

3b. (Opsional) Anda dapat memverifikasi apa yang disertakan dalam kumpulan sumber daya dengan memanggil `list-resource-sets` API. Ini mencantumkan semua kumpulan sumber daya untuk AWS akun. Di sini Anda dapat melihat bahwa kami hanya memiliki satu set sumber daya yang kami buat di atas.

```
aws route53-recovery-readiness --region us-west-2 list-resource-sets
```

```
{
    "ResourceSets": [
        {
            "ResourceSetArn": "arn:aws:route53-recovery-
readiness::111122223333:resource-set/ImportantInformationTables",
            "ResourceSetName": "ImportantInformationTables",
            "Resources": [
                {
                    "ReadinessScopes": [
                        "arn:aws:route53-recovery-readiness::111122223333:cell/west-
cell"
                    ],
                    "ResourceArn": "arn:aws:dynamodb:us-west-2:111122223333:table/
TableInUsWest2"
                },
                {
                    "ReadinessScopes": [
                        "arn:aws:route53-recovery-readiness::111122223333:cell/east-
cell"
                    ],
                    "ResourceArn": "arn:aws:dynamodb:us-west-2:111122223333:table/
TableInUsEast1"
                }
            ],
            "Tags": {}
        }
    ]
}{
    "ResourceSets": [
        {
            "ResourceSetArn": "arn:aws:route53-recovery-
readiness::111122223333:resource-set/ImportantInformationTables",
            "ResourceSetName": "ImportantInformationTables",
            "Resources": [
                {
                    "ReadinessScopes": [
                        "arn:aws:route53-recovery-readiness::111122223333:cell/west-
cell"
                    ],
                    "ResourceArn": "arn:aws:dynamodb:us-west-2:111122223333:table/
TableInUsWest2"
                },
                {
```

```
                    "ReadinessScopes": [
                        "arn:aws:route53-recovery-
readiness::&ExampleAWSAccountNo1;:cell/east-cell"
                    ],
                    "ResourceArn": "arn:aws:dynamodb:us-west-2:111122223333:table/
TableInUsEast1"
                }
            ],
            "Tags": {}
        }
    ]
}
```

Sekarang kita telah membuat sel, grup pemulihan, dan set sumber daya untuk memodelkan Simple-Service aplikasi di ARC. Selanjutnya, kita akan mengatur pemeriksaan kesiapan untuk memantau kesiapan sumber daya untuk gagal.

4. Buat pemeriksaan kesiapan

Pemeriksaan kesiapan menerapkan seperangkat aturan untuk setiap sumber daya dalam kumpulan sumber daya yang dilampirkan pada cek. Aturan khusus untuk setiap jenis sumber daya. Artinya, ada aturan yang berbeda untuk`AWS::DynamoDB::Table`,`AWS::EC2::Instance`, dan sebagainya. Aturan memeriksa berbagai dimensi untuk sumber daya, termasuk konfigurasi, kapasitas (jika tersedia dan berlaku), batas (jika tersedia dan berlaku), dan konfigurasi perutean.

> **Note**
>
> Untuk melihat aturan yang diterapkan ke sumber daya dalam pemeriksaan kesiapan, Anda dapat menggunakan `get-readiness-check-resource-status` API, seperti yang dijelaskan pada langkah 5. Untuk melihat daftar semua aturan kesiapan di ARC, gunakan `list-rules` atau lihatDeskripsi aturan kesiapan di ARC. ARC memiliki seperangkat aturan khusus yang dijalankan untuk setiap jenis sumber daya; mereka tidak dapat disesuaikan saat ini.

4a. Buat pemeriksaan kesiapan untuk kumpulan sumber daya, ImportantInformationTables.

```
aws route53-recovery-readiness --region us-west-2 create-readiness-check \
    --readiness-check-name ImportantInformationTableCheck --resource-set-name
 ImportantInformationTables
```

```
{
    "ReadinessCheckArn": "arn:aws:route53-recovery-readiness::111122223333:readiness-
check/ImportantInformationTableCheck",
    "ReadinessCheckName": "ImportantInformationTableCheck",
    "ResourceSet": "ImportantInformationTables",
    "Tags": {}
}
```

4b. (Opsional) Untuk memverifikasi bahwa pemeriksaan kesiapan berhasil dibuat, jalankan `list-readiness-checks` API. API ini menunjukkan semua pemeriksaan kesiapan di akun.

```
aws route53-recovery-readiness --region us-west-2 list-readiness-checks
```

```
{
    "ReadinessChecks": [
        {
            "ReadinessCheckArn": "arn:aws:route53-recovery-
readiness::111122223333:readiness-check/ImportantInformationTableCheck",
            "ReadinessCheckName": "ImportantInformationTableCheck",
            "ResourceSet": "ImportantInformationTables",
            "Tags": {}
        }
    ]
}
```

5. Memantau pemeriksaan kesiapan

Sekarang kita telah memodelkan aplikasi dan menambahkan pemeriksaan kesiapan, kita siap untuk memantau sumber daya. Anda dapat memodelkan kesiapan aplikasi Anda di empat tingkat: tingkat pemeriksaan kesiapan (sekelompok sumber daya), tingkat sumber daya individu, tingkat sel (semua sumber daya di Availability Zone atau Region), dan tingkat grup pemulihan (aplikasi secara keseluruhan). Perintah untuk mendapatkan masing-masing jenis status kesiapan ini disediakan di bawah ini.

5a. Lihat status pemeriksaan kesiapan Anda.

```
aws route53-recovery-readiness --region us-west-2 get-readiness-check-status\
    --readiness-check-name ImportantInformationTableCheck
```

```
{
```

```
    "Readiness": "READY",
    "Resources": [
        {
            "LastCheckedTimestamp": "2021-01-07T00:53:39Z",
            "Readiness": "READY",
            "ResourceArn": "arn:aws:dynamodb:us-west-2:111122223333:table/
TableInUsWest2"
        },
        {
            "LastCheckedTimestamp": "2021-01-07T00:53:39Z",
            "Readiness": "READY",
            "ResourceArn": "arn:aws:dynamodb:us-west-2:111122223333:table/
TableInUsEast2"
    ]
}
```

5b. Lihat status kesiapan terperinci dari satu sumber daya dalam pemeriksaan kesiapan, termasuk status setiap aturan yang dicentang.

```
aws route53-recovery-readiness --region us-west-2 get-readiness-check-resource-status \
    --readiness-check-name ImportantInformationTableCheck \
    --resource-identifier "arn:aws:dynamodb:us-west-2:111122223333:table/
TableInUsWest2"
```

```
{"Readiness": "READY",
    "Rules": [
        {
            "LastCheckedTimestamp": "2021-01-07T00:55:41Z",
            "Messages": [],
            "Readiness": "READY",
            "RuleId": "DynamoTableStatus"
        },
        {
            "LastCheckedTimestamp": "2021-01-07T00:55:41Z",
            "Messages": [],
            "Readiness": "READY",
            "RuleId": "DynamoCapacity"
        },
        {
            "LastCheckedTimestamp": "2021-01-07T00:55:41Z",
            "Messages": [],
            "Readiness": "READY",
            "RuleId": "DynamoPeakRcuWcu"
```

```
        },
        {
            "LastCheckedTimestamp": "2021-01-07T00:55:41Z",
            "Messages": [],
            "Readiness": "READY",
            "RuleId": "DynamoGSIsPeakRcuWcu"
        },
        {
            "LastCheckedTimestamp": "2021-01-07T00:55:41Z",
            "Messages": [],
            "Readiness": "READY",
            "RuleId": "DynamoGSIsConfig"
        },
        {
            "LastCheckedTimestamp": "2021-01-07T00:55:41Z",
            "Messages": [],
            "Readiness": "READY",
            "RuleId": "DynamoGSIsStatus"
        },
        {
            "LastCheckedTimestamp": "2021-01-07T00:55:41Z",
            "Messages": [],
            "Readiness": "READY",
            "RuleId": "DynamoGSIsCapacity"
        },
        {
            "LastCheckedTimestamp": "2021-01-07T00:55:41Z",
            "Messages": [],
            "Readiness": "READY",
            "RuleId": "DynamoReplicationLatency"
        },
        {
            "LastCheckedTimestamp": "2021-01-07T00:55:41Z",
            "Messages": [],
            "Readiness": "READY",
            "RuleId": "DynamoAutoScalingConfiguration"
        },
        {
            "LastCheckedTimestamp": "2021-01-07T00:55:41Z",
            "Messages": [],
            "Readiness": "READY",
            "RuleId": "DynamoLimits"
        }
    ]
```

```
}
```

5c. Lihat kesiapan keseluruhan untuk sel.

```
aws route53-recovery-readiness --region us-west-2 get-cell-readiness-summary \
    --cell-name west-cell
```

```
{
    "Readiness": "READY",
    "ReadinessChecks": [
        {
            "Readiness": "READY",
            "ReadinessCheckName": "ImportantTableCheck"
        }
    ]
}
```

5d. Terakhir, lihat kesiapan tingkat atas aplikasi Anda, di tingkat grup pemulihan.

```
aws route53-recovery-readiness --region us-west-2 get-recovery-group-readiness-summary
 \
    --recovery-group-name simple-service-recovery-group
```

```
{
    "Readiness": "READY",
    "ReadinessChecks": [
        {
            "Readiness": "READY",
            "ReadinessCheckName": "ImportantTableCheck"
        }
    ]
}
```

Bekerja dengan kelompok pemulihan dan pemeriksaan kesiapan

Bagian ini menjelaskan dan menyediakan prosedur untuk grup pemulihan dan pemeriksaan kesiapan, termasuk membuat, memperbarui, dan menghapus sumber daya ini.

Membuat, memperbarui, dan menghapus grup pemulihan di ARC

Grup pemulihan mewakili aplikasi Anda di Amazon Application Recovery Controller (ARC). Ini biasanya terdiri dari dua atau lebih sel yang merupakan replika satu sama lain dalam hal sumber daya dan fungsionalitas, sehingga Anda dapat gagal dari satu ke yang lain. Setiap sel menyertakan Amazon Resource Names (ARNs) untuk sumber daya aktif untuk satu AWS Wilayah atau Availability Zone. Sumber daya dapat berupa penyeimbang beban Elastic Load Balancing, grup Auto Scaling, atau sumber daya lainnya. Sel terkait yang mewakili zona atau Wilayah lain memiliki sumber daya siaga dari jenis yang sama yang ada di sel aktif Anda — penyeimbang beban, grup Auto Scaling, dan sebagainya.

Sel mewakili replika aplikasi Anda. Pemeriksaan kesiapan di ARC membantu Anda menentukan apakah aplikasi Anda siap gagal dari satu replika ke replika lainnya. Namun, Anda harus membuat keputusan tentang apakah akan gagal dari atau ke replika berdasarkan pemantauan dan sistem pemeriksaan kesehatan Anda, dan mempertimbangkan pemeriksaan kesiapan sebagai layanan pelengkap untuk sistem tersebut.

Kesiapan memeriksa sumber daya audit untuk menentukan kesiapan mereka berdasarkan seperangkat aturan yang telah ditentukan sebelumnya untuk jenis sumber daya tersebut. Setelah Anda membuat grup pemulihan dengan replika, Anda menambahkan pemeriksaan kesiapan ARC untuk sumber daya dalam aplikasi Anda, sehingga ARC dapat membantu memastikan bahwa replika memiliki pengaturan dan konfigurasi yang sama dari waktu ke waktu.

Topik

Membuat grup pemulihan

Langkah-langkah di bagian ini menjelaskan cara membuat grup pemulihan di konsol ARC. Untuk mempelajari tentang menggunakan operasi API kesiapan pemulihan dengan Amazon Application Recovery Controller (ARC), lihat Kesiapan memeriksa operasi API.

Untuk membuat grup pemulihan

1. Buka konsol ARC dihttps://console.aws.amazon.com/route53recovery/home#/dashboard.
2. Pilih Pemeriksaan Kesiapan.
3. Pada halaman Kesiapan pemulihan, pilih Buat, lalu pilih grup Pemulihan.

4. Masukkan nama untuk grup pemulihan Anda, lalu pilih Berikutnya.
5. Pilih Buat sel, lalu pilih Tambahkan sel.
6. Masukkan nama untuk sel. Misalnya, jika Anda memiliki replika aplikasi di US West (California N.), Anda dapat menambahkan sel bernama. `MyApp-us-west-1`
7. Pilih Tambahkan sel, dan tambahkan nama untuk sel kedua. Misalnya, jika Anda memiliki replika di US East (Ohio), Anda dapat menambahkan sel bernama. `MyApp-us-east-2`
8. Jika Anda ingin menambahkan sel bersarang (replika di Availability Zones dalam Regions), pilih Tindakan, pilih Tambahkan sel bersarang, lalu masukkan nama.
9. Ketika Anda telah menambahkan semua sel dan sel bersarang untuk replika aplikasi Anda, pilih Berikutnya.
10. Tinjau grup pemulihan Anda, lalu pilih Buat grup pemulihan.

Memperbarui dan menghapus grup dan sel pemulihan

Langkah-langkah di bagian ini menjelaskan cara memperbarui dan menghapus grup pemulihan, dan menghapus sel di konsol ARC. Untuk mempelajari tentang menggunakan operasi API kesiapan pemulihan dengan Amazon Application Recovery Controller (ARC), lihat Kesiapan memeriksa operasi API.

Untuk memperbarui atau menghapus grup pemulihan, atau menghapus sel

1. Buka konsol ARC dihttps://console.aws.amazon.com/route53recovery/home#/dashboard.
2. Pilih Pemeriksaan Kesiapan.
3. Pada halaman kesiapan pemulihan, pilih grup pemulihan.
4. Untuk bekerja dengan grup pemulihan, pilih Tindakan, lalu pilih Edit grup pemulihan atau Hapus grup pemulihan.
5. Saat mengedit grup pemulihan, Anda dapat menambahkan atau menghapus sel atau sel bersarang.
   - Untuk menambahkan sel, pilih Tambahkan sel.
   - Untuk menghapus sel, di bawah label Tindakan di sebelah sel, pilih Hapus sel.

Membuat dan memperbarui pemeriksaan kesiapan di ARC

Bagian ini menyediakan prosedur untuk pemeriksaan kesiapan dan kumpulan sumber daya, termasuk membuat, memperbarui, dan menghapus sumber daya ini.

Membuat dan memperbarui pemeriksaan kesiapan

Langkah-langkah di bagian ini menjelaskan cara membuat pemeriksaan kesiapan di konsol ARC. Untuk mempelajari tentang menggunakan operasi API kesiapan pemulihan dengan Amazon Application Recovery Controller (ARC), lihat Kesiapan memeriksa operasi API.

Untuk memperbarui pemeriksaan kesiapan, Anda dapat mengedit kumpulan sumber daya untuk pemeriksaan kesiapan, untuk menambah atau menghapus sumber daya atau untuk mengubah ruang lingkup kesiapan untuk sumber daya.

Untuk membuat pemeriksaan kesiapan

1. Buka konsol ARC dihttps://console.aws.amazon.com/route53recovery/home#/dashboard.
2. Pilih Pemeriksaan Kesiapan.
3. Pada halaman Kesiapan, pilih Buat, lalu pilih cek Kesiapan.
4. Masukkan nama untuk pemeriksaan kesiapan Anda, pilih jenis sumber daya yang ingin Anda periksa, lalu pilih Berikutnya.
5. Tambahkan kumpulan sumber daya untuk pemeriksaan kesiapan Anda. Kumpulan sumber daya adalah sekelompok sumber daya dari jenis yang sama dalam replika yang berbeda. Pilih salah satu cara berikut:

   - Buat pemeriksaan kesiapan dengan sumber daya dalam kumpulan sumber daya yang telah Anda buat.
   - Buat kumpulan sumber daya baru.

   Jika Anda memilih untuk membuat kumpulan sumber daya baru, masukkan nama untuk itu dan pilih Tambah.
6. Salin dan tempel Nama Sumber Daya Amazon (ARNs) satu per satu untuk setiap sumber daya yang ingin Anda sertakan dalam set, lalu pilih Berikutnya.

> **Tip**
>
> Untuk contoh dan informasi selengkapnya tentang format ARN yang diharapkan ARC untuk setiap jenis sumber daya, lihat. Jenis sumber daya dan format ARN di ARC

7. Jika Anda suka, lihat aturan kesiapan yang akan digunakan saat ARC memeriksa jenis sumber daya yang Anda sertakan dalam pemeriksaan kesiapan ini. Lalu pilih Berikutnya.
8. (Opsional) Di bawah nama grup Pemulihan, pilih grup pemulihan untuk mengaitkan pemeriksaan kesiapan dan kemudian, untuk setiap ARN sumber daya, pilih sel (Wilayah atau Zona Ketersediaan) dari menu tarik-turun tempat sumber daya berada. Jika sumber daya tingkat aplikasi, seperti kebijakan perutean DNS, pilih sumber daya global (tanpa sel).

   Ini menentukan cakupan kesiapan untuk sumber daya dalam pemeriksaan kesiapan.

   > **Important**
   >
   > Meskipun langkah ini opsional, cakupan kesiapan harus ditambahkan untuk mendapatkan informasi kesiapan ringkasan untuk grup dan sel pemulihan Anda. Jika Anda melewati langkah ini dan tidak mengaitkan pemeriksaan kesiapan dengan sumber daya grup pemulihan Anda dengan memilih cakupan kesiapan di sini, ARC tidak dapat mengembalikan informasi kesiapan ringkasan untuk grup atau sel pemulihan.

9. Pilih Berikutnya.
10. Tinjau informasi di halaman konfirmasi, lalu pilih Buat cek kesiapan.

Untuk menghapus cek kesiapan

1. Buka konsol ARC dihttps://console.aws.amazon.com/route53recovery/home#/dashboard.
2. Pilih Pemeriksaan Kesiapan.
3. Pilih pemeriksaan kesiapan, dan di bawah Tindakan, pilih Hapus.

Membuat dan mengedit kumpulan sumber daya

Biasanya, Anda membuat kumpulan sumber daya sebagai bagian dari membuat pemeriksaan kesiapan, tetapi Anda juga dapat membuat kumpulan sumber daya secara terpisah. Anda juga dapat mengedit kumpulan sumber daya untuk menambah atau menghapus sumber daya. Langkah-langkah

di bagian ini menjelaskan cara membuat atau mengedit kumpulan sumber daya di konsol ARC. Untuk mempelajari tentang menggunakan operasi API kesiapan pemulihan dengan Amazon Application Recovery Controller (ARC), lihat Kesiapan memeriksa operasi API.

Untuk membuat kumpulan sumber daya

1. Buka konsol Route 53 di https://console.aws.amazon.com/route53/rumah.
2. Di bawah Application Recovery Controller, pilih set Sumber Daya.
3. Pilih Buat.
4. Masukkan nama untuk kumpulan sumber daya, lalu pilih jenis sumber daya yang akan disertakan dalam kumpulan.
5. Pilih Tambah, lalu masukkan Nama Sumber Daya Amazon (ARN) untuk sumber daya yang akan ditambahkan ke set.
6. Setelah selesai menambahkan sumber daya, pilih Buat kumpulan sumber daya.

Untuk mengedit kumpulan sumber daya

1. Buka konsol ARC dihttps://console.aws.amazon.com/route53recovery/home#/dashboard.
2. Pilih Pemeriksaan Kesiapan.
3. Di bawah Set sumber daya, pilih Tindakan, lalu pilih Edit.
4. Lakukan salah satu hal berikut ini:
   - Untuk menghapus sumber daya dari set, pilih Hapus.
   - Untuk menambahkan sumber daya ke set, pilih Tambah, lalu masukkan Nama Sumber Daya Amazon (ARN) untuk sumber daya.
5. Anda juga dapat mengedit ruang lingkup kesiapan untuk sumber daya, untuk mengaitkan sumber daya dengan sel yang berbeda untuk pemeriksaan kesiapan.
6. Pilih Simpan.

Memantau status kesiapan di ARC

Anda dapat melihat kesiapan untuk aplikasi Anda di Amazon Application Recovery Controller (ARC) pada level berikut:

- Tingkat pemeriksaan kesiapan untuk sumber daya dalam kumpulan sumber daya

- Tingkat sumber daya individu
- Level sel (replika aplikasi) untuk semua sumber daya di Availability Zone atau AWS Region
- Tingkat kelompok pemulihan untuk aplikasi secara keseluruhan

Anda dapat diberi tahu tentang perubahan status kesiapan, atau Anda dapat memantau perubahan status kesiapan di konsol Route 53 atau dengan menggunakan perintah ARC CLI.

Pemberitahuan status kesiapan

Anda dapat menggunakan Amazon EventBridge untuk menyiapkan aturan berbasis peristiwa untuk memantau sumber daya ARC dan memberi tahu Anda tentang perubahan status kesiapan. Untuk informasi selengkapnya, lihat Menggunakan pemeriksaan kesiapan di ARC dengan Amazon EventBridge.

Memantau status kesiapan di konsol ARC

Prosedur berikut menjelaskan cara memantau kesiapan pemulihan di. AWS Management Console

1. Buka konsol ARC dihttps://console.aws.amazon.com/route53recovery/home#/dashboard.
2. Pilih Pemeriksaan Kesiapan.
3. Pada halaman Kesiapan, di bawah grup Pemulihan, lihat status kesiapan grup Pemulihan untuk setiap grup pemulihan (aplikasi).

   Anda juga dapat melihat kesiapan sel tertentu atau sumber daya individu.

Memantau status kesiapan dengan menggunakan perintah CLI

Bagian ini memberikan contoh AWS CLI perintah yang akan digunakan untuk melihat status kesiapan untuk aplikasi dan sumber daya Anda pada tingkat yang berbeda.

Kesiapan untuk satu set sumber daya

Status pemeriksaan kesiapan yang Anda buat untuk kumpulan sumber daya (sekelompok sumber daya).

```
aws route53-recovery-readiness --region us-west-2 get-readiness-check-
status --readiness-check-name ReadinessCheckName
```

Kesiapan untuk satu sumber daya

Untuk mendapatkan status sumber daya tunggal dalam pemeriksaan kesiapan, termasuk status setiap aturan kesiapan yang diperiksa, tentukan nama pemeriksaan kesiapan dan ARN sumber daya. Sebagai contoh:

```
aws route53-recovery-readiness --region us-west-2 get-readiness-
check-status --readiness-check-name ReadinessCheckName --resource-arn
"arn:aws:dynamodb:us-west-2:111122223333:table/TableName"
```

Kesiapan untuk sel

Status sel tunggal, yaitu Wilayah atau Zona Ketersediaan.

```
aws route53-recovery-readiness --region us-west-2 get-cell-readiness-
summary --cell-name CellName
```

Kesiapan untuk aplikasi

Status aplikasi keseluruhan, di tingkat kelompok pemulihan.

```
aws route53-recovery-readiness --region us-west-2 get-recovery-group-
readiness-summary --recovery-group-name RecoveryGroupName
```

Mendapatkan rekomendasi arsitektur di ARC

Jika Anda memiliki aplikasi yang sudah ada, Amazon Application Recovery Controller (ARC) dapat mengevaluasi arsitektur aplikasi dan kebijakan perutean Anda untuk memberikan rekomendasi untuk memodifikasi desain guna meningkatkan ketahanan pemulihan aplikasi Anda. Setelah Anda membuat grup pemulihan di ARC yang mewakili aplikasi Anda, ikuti langkah-langkah di bagian ini untuk mendapatkan rekomendasi untuk arsitektur aplikasi Anda.

Kami menyarankan Anda menentukan sumber daya target untuk sumber daya target DNS untuk grup pemulihan Anda, jika Anda belum menentukannya, sehingga kami dapat memberikan rekomendasi yang lebih rinci. Ketika Anda memberikan informasi tambahan, ARC dapat memberikan rekomendasi yang lebih baik untuk Anda. Misalnya, jika Anda memasukkan catatan sumber daya Amazon Route 53 atau Network Load Balancer sebagai sumber daya target, ARC dapat memberikan informasi tentang apakah Anda telah membuat jumlah sel optimal untuk grup pemulihan Anda.

Perhatikan hal berikut untuk sumber daya target DNS:

- Tentukan hanya catatan sumber daya Route 53 atau Network Load Balancer untuk sumber daya target.
- Buat hanya satu sumber daya target DNS untuk setiap grup pemulihan.
- Direkomendasikan: Buat satu sumber daya target DNS untuk setiap sel.
- Kelompokkan sumber daya target DNS ke dalam satu set sumber daya dengan pemeriksaan kesiapan.

Prosedur berikut menjelaskan cara membuat sumber daya target DNS dan mendapatkan rekomendasi arsitektur untuk aplikasi Anda.

Untuk mendapatkan rekomendasi untuk memperbarui arsitektur Anda

1. Buka konsol ARC dihttps://console.aws.amazon.com/route53recovery/home#/dashboard.
2. Pilih Pemeriksaan Kesiapan.
3. Di bawah nama grup Pemulihan, pilih grup pemulihan yang mewakili aplikasi Anda.
4. Pada halaman Recovery group details, pada menu Action, pilih Dapatkan rekomendasi arsitektur untuk grup pemulihan ini.
5. Jika Anda belum membuat pemeriksaan kesiapan sumber daya target DNS, buat satu sehingga ARC dapat memberikan rekomendasi arsitektur. Pilih Buat sumber daya target DNS.

   Untuk informasi selengkapnya tentang sumber daya target DNS, lihatKomponen pemeriksaan kesiapan .
6. Untuk membuat kumpulan sumber daya untuk sumber daya target DNS, Anda membuat pemeriksaan kesiapan. Masukkan nama untuk pemeriksaan kesiapan, dan kemudian, untuk jenis pemeriksaan kesiapan, pilih sumber daya target DNS.
7. Masukkan nama untuk kumpulan sumber daya.
8. Masukkan atribut untuk aplikasi Anda, termasuk nama DNS, ARN zona yang dihosting, dan ID set rekaman.

> **Tip**
>
> Untuk melihat format ARN zona yang dihosting, lihat format ARN untuk zona yang dihosting di. Jenis sumber daya dan format ARN di ARC

Secara opsional, tetapi sangat disarankan, pilih Tambahkan atribut opsional dan berikan ARN Network Load Balancer atau catatan sumber daya Route 53 domain Anda.

9. (Opsional) Dalam konfigurasi grup Pemulihan, pilih sel untuk sumber daya target DNS Anda, untuk mengatur cakupan kesiapan.
10. Pilih Buat kumpulan sumber daya.
11. Pada halaman Recovery group details, pilih Dapatkan rekomendasi arsitektur. ARC menampilkan serangkaian rekomendasi pada halaman.

Tinjau daftar rekomendasi. Kemudian Anda dapat memutuskan apakah dan bagaimana membuat perubahan untuk meningkatkan ketahanan pemulihan aplikasi Anda.

Membuat otorisasi lintas akun di ARC

Anda mungkin memiliki sumber daya yang didistribusikan di beberapa AWS akun, yang dapat membuatnya sulit untuk mendapatkan pandangan komprehensif tentang kesehatan aplikasi Anda. Hal ini juga dapat membuat sulit untuk mendapatkan informasi yang diperlukan untuk membuat keputusan cepat. Untuk membantu merampingkan ini untuk pemeriksaan kesiapan di Amazon Application Recovery Controller (ARC), Anda dapat menggunakan otorisasi lintas akun.

Otorisasi lintas akun di ARC berfungsi dengan fitur pemeriksaan kesiapan. Dengan otorisasi lintas akun, Anda dapat menggunakan satu AWS akun pusat untuk memantau sumber daya Anda yang terletak di beberapa AWS akun. Di setiap akun yang memiliki sumber daya yang ingin Anda pantau, Anda mengotorisasi akun pusat untuk memiliki akses ke sumber daya tersebut. Kemudian akun pusat dapat membuat pemeriksaan kesiapan untuk sumber daya di semua akun dan dari akun pusat, Anda dapat memantau kesiapan untuk failover.

> **Note**
>
> Pengaturan otorisasi lintas akun tidak tersedia di konsol. Sebagai gantinya, gunakan operasi ARC API untuk menyiapkan dan bekerja dengan otorisasi lintas akun. Untuk membantu Anda memulai, bagian ini memberikan contoh AWS CLI perintah.

Katakanlah aplikasi memiliki akun yang memiliki sumber daya di Wilayah AS Barat (Oregon) (us-west-2), dan ada juga akun yang memiliki sumber daya yang ingin Anda pantau di Wilayah AS Timur

(Virginia N.) (us-east-1). ARC dapat memungkinkan akses bagi Anda untuk memantau kedua set sumber daya dari satu akun, us-west-2, dengan menggunakan otorisasi lintas akun.

Misalnya, katakanlah Anda memiliki AWS akun berikut:

- Akun AS-Barat: 99999999999999
- Akun AS-Timur: 111111111111

Di akun us-east-1 (11111111111111), kami dapat mengaktifkan otorisasi lintas akun untuk mengizinkan akses oleh akun us-west-2 (9999999999999999) dengan menentukan Nama Sumber Daya Amazon (ARN) untuk pengguna (root) di akun IAM us-west-2:. `arn:aws:iam::999999999999:root` Setelah kami membuat otorisasi, akun us-west-2 dapat menambahkan sumber daya yang dimiliki oleh us-east-1 ke kumpulan sumber daya dan membuat pemeriksaan kesiapan untuk dijalankan pada kumpulan sumber daya.

Contoh berikut menggambarkan pengaturan otorisasi lintas akun untuk satu akun. Anda harus mengaktifkan otorisasi lintas akun di setiap akun tambahan yang memiliki AWS sumber daya yang ingin Anda tambahkan dan pantau di ARC.

> **Note**
>
> ARC adalah layanan global yang mendukung titik akhir di beberapa AWS Wilayah tetapi Anda harus menentukan Wilayah AS Barat (Oregon) (yaitu, tentukan parameter`--region us-west-2`) di sebagian besar perintah ARC CLI.

AWS CLI Perintah berikut menunjukkan cara mengatur otorisasi lintas akun untuk contoh ini:

```
aws route53-recovery-readiness --region us-west-2 --profile profile-in-us-east-1-
account \
    create-cross-account-authorization --cross-account-authorization
 arn:aws:iam::999999999999:root
```

Untuk menonaktifkan otorisasi ini, lakukan hal berikut:

```
aws route53-recovery-readiness --region us-west-2 --profile profile-in-us-east-1-
account \
    delete-cross-account-authorization --cross-account-authorization
 arn:aws:iam::999999999999:root
```

Untuk memeriksa akun tertentu untuk semua akun yang telah Anda berikan otorisasi lintas akun, gunakan perintah. `list-cross-account-authorizations` Perhatikan bahwa saat ini, Anda tidak dapat memeriksa ke arah lain. Artinya, tidak ada operasi API yang dapat Anda gunakan dengan profil akun untuk mencantumkan semua akun yang telah diberikan otorisasi lintas akun untuk menambah dan memantau sumber daya.

```
aws route53-recovery-readiness --region us-west-2 --profile profile-in-us-east-1-
account \
    list-cross-account-authorizations
```

```
{
    "CrossAccountAuthorizations": [
        "arn:aws:iam::999999999999:root"
    ]
}
```

Aturan kesiapan, jenis sumber daya, dan ARNS

Bagian ini mencakup informasi referensi tentang deskripsi aturan kesiapan, dan jenis sumber daya yang didukung serta format untuk Amazon Resource Names (ARNs) yang Anda gunakan untuk kumpulan sumber daya.

Deskripsi aturan kesiapan di ARC

Bagian ini mencantumkan deskripsi aturan kesiapan untuk semua jenis sumber daya yang didukung oleh Amazon Application Recovery Controller (ARC). Untuk melihat daftar jenis sumber daya yang didukung oleh ARC, lihatJenis sumber daya dan format ARN di ARC.

Anda juga dapat melihat deskripsi aturan kesiapan di konsol ARC atau dengan menggunakan operasi API, dengan melakukan hal berikut:

- Untuk melihat aturan kesiapan di konsol, ikuti langkah-langkah dalam prosedur berikut:Lihat aturan kesiapan di konsol.
- Untuk melihat aturan kesiapan menggunakan API, lihat ListRulesoperasinya.

Topik

- Aturan kesiapan di ARC

- Lihat aturan kesiapan di konsol

Aturan kesiapan di ARC

Bagian ini mencantumkan kumpulan aturan kesiapan untuk setiap jenis sumber daya yang didukung oleh ARC.

Saat Anda melihat deskripsi aturan, Anda dapat melihat bahwa sebagian besar dari mereka menyertakan istilah Memeriksa semua atau Memeriksa masing-masing. Untuk memahami bagaimana istilah ini menjelaskan cara kerja aturan dalam konteks pemeriksaan kesiapan, dan detail lainnya tentang cara ARC menetapkan status kesiapan, lihat Bagaimana aturan kesiapan menentukan status kesiapan.

Aturan kesiapan

ARC mengaudit sumber daya dengan menggunakan aturan kesiapan berikut.

Amazon API Gateway Versi 1 tahap

- ApiGwV1ApiKeyCount: Memeriksa semua tahapan API Gateway untuk memastikan bahwa mereka memiliki jumlah Kunci API yang sama yang ditautkan ke mereka.
- ApiGwV1ApiKeySource: Memeriksa semua tahapan API Gateway untuk memastikan bahwa mereka memiliki nilai yang sama. `API Key Source`
- ApiGwV1BasePath: Memeriksa semua tahapan API Gateway untuk memastikan bahwa mereka ditautkan ke jalur dasar yang sama.
- ApiGwV1BinaryMediaTypes: Memeriksa semua tahap API Gateway untuk memastikan bahwa mereka mendukung jenis media biner yang sama.
- ApiGwV1CacheClusterEnabled: Memeriksa semua tahapan API Gateway untuk memastikan bahwa semua telah `Cache Cluster` diaktifkan, atau tidak ada yang melakukannya.
- ApiGwV1CacheClusterSize: Memeriksa semua tahap API Gateway untuk memastikan bahwa mereka memiliki yang sama`Cache Cluster Size`. Jika satu memiliki nilai yang lebih besar, yang lain ditandai TIDAK SIAP.
- ApiGwV1CacheClusterStatus: Memeriksa semua tahapan API Gateway untuk memastikan `Cache Cluster` bahwa dalam status TERSEDIA.
- ApiGwV1DisableExecuteApiEndpoint: Memeriksa semua tahapan API Gateway untuk memastikan bahwa semua telah `Execute API Endpoint` dinonaktifkan, atau tidak ada yang melakukannya.

- ApiGwV1DomainName: Memeriksa semua tahap API Gateway untuk memastikan bahwa mereka ditautkan ke nama domain yang sama.
- ApiGwV1EndpointConfiguration: Memeriksa semua tahapan API Gateway untuk memastikan bahwa mereka ditautkan ke domain dengan konfigurasi titik akhir yang sama.
- ApiGwV1EndpointDomainNameStatus: Memeriksa semua tahapan API Gateway untuk memastikan bahwa nama domain yang ditautkan berada dalam status TERSEDIA.
- ApiGwV1MethodSettings: Memeriksa semua tahapan API Gateway untuk memastikan bahwa mereka memiliki nilai yang sama. `Method Settings`
- ApiGwV1MutualTlsAuthentication: Memeriksa semua tahapan API Gateway untuk memastikan bahwa mereka memiliki nilai yang sama. `Mutual TLS Authentication`
- ApiGwV1Policy: Memeriksa semua tahapan API Gateway untuk memastikan bahwa semua menggunakan kebijakan tingkat API, atau tidak ada yang melakukannya.
- ApiGwV1RegionalDomainName: Memeriksa semua tahapan API Gateway untuk memastikan bahwa mereka ditautkan ke nama domain Regional yang sama. Catatan: Aturan ini tidak mempengaruhi status kesiapan.
- ApiGwV1ResourceMethodConfigs: Memeriksa semua tahapan API Gateway untuk memastikan bahwa mereka memiliki hierarki sumber daya yang serupa, termasuk konfigurasi terkait.
- ApiGwV1SecurityPolicy: Memeriksa semua tahapan API Gateway untuk memastikan bahwa mereka memiliki nilai yang sama. `Security Policy`
- ApiGwV1Quotas: Memeriksa semua grup API Gateway untuk memastikan bahwa mereka sesuai dengan kuota (batas) yang dikelola oleh Service Quotas.
- ApiGwV1UsagePlans: Memeriksa semua tahapan API Gateway untuk memastikan bahwa mereka ditautkan `Usage Plans` dengan konfigurasi yang sama.

Amazon API Gateway Versi 2 tahap

- ApiGwV2ApiKeySelectionExpression: Memeriksa semua tahap API Gateway memastikan bahwa mereka memiliki nilai yang sama untuk`API Key Selection Expression`.
- ApiGwV2ApiMappingSelectionExpression: Memeriksa semua tahapan API Gateway untuk memastikan bahwa mereka memiliki nilai yang sama. `API Mapping Selection Expression`
- ApiGwV2CorsConfiguration: Memeriksa semua tahapan API Gateway untuk memastikan bahwa mereka memiliki konfigurasi terkait CORS yang sama.
- ApiGwV2DomainName: Memeriksa semua tahap API Gateway untuk memastikan bahwa mereka ditautkan ke nama domain yang sama.

- ApiGwV2DomainNameStatus: Memeriksa semua tahapan API Gateway untuk memastikan bahwa nama domain berada dalam status TERSEDIA.
- ApiGwV2EndpointType: Memeriksa semua tahapan API Gateway untuk memastikan bahwa mereka memiliki nilai yang sama. `Endpoint Type`
- ApiGwV2Quotas: Memeriksa semua grup API Gateway untuk memastikan bahwa mereka sesuai dengan kuota (batas) yang dikelola oleh Service Quotas.
- ApiGwV2MutualTlsAuthentication: Memeriksa semua tahapan API Gateway untuk memastikan bahwa mereka memiliki nilai yang sama. `Mutual TLS Authentication`
- ApiGwV2ProtocolType: Memeriksa semua tahapan API Gateway untuk memastikan bahwa mereka memiliki nilai yang sama. `Protocol Type`
- ApiGwV2RouteConfigs: Memeriksa semua tahap API Gateway untuk memastikan bahwa mereka memiliki hierarki rute yang sama dengan konfigurasi yang sama.
- ApiGwV2RouteSelectionExpression: Memeriksa semua tahapan API Gateway untuk memastikan bahwa mereka memiliki nilai yang sama. `Route Selection Expression`
- ApiGwV2RouteSettings: Memeriksa semua tahapan API Gateway untuk memastikan bahwa mereka memiliki nilai yang sama. `Default Route Settings`
- ApiGwV2SecurityPolicy: Memeriksa semua tahapan API Gateway untuk memastikan bahwa mereka memiliki nilai yang sama. `Security Policy`
- ApiGwV2StageVariables: Memeriksa semua tahap API Gateway untuk memastikan bahwa semuanya memiliki tahap yang `Stage Variables` sama dengan tahapan lainnya.
- ApiGwV2ThrottlingBurstLimit: Memeriksa semua tahapan API Gateway untuk memastikan bahwa mereka memiliki nilai yang sama. `Throttling Burst Limit`
- ApiGwV2ThrottlingRateLimit: Memeriksa semua tahapan API Gateway untuk memastikan bahwa mereka memiliki nilai yang sama. `Throttling Rate Limit`

Cluster Amazon Aurora

- RdsClusterStatus: Memeriksa setiap cluster Aurora untuk memastikan bahwa ia memiliki status salah `AVAILABLE` satu atau. `BACKING-UP`
- RdsEngineMode: Memeriksa semua cluster Aurora untuk memastikan bahwa mereka memiliki nilai yang sama. `Engine Mode`
- RdsEngineVersion: Memeriksa semua cluster Aurora untuk memastikan bahwa mereka memiliki nilai yang sama. `Major Version`
- RdsGlobalReplicaLag: Memeriksa setiap cluster Aurora untuk memastikan bahwa ia `Global Replica Lag` memiliki waktu kurang dari 30 detik.

- RdsNormalizedCapacity: Memeriksa semua cluster Aurora untuk memastikan bahwa mereka memiliki kapasitas yang dinormalisasi dalam 15% dari maksimum dalam kumpulan sumber daya.
- RdsInstanceType: Memeriksa semua cluster Aurora untuk memastikan bahwa mereka memiliki tipe instance yang sama.
- RdsQuotasMemeriksa semua cluster Aurora untuk memastikan bahwa mereka sesuai dengan kuota (limit) yang dikelola oleh Service Quotas.

Grup Auto Scaling

- AsgMinSizeAndMaxSize: Memeriksa semua grup Auto Scaling untuk memastikan bahwa mereka memiliki ukuran grup minimum dan maksimum yang sama.
- AsgAZCount: Memeriksa semua grup Auto Scaling untuk memastikan bahwa mereka memiliki jumlah Availability Zone yang sama.
- AsgInstanceTypes: Memeriksa semua grup Auto Scaling untuk memastikan bahwa mereka memiliki tipe instans yang sama. Catatan: Aturan ini tidak mempengaruhi status kesiapan.
- AsgInstanceSizes: Memeriksa semua grup Auto Scaling untuk memastikan bahwa mereka memiliki ukuran instans yang sama.
- AsgNormalizedCapacity: Memeriksa semua grup Auto Scaling untuk memastikan bahwa mereka memiliki kapasitas yang dinormalisasi dalam 15% dari maksimum dalam kumpulan sumber daya.
- AsgQuotas: Memeriksa semua grup Auto Scaling untuk memastikan bahwa mereka sesuai dengan kuota (limit) yang dikelola oleh Service Quotas.

CloudWatch alarm

- CloudWatchAlarmState: Memeriksa CloudWatch alarm untuk memastikan bahwa masing-masing tidak dalam `ALARM` atau `INSUFFICIENT_DATA` negara bagian.

Gateway pelanggan

- CustomerGatewayIpAddressMemeriksa semua gateway pelanggan untuk memastikan bahwa mereka memiliki alamat IP yang sama.
- CustomerGatewayState: Memeriksa gateway pelanggan untuk memastikan bahwa masing-masing berada di negara bagian. `AVAILABLE`
- CustomerGatewayVPNTypeMemeriksa semua gateway pelanggan untuk memastikan bahwa mereka memiliki jenis VPN yang sama.

DNS target resources

- DnsTargetResourceHostedZoneConfigurationRule: Memeriksa semua sumber daya target DNS untuk memastikan bahwa mereka memiliki ID zona yang dihosting Amazon Route 53 yang sama dan bahwa setiap zona yang dihosting tidak bersifat pribadi. Catatan: Aturan ini tidak mempengaruhi status kesiapan.
- DnsTargetResourceRecordSetConfigurationRule: Memeriksa semua sumber daya target DNS untuk memastikan bahwa mereka memiliki waktu cache catatan sumber daya yang sama untuk hidup (TTL) dan kurang dari atau sama dengan 300. TTLs
- DnsTargetResourceRoutingRule: Memeriksa setiap sumber daya target DNS yang terkait dengan kumpulan catatan sumber daya alias untuk memastikan bahwa ia merutekan lalu lintas ke nama DNS yang dikonfigurasi pada sumber daya target. Catatan: Aturan ini tidak mempengaruhi status kesiapan.
- DnsTargetResourceHealthCheckRuleMemeriksa semua sumber daya target DNS untuk memastikan bahwa pemeriksaan kesehatan terkait dengan kumpulan catatan sumber daya mereka bila sesuai dan bukan sebaliknya. Catatan: Aturan ini tidak mempengaruhi status kesiapan.

Tabel Amazon DynamoDB

- DynamoConfiguration: Memeriksa semua tabel DynamoDB untuk memastikan bahwa mereka memiliki kunci, atribut, enkripsi sisi server, dan konfigurasi aliran yang sama.
- DynamoTableStatus: Memeriksa setiap tabel DynamoDB untuk memastikan bahwa ia memiliki status AKTIF.
- DynamoCapacity: Memeriksa semua tabel DynamoDB untuk memastikan bahwa kapasitas baca dan kapasitas tulis yang disediakan berada dalam 20% dari kapasitas maksimum dalam kumpulan sumber daya.
- DynamoPeakRcuWcu: Memeriksa setiap tabel DynamoDB untuk memastikan bahwa ia memiliki lalu lintas puncak yang sama dengan tabel lain, untuk memastikan kapasitas yang disediakan.
- DynamoGsiPeakRcuWcu: Memeriksa setiap tabel DynamoDB untuk memastikan bahwa ia memiliki kapasitas baca dan tulis maksimum yang serupa dengan tabel lainnya, untuk memastikan kapasitas yang disediakan.
- DynamoGsiConfig: Memeriksa semua tabel DynamoDB yang memiliki indeks sekunder global untuk memastikan bahwa tabel menggunakan indeks, skema kunci, dan proyeksi yang sama.
- DynamoGsiStatus: Memeriksa semua tabel DynamoDB yang memiliki indeks sekunder global untuk memastikan bahwa indeks sekunder global memiliki status AKTIF.

- DynamoGsiCapacity: Memeriksa semua tabel DynamoDB yang memiliki indeks sekunder global untuk memastikan bahwa tabel telah menyediakan kapasitas baca GSI dan kapasitas tulis GSI dalam 20% dari kapasitas maksimum dalam kumpulan sumber daya.
- DynamoReplicationLatency: Memeriksa semua tabel DynamoDB yang merupakan tabel global untuk memastikan bahwa mereka memiliki latensi replikasi yang sama.
- DynamoAutoScalingConfiguration: Memeriksa semua tabel DynamoDB yang mengaktifkan Auto Scaling untuk memastikan bahwa tabel tersebut memiliki kapasitas baca dan tulis minimum, maksimum, dan target yang sama.
- DynamoQuotas: Memeriksa semua tabel DynamoDB untuk memastikan bahwa mereka sesuai dengan kuota (batas) yang dikelola oleh Service Quotas.

Elastic Load Balancing (Classic Load Balancers)

- ElbV1CheckAzCount: Memeriksa setiap Classic Load Balancer untuk memastikan bahwa itu terpasang hanya pada satu Availability Zone. Catatan: Aturan ini tidak mempengaruhi status kesiapan.
- ElbV1AnyInstances: Memeriksa semua Classic Load Balancer untuk memastikan bahwa mereka memiliki setidaknya satu instance. EC2
- ElbV1AnyInstancesHealthy: Memeriksa semua Classic Load Balancer untuk memastikan bahwa mereka memiliki setidaknya satu instance sehat. EC2
- ElbV1Scheme: Memeriksa semua Classic Load Balancer untuk memastikan bahwa mereka memiliki skema penyeimbang beban yang sama.
- ElbV1HealthCheckThreshold: Memeriksa semua Classic Load Balancer untuk memastikan bahwa mereka memiliki nilai ambang pemeriksaan kesehatan yang sama.
- ElbV1HealthCheckInterval: Memeriksa semua Classic Load Balancer untuk memastikan bahwa mereka memiliki nilai interval pemeriksaan kesehatan yang sama.
- ElbV1CrossZoneRoutingEnabled: Memeriksa semua Classic Load Balancer untuk memastikan bahwa mereka memiliki nilai yang sama untuk penyeimbangan beban lintas zona (ENABLED atau DISABLED).
- ElbV1AccessLogsEnabledAttribute: Memeriksa semua Classic Load Balancer untuk memastikan bahwa mereka memiliki nilai yang sama untuk log akses (ENABLED atau DISABLED).
- ElbV1ConnectionDrainingEnabledAttribute: Memeriksa semua Classic Load Balancer untuk memastikan bahwa mereka memiliki nilai yang sama untuk pengurasan koneksi (DIAKTIFKAN atau DINONAKTIFKAN).

- ElbV1ConnectionDrainingTimeoutAttribute: Memeriksa semua Classic Load Balancer untuk memastikan bahwa mereka memiliki nilai batas waktu pengurasan koneksi yang sama.
- ElbV1IdleTimeoutAttribute: Memeriksa semua Classic Load Balancer untuk memastikan bahwa mereka memiliki nilai yang sama untuk batas waktu idle.
- ElbV1ProvisionedCapacityLcuCount: Memeriksa semua Classic Load Balancer dengan LCU yang disediakan lebih besar dari 10 untuk memastikan bahwa mereka berada dalam 20% dari LCU penyediaan tertinggi dalam kumpulan sumber daya.
- ElbV1ProvisionedCapacityStatus: Memeriksa status kapasitas yang disediakan pada setiap Classic Load Balancer untuk memastikan bahwa kapasitas tersebut tidak memiliki nilai DISABLED atau PENDING.

Volume Amazon EBS

- EbsVolumeEncryption: Memeriksa semua EBS volume untuk memastikan bahwa mereka memiliki nilai yang sama untuk enkripsi (ENABLED atau DISABLED).
- EbsVolumeEncryptionDefault: Memeriksa semua EBS volume untuk memastikan bahwa mereka memiliki nilai yang sama untuk enkripsi secara default (ENABLED atau DISABLED).
- EbsVolumeIops: Memeriksa semua EBS volume untuk memastikan bahwa mereka memiliki operasi input/output per detik (IOPS) yang sama.
- EbsVolumeKmsKeyId: Memeriksa semua EBS volume untuk memastikan bahwa mereka memiliki ID AWS KMS kunci default yang sama.
- EbsVolumeMultiAttach: Memeriksa semua EBS volume untuk memastikan bahwa mereka memiliki nilai yang sama untuk multi-attach (ENABLED atau DISABLED).
- EbsVolumeQuotas: Memeriksa semua EBS volume untuk memastikan bahwa mereka sesuai dengan kuota (batas) yang ditetapkan oleh Service Quotas.
- EbsVolumeSize: Memeriksa semua EBS volume untuk memastikan bahwa mereka memiliki ukuran yang dapat dibaca yang sama.
- EbsVolumeState: Memeriksa semua EBS volume untuk memastikan bahwa mereka memiliki status volume yang sama.
- EbsVolumeType: Memeriksa semua EBS volume untuk memastikan bahwa mereka memiliki jenis volume yang sama.

AWS Lambda fungsi

- LambdaMemorySize: Memeriksa semua fungsi Lambda untuk memastikan bahwa mereka memiliki ukuran memori yang sama. Jika seseorang memiliki lebih banyak memori, yang lain ditandai`NOT READY`.

- LambdaFunctionTimeout: Memeriksa semua fungsi Lambda untuk memastikan bahwa mereka memiliki nilai batas waktu yang sama. Jika satu memiliki nilai yang lebih besar, yang lain ditandai`NOT READY`.
- LambdaFunctionRuntime: Memeriksa semua fungsi Lambda untuk memastikan bahwa semuanya memiliki runtime yang sama.
- LambdaFunctionReservedConcurrentExecutions: Memeriksa semua fungsi Lambda untuk memastikan bahwa semuanya memiliki nilai yang sama. `Reserved Concurrent Executions` Jika satu memiliki nilai yang lebih besar, yang lain ditandai`NOT READY`.
- LambdaFunctionDeadLetterConfig: Memeriksa semua fungsi Lambda untuk memastikan bahwa semuanya `Dead Letter Config` memiliki definisi, atau tidak ada yang melakukannya.
- LambdaFunctionProvisionedConcurrencyConfig: Memeriksa semua fungsi Lambda untuk memastikan bahwa mereka memiliki nilai yang sama untuk. `Provisioned Concurrency`
- LambdaFunctionSecurityGroupCount: Memeriksa semua fungsi Lambda untuk memastikan bahwa mereka memiliki nilai yang sama untuk. `Security Groups`
- LambdaFunctionSubnetIdCount: Memeriksa semua fungsi Lambda untuk memastikan bahwa mereka memiliki nilai yang sama untuk. `Subnet Ids`
- LambdaFunctionEventSourceMappingMatch: Memeriksa semua fungsi Lambda untuk memastikan bahwa semua properti yang `Event Source Mapping` dipilih cocok di antara mereka.
- LambdaFunctionLimitsRule: Memeriksa semua fungsi Lambda untuk memastikan bahwa mereka sesuai dengan kuota (batas) yang dikelola oleh Service Quotas.

Network Load Balancers dan Application Load Balancer

- ElbV2CheckAzCount: Memeriksa setiap Network Load Balancer untuk memastikan bahwa itu terpasang hanya pada satu Availability Zone. Catatan: Aturan ini tidak mempengaruhi status kesiapan.
- ElbV2TargetGroupsCanServeTraffic: Memeriksa setiap Network Load Balancer dan Application Load Balancer untuk memastikan bahwa ia memiliki setidaknya satu instans Amazon yang sehat. EC2
- ElbV2State: Memeriksa setiap Network Load Balancer dan Application Load Balancer untuk memastikan bahwa itu dalam keadaan. `ACTIVE`
- ElbV2IpAddressTypeMemeriksa semua Network Load Balancer dan Application Load Balancer untuk memastikan bahwa mereka memiliki jenis alamat IP yang sama.
- ElbV2SchemeMemeriksa semua Network Load Balancer dan Application Load Balancer untuk memastikan bahwa mereka memiliki skema yang sama.

- ElbV2Type: Memeriksa semua Network Load Balancer dan Application Load Balancer untuk memastikan bahwa mereka memiliki tipe yang sama.
- ElbV2S3LogsEnabled: Memeriksa semua Network Load Balancer dan Application Load Balancer untuk memastikan bahwa mereka memiliki nilai yang sama untuk log akses server Amazon S3 (DIAKTIFKAN atau DINONAKTIFKAN).
- ElbV2DeletionProtection: Memeriksa semua Network Load Balancer dan Application Load Balancer untuk memastikan bahwa mereka memiliki nilai yang sama untuk perlindungan penghapusan (ENABLED atau DISABLED).
- ElbV2IdleTimeoutSeconds: Memeriksa semua Network Load Balancer dan Application Load Balancer untuk memastikan bahwa mereka memiliki nilai yang sama untuk detik waktu idle.
- ElbV2HttpDropInvalidHeaders: Memeriksa semua Network Load Balancer dan Application Load Balancer untuk memastikan bahwa mereka memiliki nilai yang sama untuk HTTP drop header yang tidak valid.
- ElbV2Http2Enabled: Memeriksa semua Network Load Balancer dan Application Load Balancer untuk memastikan bahwa mereka memiliki nilai yang sama untuk HTTP2 (DIAKTIFKAN atau DINONAKTIFKAN).
- ElbV2CrossZoneEnabled: Memeriksa semua Network Load Balancer dan Application Load Balancer untuk memastikan bahwa mereka memiliki nilai yang sama untuk penyeimbangan beban lintas zona (ENABLED atau DISABLED).
- ElbV2ProvisionedCapacityLcuCount: Memeriksa semua Network Load Balancer dan Application Load Balancer dengan LCU yang disediakan lebih besar dari 10 untuk memastikan bahwa mereka berada dalam 20% dari LCU yang disediakan tertinggi dalam kumpulan sumber daya.
- ElbV2ProvisionedCapacityEnabled: Memeriksa semua Network Load Balancer dan Application Load Balancer menyediakan status kapasitas untuk memastikan bahwa itu tidak memiliki nilai DISABLED atau PENDING.

Klaster Amazon MSK

- MskClusterClientSubnet: Memeriksa setiap cluster MSK untuk memastikan bahwa ia hanya memiliki dua atau hanya tiga subnet klien.
- MskClusterInstanceType: Memeriksa semua kluster MSK untuk memastikan bahwa mereka memiliki jenis instans Amazon EC2 yang sama.
- MskClusterSecurityGroupsMemeriksa semua kluster MSK untuk memastikan bahwa mereka memiliki kelompok keamanan yang sama.

- MskClusterStorageInfo: Memeriksa semua cluster MSK untuk memastikan bahwa mereka memiliki ukuran volume penyimpanan EBS yang sama. Jika satu memiliki nilai yang lebih besar, yang lain ditandai TIDAK SIAP.
- MskClusterACMCertificate: Memeriksa semua kluster MSK untuk memastikan bahwa mereka memiliki daftar sertifikat otorisasi klien yang sama. ARNs
- MskClusterServerProperties: Memeriksa semua kluster MSK untuk memastikan bahwa mereka memiliki nilai yang sama untuk. `Current Broker Software Info`
- MskClusterKafkaVersion: Memeriksa semua cluster MSK untuk memastikan bahwa mereka memiliki versi Kafka yang sama.
- MskClusterEncryptionInTransitInCluster: Memeriksa semua kluster MSK untuk memastikan bahwa mereka memiliki nilai yang sama untuk. `Encryption In Transit In Cluster`
- MskClusterEncryptionInClientBroker: Memeriksa semua kluster MSK untuk memastikan bahwa mereka memiliki nilai yang sama untuk. `Encryption In Transit Client Broker`
- MskClusterEnhancedMonitoring: Memeriksa semua kluster MSK untuk memastikan bahwa mereka memiliki nilai yang sama untuk. `Enhanced Monitoring`
- MskClusterOpenMonitoringInJmx: Memeriksa semua kluster MSK untuk memastikan bahwa mereka memiliki nilai yang sama untuk. `Open Monitoring JMX Exporter`
- MskClusterOpenMonitoringInNode: Memeriksa semua kluster MSK untuk memastikan bahwa mereka memiliki nilai yang sama `Open Monitoring Not Exporter.`
- MskClusterLoggingInS3: Memeriksa semua kluster MSK untuk memastikan bahwa mereka memiliki nilai yang sama untuk. `Is Logging in S3`
- MskClusterLoggingInFirehose: Memeriksa semua kluster MSK untuk memastikan bahwa mereka memiliki nilai yang sama untuk. `Is Logging In Firehose`
- MskClusterLoggingInCloudWatch: Memeriksa semua kluster MSK untuk memastikan bahwa mereka memiliki nilai yang sama untuk. `Is Logging Available In CloudWatch Logs`
- MskClusterNumberOfBrokerNodes: Memeriksa semua kluster MSK untuk memastikan mereka memiliki nilai yang sama. `Number of Broker Nodes` Jika satu memiliki nilai yang lebih besar, yang lain ditandai TIDAK SIAP.
- MskClusterState: Memeriksa setiap cluster MSK untuk memastikan bahwa itu dalam keadaan AKTIF.
- MskClusterLimitsRule: Memeriksa semua fungsi Lambda untuk memastikan bahwa mereka sesuai dengan kuota (batas) yang dikelola oleh Service Quotas.

Pemeriksaan kesehatan Amazon Route 53

- R53HealthCheckType: Memeriksa setiap pemeriksaan kesehatan Route 53 untuk memastikan bahwa itu bukan tipe DIHITUNG dan bahwa semua pemeriksaan memiliki jenis yang sama.
- R53HealthCheckDisabled: Memeriksa setiap pemeriksaan kesehatan Route 53 untuk memastikan bahwa itu tidak memiliki status DISABLED.
- R53HealthCheckStatus: Memeriksa setiap pemeriksaan kesehatan Route 53 untuk memastikan bahwa ia memiliki status SUKSES.
- R53HealthCheckRequestIntervalMemeriksa semua pemeriksaan kesehatan Route 53 untuk memastikan bahwa mereka semua memiliki nilai yang sama. `Request Interval`
- R53HealthCheckFailureThreshold: Memeriksa semua pemeriksaan kesehatan Route 53 untuk memastikan bahwa mereka semua memiliki nilai yang sama `Failure Threshold.`
- R53HealthCheckEnableSNI: Memeriksa semua pemeriksaan kesehatan Route 53 untuk memastikan bahwa mereka semua memiliki nilai yang sama `Enable SNI.`
- R53HealthCheckSearchString: Memeriksa semua pemeriksaan kesehatan Route 53 untuk memastikan bahwa mereka semua memiliki nilai yang sama `Search String.`
- R53HealthCheckRegions: Memeriksa semua pemeriksaan kesehatan Route 53 untuk memastikan bahwa mereka semua memiliki daftar AWS Wilayah yang sama.
- R53HealthCheckMeasureLatencyMemeriksa semua pemeriksaan kesehatan Route 53 untuk memastikan bahwa mereka semua memiliki nilai yang sama. `Measure Latency`
- R53HealthCheckInsufficientDataHealthStatusMemeriksa semua pemeriksaan kesehatan Route 53 untuk memastikan bahwa mereka semua memiliki nilai yang sama. `Insufficient Data Health Status`
- R53HealthCheckInverted: Memeriksa semua pemeriksaan kesehatan Route 53 untuk memastikan bahwa semuanya Terbalik, atau semuanya tidak Terbalik.
- R53HealthCheckResourcePathMemeriksa semua pemeriksaan kesehatan Route 53 untuk memastikan bahwa mereka semua memiliki nilai yang sama. `Resource Path`
- R53HealthCheckCloudWatchAlarm: Memeriksa semua pemeriksaan kesehatan Route 53 untuk memastikan bahwa CloudWatch alarm yang terkait dengannya memiliki pengaturan dan konfigurasi yang sama.

Berlangganan Amazon SNS

- SnsSubscriptionProtocol: Memeriksa semua langganan SNS untuk memastikan bahwa mereka memiliki protokol yang sama.

- SnsSubscriptionSqsLambdaEndpoint: Memeriksa semua langganan SNS yang memiliki titik akhir Lambda atau SQS untuk memastikan bahwa mereka memiliki titik akhir yang berbeda.
- SnsSubscriptionNonAwsEndpoint: Memeriksa semua langganan SNS yang memiliki tipe titik akhir AWS non-layanan, misalnya, email, untuk memastikan bahwa langganan memiliki titik akhir yang sama.
- SnsSubscriptionPendingConfirmation: Memeriksa semua langganan SNS untuk memastikan bahwa mereka memiliki nilai yang sama untuk 'Konfirmasi Tertunda'.
- SnsSubscriptionDeliveryPolicy: Memeriksa semua langganan SNS yang menggunakan HTTP/S untuk memastikan bahwa mereka memiliki nilai yang sama untuk 'Periode Pengiriman Efektif'.
- SnsSubscriptionRawMessageDelivery: Memeriksa semua langganan SNS untuk memastikan bahwa mereka memiliki nilai yang sama untuk 'Pengiriman Pesan Mentah'.
- SnsSubscriptionFilter: Memeriksa semua langganan SNS untuk memastikan bahwa mereka memiliki nilai yang sama untuk 'Kebijakan Filter'.
- SnsSubscriptionRedrivePolicy: Memeriksa semua langganan SNS untuk memastikan bahwa mereka memiliki nilai yang sama untuk 'Kebijakan Penggerak Ulang'.
- SnsSubscriptionEndpointEnabled: Memeriksa semua langganan SNS untuk memastikan bahwa mereka memiliki nilai yang sama untuk 'Titik Akhir Diaktifkan'.
- SnsSubscriptionLambdaEndpointValid: Memeriksa semua langganan SNS yang memiliki titik akhir Lambda untuk memastikan bahwa mereka memiliki titik akhir Lambda yang valid.
- SnsSubscriptionSqsEndpointValidRule: Memeriksa semua langganan SNS yang menggunakan titik akhir SQS untuk memastikan bahwa mereka memiliki titik akhir SQS yang valid.
- SnsSubscriptionQuotas: Memeriksa semua langganan SNS untuk memastikan bahwa mereka sesuai dengan kuota (batas) yang dikelola oleh Service Quotas.

Topik Amazon SNS

- SnsTopicDisplayName: Memeriksa semua topik SNS untuk memastikan bahwa mereka memiliki nilai yang sama untuk. `Display Name`
- SnsTopicDeliveryPolicy: Memeriksa semua topik SNS yang memiliki pelanggan HTTPS untuk memastikan bahwa mereka memiliki hal yang sama. `EffectiveDeliveryPolicy`
- SnsTopicSubscriptionMemeriksa semua topik SNS untuk memastikan bahwa mereka memiliki jumlah pelanggan yang sama untuk setiap protokol mereka.
- SnsTopicAwsKmsKey: Memeriksa semua topik SNS untuk memastikan bahwa semua topik atau tidak ada topik memiliki AWS KMS kunci.

- SnsTopicQuotas: Memeriksa semua topik SNS untuk memastikan bahwa mereka sesuai dengan kuota (batas) yang dikelola oleh Service Quotas.

Antrean Amazon SQS

- SqsQueueType: Memeriksa semua antrian SQS untuk memastikan bahwa semuanya memiliki nilai yang sama. `Type`
- SqsQueueDelaySeconds: Memeriksa semua antrian SQS untuk memastikan bahwa mereka semua memiliki nilai yang sama untuk. `Delay Seconds`
- SqsQueueMaximumMessageSize: Memeriksa semua antrian SQS untuk memastikan bahwa mereka semua memiliki nilai yang sama untuk. `Maximum Message Size`
- SqsQueueMessageRetentionPeriod: Memeriksa semua antrian SQS untuk memastikan bahwa mereka semua memiliki nilai yang sama untuk. `Message Retention Period`
- SqsQueueReceiveMessageWaitTimeSeconds: Memeriksa semua antrian SQS untuk memastikan bahwa mereka semua memiliki nilai yang sama untuk. `Receive Message Wait Time Seconds`
- SqsQueueRedrivePolicyMaxReceiveCount: Memeriksa semua antrian SQS untuk memastikan bahwa mereka semua memiliki nilai yang sama untuk. `Redrive Policy Max Receive Count`
- SqsQueueVisibilityTimeout: Memeriksa semua antrian SQS untuk memastikan bahwa mereka semua memiliki nilai yang sama untuk. `Visibility Timeout`
- SqsQueueContentBasedDeduplication: Memeriksa semua antrian SQS untuk memastikan bahwa mereka semua memiliki nilai yang sama untuk. `Content-Based Deduplication`
- SqsQueueQuotas: Memeriksa semua antrian SQS untuk memastikan bahwa mereka sesuai dengan kuota (limit) yang dikelola oleh Service Quotas.

Amazon VPCs

- VpcCidrBlock: Memeriksa semua VPCs untuk memastikan bahwa mereka semua memiliki nilai yang sama untuk ukuran jaringan blok CIDR.
- VpcCidrBlocksSameProtocolVersion: Memeriksa semua VPCs yang memiliki blok CIDR yang sama untuk memastikan bahwa mereka memiliki nilai yang sama untuk nomor versi Internet Stream Protocol.
- VpcCidrBlocksStateInAssociationSets: Memeriksa semua set asosiasi blok CIDR untuk semua VPCs untuk memastikan bahwa mereka semua memiliki blok CIDR yang berada dalam keadaan. `ASSOCIATED`

- VpcIpv6CidrBlocksStateInAssociationSets: Memeriksa semua kumpulan asosiasi blok CIDR untuk semua VPCs untuk memastikan bahwa mereka semua memiliki blok CIDR dengan jumlah alamat yang sama.
- VpcCidrBlocksInAssociationSets: Memeriksa semua set asosiasi blok CIDR untuk semua VPCs untuk memastikan bahwa mereka semua memiliki ukuran yang sama.
- VpcIpv6CidrBlocksInAssociationSets: Memeriksa semua set asosiasi blok IPv6 CIDR untuk semua VPCs untuk memastikan bahwa mereka memiliki ukuran yang sama.
- VpcState: Memeriksa setiap VPC untuk memastikan bahwa itu dalam `AVAILABLE` keadaan.
- VpcInstanceTenancy: Memeriksa semua VPCs untuk memastikan bahwa mereka semua memiliki nilai yang sama untuk`Instance Tenancy`.
- VpcIsDefault: Memeriksa semua VPCs untuk memastikan bahwa mereka memiliki nilai yang sama `Is Default.`
- VpcSubnetState: Memeriksa setiap subnet VPC untuk memastikan bahwa itu dalam keadaan TERSEDIA.
- VpcSubnetAvailableIpAddressCount: Memeriksa setiap subnet VPC untuk memastikan bahwa ia memiliki jumlah alamat IP yang tersedia lebih besar dari nol.
- VpcSubnetCount: Memeriksa semua subnet VPC untuk memastikan bahwa mereka memiliki jumlah subnet yang sama.
- VpcQuotas: Memeriksa semua subnet VPC untuk memastikan bahwa subnet tersebut sesuai dengan kuota (limit) yang dikelola oleh Service Quotas.

AWS VPN koneksi

- VpnConnectionsRouteCountMemeriksa semua koneksi VPN untuk memastikan bahwa mereka memiliki setidaknya satu rute, dan juga jumlah rute yang sama.
- VpnConnectionsEnableAcceleration: Memeriksa semua koneksi VPN untuk memastikan bahwa mereka memiliki nilai yang sama. `Enable Accelerations`
- VpnConnectionsStaticRoutesOnly: Memeriksa semua koneksi VPN untuk memastikan bahwa mereka memiliki nilai yang sama `Static Routes Only.`
- VpnConnectionsCategory: Memeriksa semua koneksi VPN untuk memastikan bahwa mereka memiliki kategori. `VPN`
- VpnConnectionsCustomerConfiguration: Memeriksa semua koneksi VPN untuk memastikan bahwa mereka memiliki nilai yang sama. `Customer Gateway Configuration`
- VpnConnectionsCustomerGatewayId: Memeriksa setiap koneksi VPN untuk memastikan bahwa ia memiliki gateway pelanggan yang terpasang.

- VpnConnectionsRoutesState: Memeriksa semua koneksi VPN untuk memastikan bahwa mereka berada dalam `AVAILABLE` keadaan.
- VpnConnectionsVgwTelemetryStatus: Memeriksa setiap koneksi VPN untuk memastikan bahwa ia memiliki status VGW. `UP`
- VpnConnectionsVgwTelemetryIpAddress: Memeriksa setiap koneksi VPN untuk memastikan bahwa ia memiliki alamat IP luar yang berbeda untuk setiap telemetri VGW.
- VpnConnectionsTunnelOptions: Memeriksa semua koneksi VPN untuk memastikan bahwa mereka memiliki opsi terowongan yang sama.
- VpnConnectionsRoutesCidr: Memeriksa semua koneksi VPN untuk memastikan bahwa mereka memiliki blok CIDR tujuan yang sama.
- VpnConnectionsInstanceType: Memeriksa semua koneksi VPN untuk memastikan bahwa mereka memiliki koneksi yang sama`Instance Type`.

AWS VPN gerbang

- VpnGatewayState: Memeriksa semua gateway VPN untuk memastikan bahwa mereka berada dalam keadaan TERSEDIA.
- VpnGatewayAsn: Memeriksa semua gateway VPN untuk memastikan bahwa mereka memiliki ASN yang sama.
- VpnGatewayType: Memeriksa semua gateway VPN untuk memastikan bahwa mereka memiliki tipe yang sama.
- VpnGatewayAttachment: Memeriksa semua gateway VPN untuk memastikan bahwa mereka memiliki konfigurasi lampiran yang sama.

Lihat aturan kesiapan di konsol

Anda dapat melihat aturan kesiapan pada AWS Management Console, terdaftar oleh setiap jenis sumber daya.

Untuk melihat aturan kesiapan di konsol

1. Buka konsol ARC dihttps://console.aws.amazon.com/route53recovery/home#/dashboard.
2. Pilih Pemeriksaan Kesiapan.
3. Di bawah Jenis sumber daya, pilih jenis sumber daya yang ingin Anda lihat aturannya.

Jenis sumber daya dan format ARN di ARC

Saat membuat kumpulan sumber daya di Amazon Application Recovery Controller (ARC), Anda menentukan jenis sumber daya yang akan disertakan dalam set dan Amazon Resource Names (ARNs) untuk setiap sumber daya yang akan disertakan. ARC mengharapkan format ARN tertentu untuk setiap jenis sumber daya. Bagian ini mencantumkan jenis sumber daya yang didukung oleh ARC dan format ARN terkait untuk masing-masing sumber daya.

Format spesifik tergantung pada sumber daya. Saat Anda memberikan ARN, ganti *`italicized`* teks dengan informasi khusus sumber daya Anda.

> **Note**
>
> Ketahuilah bahwa format ARN yang dibutuhkan ARC untuk sumber daya mungkin berbeda dari format ARN yang dibutuhkan layanan itu sendiri untuk sumber dayanya. Misalnya, format ARN yang dijelaskan di bagian Jenis sumber daya untuk setiap layanan dalam Referensi Otorisasi Layanan mungkin tidak menyertakan Akun AWS ID atau informasi lain yang dibutuhkan ARC untuk mendukung fitur dalam layanan ARC.

AWS::ApiGateway::Stage

Tahap Amazon API Gateway Versi 1.

- Format ARN: `arn:`*`partition`*`:apigateway:`*`region`*`:`*`account`*`:/restapis/`*`api-id`*`/stages/`*`stage-name`*

  Contoh: `arn:aws:apigateway:us-east-1:111122223333:/restapis/123456789/stages/ExampleStage`

  Untuk informasi selengkapnya, lihat referensi API Gateway Amazon Resource Name (ARN).

AWS::ApiGatewayV2::Stage

Tahap Amazon API Gateway Versi 2.

- Format ARN: `arn:`*`partition`*`:apigateway:`*`region`*`:`*`account`*`:/apis/`*`api-id`*`/stages/`*`stage-name`*

  Contoh: `arn:aws:apigateway:us-east-1:111122223333:/apis/123456789/stages/ExampleStage`

Untuk informasi selengkapnya, lihat referensi API Gateway Amazon Resource Name (ARN).

AWS::CloudWatch::Alarm

CloudWatch Alarm Amazon.

- Format ARN: `arn:partition:cloudwatch:region:account:alarm:alarm-name`

  Contoh: `arn:aws:cloudwatch:us-west-2:111122223333:alarm:test-alarm-1`

  Untuk informasi selengkapnya, lihat Jenis sumber daya yang ditentukan oleh Amazon CloudWatch.

AWS::DynamoDB::Table

Tabel Amazon DynamoDB.

- Format ARN: `arn:partition:dynamodb:region:account:table/table-name`

  Contoh: `arn:aws:dynamodb:us-west-2:111122223333:table/BigTable`

  Untuk informasi selengkapnya, lihat sumber daya dan operasi DynamoDB.

AWS::EC2::CustomerGateway

Perangkat gateway pelanggan.

- Format ARN: `arn:partition:ec2:region:account:customer-gateway/CustomerGatewayId`

  Contoh: `arn:aws:ec2:us-west-2:111122223333:customer-gateway/vcg-123456789`

  Untuk informasi selengkapnya, lihat Jenis sumber daya yang ditentukan oleh Amazon EC2.

AWS::EC2::Volume

Volume Amazon EBS.

- Format ARN: `arn:partition:ec2:region:account:volume/VolumeId`

  Contoh: `arn:aws:ec2:us-west-2:111122223333:volume/volume-of-cylinder-is-pi`

  Untuk informasi selengkapnya, lihat referensi API Gateway Amazon Resource Name (ARN).

AWS::ElasticLoadBalancing::LoadBalancer

Sebuah Classic Load Balancer.

- Format ARN: `arn:partition:elasticloadbalancing:region:account:loadbalancer/LoadBalancerN`

  Contoh: `arn:aws:elasticloadbalancing:us-west-2:111122223333:loadbalancer/123456789abcbdeCLB`

  Untuk informasi selengkapnya, lihat sumber daya Elastic Load Balancing.

AWS::ElasticLoadBalancingV2::LoadBalancer

Network Load Balancer atau Application Load Balancer.

- Format ARN untuk Network Load Balancer: `arn:partition:elasticloadbalancing:region:account:loadbalancer/net/LoadBalancerName`

  Contoh untuk Network Load Balancer: `arn:aws:elasticloadbalancing:us-west-2:111122223333:loadbalancer/net/sandbox-net/123456789acbdeNLB`
- Format ARN untuk Application Load Balancer: `arn:partition:elasticloadbalancing:region:account:loadbalancer/app/LoadBalancerName`

  Contoh untuk Application Load Balancer: `arn:aws:elasticloadbalancing:us-west-2:111122223333:loadbalancer/app/sandbox-alb/123456789acbdeALB`

  Untuk informasi selengkapnya, lihat sumber daya Elastic Load Balancing.

AWS::Lambda::Function

Sebuah AWS Lambda fungsi.

- Format ARN: `arn:partition:lambda:region:account:function:FunctionName`

  Contoh: `arn:aws:lambda:us-west-2:111122223333:function:my-function`

  Untuk informasi selengkapnya, lihat Sumber daya dan ketentuan untuk tindakan Lambda.

AWS::MSK::Cluster

Cluster MSK Amazon.

- Format ARN: `arn:partition:kafka:region:account:cluster/ClusterName/UUID`

  Contoh: `arn:aws:kafka:us-east-1:111122223333:cluster/demo-cluster-1/123456-1111-2222-3333`

  Untuk informasi selengkapnya, lihat Jenis sumber daya yang ditentukan oleh Amazon Managed Streaming for Apache Kafka.

AWS::RDS::DBCluster

Cluster Aurora DB.

- Format ARN: `arn:partition:rds:region:account:cluster:DbClusterInstanceName`

  Contoh: `arn:aws:rds:us-west-2:111122223333:cluster:database-1`

  Untuk informasi selengkapnya, lihat Bekerja dengan Nama Sumber Daya Amazon (ARNs) di Amazon RDS.

AWS::Route53::HealthCheck

Pemeriksaan kesehatan Amazon Route 53.

- Format ARN: `arn:partition:route53:::healthcheck/Id`

  Contoh: `arn:aws:route53:::healthcheck/123456-1111-2222-3333`

AWS::SQS::Queue

Antrian Amazon SQS.

- Format ARN: `arn:partition:sqs:region:account:QueueName`

  Contoh: `arn:aws:sqs:us-west-2:111122223333:StandardQueue`

  Untuk informasi selengkapnya, lihat sumber daya dan operasi Amazon Simple Queue Service.

AWS::SNS::Topic

Topik Amazon SNS.

- Format ARN: `arn:partition:sns:region:account:TopicName`

  Contoh: `arn:aws:sns:us-west-2:111122223333:TopicName`

  Untuk informasi selengkapnya, lihat format ARN sumber daya Amazon SNS.

AWS::SNS::Subscription

Berlangganan Amazon SNS.

- Format ARN: `arn:partition:sns:region:account:TopicName:SubscriptionId`

  Contoh: `arn:aws:sns:us-west-2:111122223333:TopicName:12345678901234567890`

AWS::EC2::VPC

Virtual Private Cloud (VPC)

- Format ARN: `arn:partition:ec2:region:account:vpc/VpcId`

  Contoh: `arn:aws:ec2:us-west-2:111122223333:vpc/vpc-123456789`

  Untuk informasi selengkapnya, lihat Sumber Daya VPC.

AWS::EC2::VPNConnection

Koneksi jaringan pribadi virtual (VPN).

- Format ARN: `arn:partition:ec2:region:account:vpn-connection/VpnConnectionId`

  Contoh: `arn:aws:ec2:us-west-2:111122223333:vpn-connection/vpn-123456789`

  Untuk informasi selengkapnya, lihat Jenis sumber daya yang ditentukan oleh Amazon EC2.

AWS::EC2::VPNGateway

Gateway jaringan pribadi virtual (VPN).

- Format ARN: `arn:partition:ec2:region:account:vpn-gateway/VpnGatewayId`

  Contoh: `arn:aws:ec2:us-west-2:111122223333:vpn-gateway/vgw-123456789acbdefgh`

  Untuk informasi selengkapnya, lihat Jenis sumber daya yang ditentukan oleh Amazon EC2.

AWS::Route53RecoveryReadiness::DNSTargetResource

Sumber daya target DNS untuk pemeriksaan kesiapan mencakup jenis catatan DNS, nama domain, ARN zona yang dihosting Route 53, dan Network Load Balancer ARN atau Route 53 record set ID.

- Format ARN untuk zona yang dihosting: `arn:partition:route53::account:hostedzone/Id`

  Contoh untuk zona yang dihosting: `arn:aws:route53::111122223333:hostedzone/abcHostedZone`

  CATATAN: Anda harus menyertakan ID akun di zona yang dihosting ARNs, seperti yang ditentukan di sini. ID akun diperlukan agar ARC dapat melakukan polling sumber daya. Format ini sengaja berbeda dari format ARN yang diperlukan Amazon Route 53, dijelaskan dalam jenis Sumber Daya layanan Route 53 dalam Referensi Otorisasi Layanan.
- Format ARN untuk Network Load Balancer: `arn:partition:elasticloadbalancing:region:account:loadbalancer/net/LoadBalancerName`

  Contoh untuk Network Load Balancer: `arn:aws:elasticloadbalancing:us-west-2:111122223333:loadbalancer/net/sandbox-net/123456789acbdefgh`

  Untuk informasi selengkapnya, lihat sumber daya Elastic Load Balancing.

Pencatatan dan pemantauan untuk pemeriksaan kesiapan di Amazon Application Recovery Controller (ARC)

Anda dapat menggunakan Amazon CloudWatch, AWS CloudTrail, dan Amazon EventBridge untuk memantau pemeriksaan kesiapan di Amazon Application Recovery Controller (ARC), untuk menganalisis pola dan membantu memecahkan masalah.

> **Note**
>
> Anda harus melihat CloudWatch metrik dan log untuk ARC di Wilayah AS Barat (Oregon), baik di konsol maupun saat menggunakan. AWS CLI Saat Anda menggunakan AWS CLI, tentukan Wilayah Barat AS (Oregon) untuk perintah Anda dengan menyertakan parameter berikut:`--region us-west-2`.

Topik

- Menggunakan Amazon CloudWatch dengan pemeriksaan kesiapan di ARC
- Kesiapan pencatatan memeriksa panggilan API menggunakan AWS CloudTrail

- Menggunakan pemeriksaan kesiapan di ARC dengan Amazon EventBridge

Menggunakan Amazon CloudWatch dengan pemeriksaan kesiapan di ARC

Amazon Application Recovery Controller (ARC) menerbitkan titik data ke Amazon CloudWatch untuk pemeriksaan kesiapan Anda. CloudWatch memungkinkan Anda untuk mengambil statistik tentang titik-titik data tersebut sebagai kumpulan data deret waktu yang diurutkan, yang dikenal sebagai metrik. Anggap metrik sebagai variabel untuk memantau, dan titik data sebagai nilai variabel tersebut dari waktu ke waktu. Misalnya, Anda dapat memantau lalu lintas melalui AWS Wilayah selama periode waktu tertentu. Setiap titik data memiliki stempel waktu terkait dan unit pengukuran opsional.

Anda dapat menggunakan metrik untuk memverifikasi bahwa sistem Anda bekerja sesuai harapan. Misalnya, Anda dapat membuat CloudWatch alarm untuk memantau metrik tertentu dan memulai tindakan (seperti mengirim pemberitahuan ke alamat email) jika metrik berada di luar rentang yang Anda anggap dapat diterima.

Untuk informasi selengkapnya, lihat Panduan CloudWatch Pengguna Amazon.

Topik

- Metrik ARC
- Statistik untuk metrik ARC
- Lihat CloudWatch metrik di ARC

Metrik ARC

Namespace `AWS/Route53RecoveryReadiness` mencakup metrik berikut.

Metrik	Deskripsi
`ReadinessChecks`	Merupakan jumlah pemeriksaan kesiapan yang diproses oleh ARC. Metrik dapat dimensinya berdasarkan statusnya, tercantum di bawah ini. Satuan:`Count`. Kriteria pelaporan: Ada nilai bukan nol. Statistik: Satu-satunya statistik yang berguna adalah`Sum`.

Metrik	Deskripsi
	Dimensi • READY • NOT_READY • NOT_AUTHORIZED • UNKNOWN
Resources	Merupakan jumlah sumber daya yang diproses oleh ARC, yang dapat didimensikan oleh pengenal sumber dayanya, seperti yang didefinis ikan oleh API. Satuan:Count. Kriteria pelaporan: Ada nilai bukan nol. Statistik: Satu-satunya statistik yang berguna adalahSum. Dimensi • ResourceSetType : Ini adalah jenis sumber daya, disaring berdasarkan jumlah sumber daya per jenis tertentu yang dievaluas i oleh ARC Misalnya: AWS::CloudWatch::Alarm

Statistik untuk metrik ARC

CloudWatch menyediakan statistik berdasarkan titik data metrik yang diterbitkan oleh ARC. Statistik adalah agregasi data metrik selama periode waktu tertentu. Bila Anda meminta statistik, aliran data yang dikembalikan akan diidentifikasi dengan nama metrik dan dimensi. Dimensi adalah pasangan nama/nilai yang merupakan bagian dari identitas metrik.

Berikut ini adalah contoh kombinasi metrik/dimensi yang mungkin berguna bagi Anda:

- Lihat jumlah pemeriksaan kesiapan yang dievaluasi untuk kesiapan oleh ARC.
- Lihat jumlah total sumber daya untuk jenis kumpulan sumber daya tertentu yang dievaluasi oleh ARC.

Lihat CloudWatch metrik di ARC

Anda dapat melihat CloudWatch metrik untuk ARC menggunakan CloudWatch konsol atau. AWS CLI Di konsol, metrik ditampilkan sebagai grafik pemantauan.

Anda harus melihat CloudWatch metrik untuk ARC di Wilayah AS Barat (Oregon), baik di konsol maupun saat menggunakan. AWS CLI Saat Anda menggunakan AWS CLI, tentukan Wilayah Barat AS (Oregon) untuk perintah Anda dengan menyertakan parameter berikut:`--region us-west-2`.

Untuk melihat metrik menggunakan konsol CloudWatch

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Di panel navigasi, pilih Metrik.
3. Pilih namespace Route53 RecoveryReadiness.
4. (Opsional) Untuk melihat metrik di semua dimensi, ketik namanya di kolom pencarian.

Untuk melihat metrik menggunakan AWS CLI

Gunakan perintah list-metrics berikut untuk mencantumkan metrik yang tersedia:

```
aws cloudwatch list-metrics --namespace AWS/Route53RecoveryReadiness --region us-west-2
```

Untuk mendapatkan statistik untuk metrik menggunakan AWS CLI

Gunakan get-metric-statisticsperintah berikut untuk mendapatkan statistik untuk metrik dan dimensi tertentu. Perhatikan bahwa CloudWatch memperlakukan setiap kombinasi dimensi yang unik sebagai metrik terpisah. Anda tidak dapat mengambil statistik menggunakan kombinasi dimensi yang tidak dipublikasikan secara khusus. Anda harus menentukan dimensi yang sama yang digunakan saat metrik dibuat.

Contoh berikut mencantumkan total pemeriksaan kesiapan yang dievaluasi, per menit, untuk akun di ARC.

```
aws cloudwatch get-metric-statistics --namespace AWS/Route53RecoveryReadiness \
--metric-name ReadinessChecks \
--region us-west-2 \
--statistics Sum --period 60 \
--dimensions Name=State,Value=READY \
```

```
--start-time 2021-07-03T01:00:00Z --end-time 2021-07-03T01:20:00Z
```

Berikut ini adalah contoh output dari perintah:

```
{
    "Label": "ReadinessChecks",
    "Datapoints": [
        {
            "Timestamp": "2021-07-08T18:00:00Z",
            "Sum": 1.0,
            "Unit": "Count"
        },
        {
            "Timestamp": "2021-07-08T18:04:00Z",
            "Sum": 1.0,
            "Unit": "Count"
        },
        {
            "Timestamp": "2021-07-08T18:01:00Z",
            "Sum": 1.0,
            "Unit": "Count"
        },
        {
            "Timestamp": "2021-07-08T18:02:00Z",
            "Sum": 1.0,
            "Unit": "Count"
        },
        {
            "Timestamp": "2021-07-08T18:03:00Z",
            "Sum": 1.0,
            "Unit": "Count"
        }
    ]
}
```

Kesiapan pencatatan memeriksa panggilan API menggunakan AWS CloudTrail

terintegrasi dengan AWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan di ARC. CloudTrail menangkap semua panggilan API untuk ARC sebagai peristiwa. Panggilan yang diambil termasuk panggilan dari konsol ARC dan panggilan kode ke operasi ARC API.

Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail acara secara terus menerus ke bucket Amazon S3, termasuk acara untuk ARC. Jika Anda tidak mengonfigurasi jejak, Anda masih dapat melihat peristiwa terbaru di CloudTrail konsol dalam Riwayat acara.

Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat untuk ARC, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan.

Untuk mempelajari selengkapnya CloudTrail, lihat Panduan AWS CloudTrail Pengguna.

Informasi ARC di CloudTrail

CloudTrail diaktifkan pada Akun AWS saat Anda membuat akun. Ketika aktivitas terjadi di ARC, aktivitas tersebut dicatat dalam suatu CloudTrail peristiwa bersama dengan peristiwa AWS layanan lainnya dalam riwayat Acara. Anda dapat melihat, mencari, dan mengunduh acara terbaru di situs Anda Akun AWS. Untuk informasi selengkapnya, lihat Bekerja dengan riwayat CloudTrail Acara.

Untuk catatan acara yang sedang berlangsung di Anda Akun AWS, termasuk acara untuk ARC, buat jejak. Jejak memungkinkan CloudTrail untuk mengirimkan file log ke bucket Amazon S3. Secara default, saat Anda membuat jejak di konsol, jejak tersebut berlaku untuk semua Wilayah AWS. Jejak mencatat peristiwa dari semua Wilayah di AWS partisi dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat mengonfigurasi AWS layanan lain untuk menganalisis lebih lanjut dan menindaklanjuti data peristiwa yang dikumpulkan dalam CloudTrail log. Untuk informasi selengkapnya, lihat berikut:

- Gambaran umum untuk membuat jejak
- CloudTrail layanan dan integrasi yang didukung
- Mengonfigurasi notifikasi Amazon SNS untuk CloudTrail
- Menerima file CloudTrail log dari beberapa wilayah dan Menerima file CloudTrail log dari beberapa akun

Semua tindakan ARC dicatat oleh CloudTrail dan didokumentasikan dalam Panduan Referensi API Kesiapan Pemulihan untuk Pengontrol Pemulihan Aplikasi Amazon, Panduan Referensi API Konfigurasi Kontrol Pemulihan untuk Pengontrol Pemulihan Aplikasi Amazon, dan Panduan Referensi API Kontrol Perutean untuk Pengontrol Pemulihan Aplikasi Amazon. Misalnya, panggilan ke`CreateCluster`, `UpdateRoutingControlState` dan `CreateRecoveryGroup` tindakan menghasilkan entri dalam file CloudTrail log.

Setiap entri peristiwa atau log berisi informasi tentang siapa yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan berikut ini:

- Apakah permintaan itu dibuat dengan kredenal pengguna root atau AWS Identity and Access Management (IAM).
- Apakah permintaan tersebut dibuat dengan kredensial keamanan sementara untuk satu peran atau pengguna gabungan.
- Apakah permintaan itu dibuat oleh AWS layanan lain.

Untuk informasi selengkapnya, lihat Elemen userIdentity CloudTrail .

Melihat peristiwa ARC dalam sejarah acara

CloudTrail memungkinkan Anda melihat peristiwa terbaru dalam riwayat Acara. Untuk melihat peristiwa permintaan ARC API, Anda harus memilih US West (Oregon) di pemilih Wilayah di bagian atas konsol. Untuk informasi selengkapnya, lihat Bekerja dengan riwayat CloudTrail Acara di Panduan AWS CloudTrail Pengguna.

Memahami entri file log ARC

Trail adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai file log ke bucket Amazon S3 yang Anda tentukan. CloudTrail file log berisi satu atau lebih entri log. Peristiwa mewakili permintaan tunggal dari sumber manapun dan mencakup informasi tentang tindakan yang diminta, tanggal dan waktu tindakan, parameter permintaan, dan sebagainya. CloudTrail file log bukanlah jejak tumpukan yang diurutkan dari panggilan API publik, jadi file tersebut tidak muncul dalam urutan tertentu.

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan `CreateRecoveryGroup` tindakan untuk pemeriksaan kesiapan.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "A1B2C3D4E5F6G7EXAMPLE",
        "arn": "arn:aws:iam::111122223333:role/admin",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
```

```
                "type": "Role",
                "principalId": "AROA33L3W36EXAMPLE",
                "arn": "arn:aws:iam::111122223333:role/admin",
                "accountId": "111122223333",
                "userName": "EXAMPLENAME"
            },
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2021-07-06T17:38:05Z"
            }
        }
    },
    "eventTime": "2021-07-06T18:08:03Z",
    "eventSource": "route53-recovery-readiness.amazonaws.com",
    "eventName": "CreateRecoveryGroup",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.50",
    "userAgent": "Boto3/1.17.101 Python/3.8.10 Linux/4.14.231-180.360.amzn2.x86_64
 exec-env/AWS_Lambda_python3.8 Botocore/1.20.102",
    "requestParameters": {
        "recoveryGroupName": "MyRecoveryGroup"
    },
    "responseElements": {
        "Access-Control-Expose-Headers": "x-amzn-errortype,x-amzn-requestid,x-amzn-
errormessage,x-amzn-trace-id,x-amzn-requestid,x-amz-apigw-id,date",
        "cells": [],
        "recoveryGroupName": "MyRecoveryGroup",
        "recoveryGroupArn": "arn:aws:route53-recovery-readiness::111122223333:recovery-
group/MyRecoveryGroup",
        "tags": "***"
    },
    "requestID": "fd42dcf7-6446-41e9-b408-d096example",
    "eventID": "4b5c42df-1174-46c8-be99-d67aexample",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "111122223333"
}
```

Menggunakan pemeriksaan kesiapan di ARC dengan Amazon EventBridge

Menggunakan Amazon EventBridge, Anda dapat mengatur aturan berbasis peristiwa yang memantau sumber daya pemeriksaan kesiapan Anda di Amazon Application Recovery Controller (ARC), dan kemudian memulai tindakan target yang menggunakan layanan lain. AWS Misalnya, Anda dapat menetapkan aturan untuk mengirimkan notifikasi email dengan memberi sinyal topik Amazon SNS saat status pemeriksaan kesiapan berubah dari SIAP menjadi TIDAK SIAP.

> **Note**
>
> ARC hanya menerbitkan EventBridge acara untuk pemeriksaan kesiapan di Wilayah AS Barat (Oregon) (us-west-2). AWS Untuk menerima EventBridge acara untuk pemeriksaan kesiapan, buat EventBridge aturan di Wilayah Barat AS (Oregon).

Anda dapat membuat aturan di Amazon EventBridge untuk menindaklanjuti peristiwa pemeriksaan kesiapan ARC berikut:

- Kesiapan memeriksa kesiapan. Acara menentukan apakah kesiapan memeriksa status berubah, misalnya, dari SIAP ke TIDAK SIAP.

Untuk menangkap peristiwa ARC tertentu yang Anda minati, tentukan pola khusus peristiwa yang EventBridge dapat digunakan untuk mendeteksi peristiwa. Pola acara memiliki struktur yang sama dengan peristiwa yang cocok. Pola mengutip bidang yang ingin Anda cocokkan dan memberikan nilai yang Anda cari.

Peristiwa dipancarkan atas dasar upaya terbaik. Mereka dikirim dari ARC ke hampir real-time EventBridge dalam keadaan operasional normal. Namun, situasi dapat muncul yang mungkin menunda atau mencegah pengiriman suatu peristiwa.

Untuk informasi tentang cara kerja EventBridge aturan dengan pola peristiwa, lihat Peristiwa dan Pola Peristiwa di EventBridge.

Pantau sumber daya pemeriksaan kesiapan dengan EventBridge

Dengan EventBridge, Anda dapat membuat aturan yang menentukan tindakan yang harus diambil saat ARC memancarkan peristiwa untuk sumber daya pemeriksaan kesiapan.

Untuk mengetik atau menyalin dan menempelkan pola acara ke EventBridge konsol, di konsol, pilih opsi Masukkan opsi saya sendiri. Untuk membantu Anda menentukan pola acara yang mungkin berguna bagi Anda, topik ini mencakup contoh pola acara kesiapan.

Untuk membuat aturan untuk peristiwa sumber daya

1. Buka EventBridge konsol Amazon di https://console.aws.amazon.com/events/.
2. Wilayah AWS Untuk membuat aturan di, pilih US West (Oregon). Ini adalah Wilayah yang diperlukan untuk acara kesiapan.
3. Pilih Buat aturan.
4. Masukkan Nama untuk aturan tersebut, dan, secara opsional, deskripsi.
5. Untuk bus Acara, biarkan nilai default, default.
6. Pilih Berikutnya.
7. Untuk langkah pola acara Build, untuk sumber Event, tinggalkan nilai default, AWS peristiwa.
8. Di bawah Contoh acara, pilih Masukkan milik saya.
9. Untuk contoh peristiwa, ketik atau salin dan tempel pola acara. Sebagai contoh, lihat bagian selanjutnya.

Contoh pola acara kesiapan

Pola acara memiliki struktur yang sama dengan peristiwa yang cocok. Pola mengutip bidang yang ingin Anda cocokkan dan memberikan nilai yang Anda cari.

Anda dapat menyalin dan menempelkan pola peristiwa dari bagian ini EventBridge ke dalam untuk membuat aturan yang dapat Anda gunakan untuk memantau tindakan dan sumber daya ARC.

Pola peristiwa berikut memberikan contoh yang mungkin Anda gunakan EventBridge untuk kemampuan pemeriksaan kesiapan di ARC.

- Pilih semua acara dari pemeriksaan kesiapan ARC.

```
{
    "source": [
        "aws.route53-recovery-readiness"
    ]
}
```

- Pilih hanya acara yang terkait dengan sel.

```
{
    "source": [
        "aws.route53-recovery-readiness"
    ],
    "detail-type": [
        "Route 53 Application Recovery Controller cell readiness status change"
    ]
}
```

- Pilih hanya peristiwa yang terkait dengan sel tertentu yang disebut *`MyExampleCell`*.

```
{
    "source": [
        "aws.route53-recovery-readiness"
    ],
    "detail-type": [
        "Route 53 Application Recovery Controller cell readiness status change"
    ],
    "resources": [
        "arn:aws:route53-recovery-readiness::111122223333:cell/MyExampleCell"
    ]
}
```

- Pilih hanya acara saat grup pemulihan, sel, atau status pemeriksaan kesiapan menjadi *`NOT READY`*.

```
{
   "source":[
      "aws.route53-recovery-readiness"
   ],
   "detail-type":{
      "new-state":{
         "readiness-status":[
            "NOT_READY"
         ]
      }
   }
}
```

- Pilih hanya peristiwa ketika grup pemulihan, sel, atau pemeriksaan kesiapan menjadi apa pun kecuali *`READY`*

```
{
   "source":[
      "aws.route53-recovery-readiness"
   ],
   "detail":{
      "new-state":{
         "readiness-status":[
            {
               "anything-but":"READY"
            }
         ]
      }
   }
}
```

Berikut ini adalah contoh peristiwa ARC untuk perubahan status kesiapan grup pemulihan:

```
{
    "version": "0",
    "account":"111122223333",
    "detail-type":"Route 53 Application Recovery Controller recovery group readiness
 status change",
    "source":"route53-recovery-readiness.amazonaws.com",
    "time":"2020-11-03T00:31:54Z",
    "id": "1234a678-1b23-c123-12fd3f456e78",
    "region": "us-west-2",
    "resources":[
        "arn:aws:route53-recovery-readiness::111122223333:recovery-group/BillingApp"
    ],
    "detail": {
        "recovery-group-name": "BillingApp",
        "previous-state": {
            "readiness-status": "READY|NOT_READY|UNKNOWN|NOT_AUTHORIZED"
        },
        "new-state": {
            "readiness-status": "READY|NOT_READY|UNKNOWN|NOT_AUTHORIZED"
        }
    }
}
```

Berikut ini adalah contoh peristiwa ARC untuk perubahan status kesiapan sel:

```
{
    "version": "0",
    "account":"111122223333",
    "detail-type":"Route 53 Application Recovery Controller cell readiness status
 change",
    "source":"route53-recovery-readiness.amazonaws.com",
    "time":"2020-11-03T00:31:54Z",
    "id": "1234a678-1b23-c123-12fd3f456e78",
    "region": "us-west-2",
    "resources":[
        "arn:aws:route53-recovery-readiness::111122223333:cell/PDXCell"
    ],
    "detail": {
        "cell-name": "PDXCell",
        "previous-state": {
            "readiness-status": "READY|NOT_READY|UNKNOWN|NOT_AUTHORIZED"
        },
        "new-state": {
            "readiness-status": "READY|NOT_READY|UNKNOWN|NOT_AUTHORIZED"
        }
    }
}
```

Berikut ini adalah contoh peristiwa ARC untuk perubahan status pemeriksaan kesiapan:

```
{
    "version": "0",
    "account":"111122223333",
    "detail-type":"Route 53 Application Recovery Controller readiness check status
 change",
    "source":"route53-recovery-readiness.amazonaws.com",
    "time":"2020-11-03T00:31:54Z",
    "id": "1234a678-1b23-c123-12fd3f456e78",
    "region": "us-west-2",
    "resources":[
        "arn:aws:route53-recovery-readiness::111122223333:readiness-check/
UserTableReadinessCheck"
    ],
    "detail": {
    "readiness-check-name": "UserTableReadinessCheck",
        "previous-state": {
            "readiness-status": "READY|NOT_READY|UNKNOWN|NOT_AUTHORIZED"
        },
```

```
        "new-state": {
            "readiness-status": "READY|NOT_READY|UNKNOWN|NOT_AUTHORIZED"
        }
    }
}
```

Tentukan grup CloudWatch log yang akan digunakan sebagai target

Saat membuat EventBridge aturan, Anda harus menentukan target tempat peristiwa yang cocok dengan aturan dikirim. Untuk daftar target yang tersedia EventBridge, lihat Target yang tersedia di EventBridge konsol. Salah satu target yang dapat Anda tambahkan ke EventBridge aturan adalah grup CloudWatch log Amazon. Bagian ini menjelaskan persyaratan untuk menambahkan grup CloudWatch log sebagai target, dan menyediakan prosedur untuk menambahkan grup log saat Anda membuat aturan.

Untuk menambahkan grup CloudWatch log sebagai target, Anda dapat melakukan salah satu hal berikut:

- Buat grup log baru
- Pilih grup log yang ada

Jika Anda menentukan grup log baru menggunakan konsol saat membuat aturan, EventBridge secara otomatis membuat grup log untuk Anda. Pastikan grup log yang Anda gunakan sebagai target EventBridge aturan dimulai dengan`/aws/events`. Jika Anda ingin memilih grup log yang ada, ketahuilah bahwa hanya grup log yang dimulai dengan yang `/aws/events` muncul sebagai opsi di menu tarik-turun. Untuk informasi selengkapnya, lihat Membuat grup log baru di Panduan CloudWatch Pengguna Amazon.

Jika Anda membuat atau menggunakan grup CloudWatch log untuk digunakan sebagai target menggunakan CloudWatch operasi di luar konsol, pastikan Anda menetapkan izin dengan benar. Jika Anda menggunakan konsol untuk menambahkan grup log ke EventBridge aturan, maka kebijakan berbasis sumber daya untuk grup log diperbarui secara otomatis. Namun, jika Anda menggunakan AWS Command Line Interface atau AWS SDK untuk menentukan grup log, Anda harus memperbarui kebijakan berbasis sumber daya untuk grup log. Contoh kebijakan berikut menggambarkan izin yang harus Anda tentukan dalam kebijakan berbasis sumber daya untuk grup log:

```
{
```

```
  "Statement": [
    {
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "events.amazonaws.com",
          "delivery.logs.amazonaws.com"
        ]
      },
      "Resource": "arn:aws:logs:region:account:log-group:/aws/events/*:*",
      "Sid": "TrustEventsToStoreLogEvent"
    }
  ],
  "Version": "2012-10-17"
}
```

Anda tidak dapat mengonfigurasi kebijakan berbasis sumber daya untuk grup log menggunakan konsol. Untuk menambahkan izin yang diperlukan ke kebijakan berbasis sumber daya, gunakan operasi API. CloudWatch PutResourcePolicy Kemudian, Anda dapat menggunakan perintah describe-resource-policiesCLI untuk memeriksa apakah kebijakan Anda diterapkan dengan benar.

Untuk membuat aturan untuk acara sumber daya dan menentukan target grup CloudWatch log

1. Buka EventBridge konsol Amazon di https://console.aws.amazon.com/events/.
2. Pilih aturan Wilayah AWS yang ingin Anda buat.
3. Pilih Buat aturan lalu masukkan informasi apa pun tentang aturan itu, seperti pola acara atau detail jadwal.

   Untuk informasi selengkapnya tentang membuat EventBridge aturan untuk kesiapan, lihat Memantau sumber daya pemeriksaan kesiapan dengan. EventBridge
4. Pada halaman Pilih target, pilih CloudWatchsebagai target Anda.
5. Pilih grup CloudWatch log dari menu tarik-turun.

Identity and Access Management untuk pemeriksaan kesiapan di ARC

AWS Identity and Access Management (IAM) adalah Layanan AWS yang membantu administrator mengontrol akses ke AWS sumber daya dengan aman. Administrator IAM mengontrol siapa yang dapat diautentikasi (masuk) dan diberi wewenang (memiliki izin) untuk menggunakan sumber daya ARC. IAM adalah Layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

Daftar Isi

Bagaimana pemeriksaan kesiapan di Amazon Application Recovery Controller (ARC) bekerja dengan IAM

Sebelum Anda menggunakan IAM untuk mengelola akses ke ARC, pelajari fitur IAM apa yang tersedia untuk digunakan dengan ARC.

Sebelum Anda menggunakan IAM untuk mengelola akses ke pemeriksaan kesiapan di Amazon Application Recovery Controller (ARC), pelajari fitur IAM apa yang tersedia untuk digunakan dengan pemeriksaan kesiapan.

Fitur IAM yang dapat Anda gunakan dengan pemeriksaan kesiapan di Amazon Application Recovery Controller (ARC)

Fitur IAM	Dukungan pemeriksaan kesiapan
Kebijakan berbasis identitas	Ya
Kebijakan berbasis sumber daya	Tidak
Tindakan kebijakan	Ya
Sumber daya kebijakan	Ya
Kunci kondisi kebijakan	Ya

Fitur IAM	Dukungan pemeriksaan kesiapan
ACLs	Tidak
ABAC (tanda dalam kebijakan)	Ya
Kredensial sementara	Ya
Izin principal	Ya
Peran layanan	Tidak
Peran terkait layanan	Ya

Untuk mendapatkan tampilan tingkat tinggi dan keseluruhan tentang cara kerja AWS layanan dengan sebagian besar fitur IAM, lihat AWS layanan yang bekerja dengan IAM di Panduan Pengguna IAM.

Kebijakan berbasis identitas untuk pemeriksaan kesiapan

Mendukung kebijakan berbasis identitas: Ya

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, seperti pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana, dan berdasarkan kondisi seperti apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Tentukan izin IAM kustom dengan kebijakan terkelola pelanggan dalam Panduan Pengguna IAM.

Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan secara spesifik apakah tindakan dan sumber daya diizinkan atau ditolak, serta kondisi yang menjadi dasar dikabulkan atau ditolaknya tindakan tersebut. Anda tidak dapat menentukan secara spesifik prinsipal dalam sebuah kebijakan berbasis identitas karena prinsipal berlaku bagi pengguna atau peran yang melekat kepadanya. Untuk mempelajari semua elemen yang dapat Anda gunakan dalam kebijakan JSON, lihat Referensi elemen kebijakan JSON IAM dalam Panduan Pengguna IAM.

Untuk melihat contoh kebijakan berbasis identitas ARC, lihat. Contoh kebijakan berbasis identitas di Amazon Application Recovery Controller (ARC)

Kebijakan berbasis sumber daya dalam pemeriksaan kesiapan

Mendukung kebijakan berbasis sumber daya: Tidak

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu.

Tindakan kebijakan untuk pemeriksaan kesiapan

Mendukung tindakan kebijakan: Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen `Action` dari kebijakan JSON menjelaskan tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam sebuah kebijakan. Tindakan kebijakan biasanya memiliki nama yang sama dengan operasi AWS API terkait. Ada beberapa pengecualian, misalnya tindakan hanya izin yang tidak memiliki operasi API yang cocok. Ada juga beberapa operasi yang memerlukan beberapa tindakan dalam suatu kebijakan. Tindakan tambahan ini disebut tindakan dependen.

Sertakan tindakan dalam kebijakan untuk memberikan izin untuk melakukan operasi terkait.

Untuk melihat daftar tindakan ARC untuk pemeriksaan kesiapan, lihat Tindakan yang ditentukan oleh Kesiapan Pemulihan Amazon Route 53 di Referensi Otorisasi Layanan.

Tindakan kebijakan di ARC untuk pemeriksaan kesiapan menggunakan awalan berikut sebelum tindakan:

```
route53-recovery-readiness
```

Untuk menetapkan secara spesifik beberapa tindakan dalam satu pernyataan, pisahkan tindakan tersebut dengan koma. Misalnya, berikut ini:

```
"Action": [
      "route53-recovery-readiness:action1",
      "route53-recovery-readiness:action2"
         ]
```

Anda dapat menentukan beberapa tindakan menggunakan wildcard (*). Sebagai contoh, untuk menentukan semua tindakan yang dimulai dengan kata `Describe`, sertakan tindakan berikut:

```
"Action": "route53-recovery-readiness:Describe*"
```

Untuk melihat contoh kebijakan berbasis identitas ARC untuk pemeriksaan kesiapan, lihat. Contoh kebijakan berbasis identitas untuk pemeriksaan kesiapan di ARC

Sumber daya kebijakan untuk pemeriksaan kesiapan

Mendukung sumber daya kebijakan: Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen kebijakan JSON `Resource` menentukan objek yang menjadi target penerapan tindakan. Pernyataan harus menyertakan elemen `Resource` atau `NotResource`. Praktik terbaiknya, tentukan sumber daya menggunakan Amazon Resource Name (ARN). Anda dapat melakukan ini untuk tindakan yang mendukung jenis sumber daya tertentu, yang dikenal sebagai izin tingkat sumber daya.

Untuk tindakan yang tidak mendukung izin di tingkat sumber daya, misalnya operasi pencantuman, gunakan wildcard (*) untuk menunjukkan bahwa pernyataan tersebut berlaku untuk semua sumber daya.

```
"Resource": "*"
```

Untuk melihat daftar tindakan ARC untuk pergeseran zona, lihat Tindakan yang ditentukan oleh Kesiapan Pemulihan Amazon Route 53.

Untuk melihat contoh kebijakan berbasis identitas ARC untuk pemeriksaan kesiapan, lihat. Contoh kebijakan berbasis identitas untuk pemeriksaan kesiapan di ARC

Kunci kondisi kebijakan untuk pemeriksaan kesiapan

Mendukung kunci kondisi kebijakan khusus layanan: Yes

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen `Condition` (atau blok `Condition`) akan memungkinkan Anda menentukan kondisi yang menjadi dasar suatu pernyataan berlaku. Elemen `Condition` bersifat opsional. Anda dapat

membuat ekspresi bersyarat yang menggunakan operator kondisi, misalnya sama dengan atau kurang dari, untuk mencocokkan kondisi dalam kebijakan dengan nilai-nilai yang diminta.

Jika Anda menentukan beberapa elemen `Condition` dalam sebuah pernyataan, atau beberapa kunci dalam elemen `Condition` tunggal, maka AWS akan mengevaluasinya menggunakan operasi `AND` logis. Jika Anda menentukan beberapa nilai untuk satu kunci kondisi, AWS mengevaluasi kondisi menggunakan `OR` operasi logis. Semua kondisi harus dipenuhi sebelum izin pernyataan diberikan.

Anda juga dapat menggunakan variabel placeholder saat menentukan kondisi. Sebagai contoh, Anda dapat memberikan izin kepada pengguna IAM untuk mengakses sumber daya hanya jika izin tersebut mempunyai tanda yang sesuai dengan nama pengguna IAM mereka. Untuk informasi selengkapnya, lihat Elemen kebijakan IAM: variabel dan tanda dalam Panduan Pengguna IAM.

AWS mendukung kunci kondisi global dan kunci kondisi khusus layanan. Untuk melihat semua kunci kondisi AWS global, lihat kunci konteks kondisi AWS global di Panduan Pengguna IAM.

Untuk melihat daftar tindakan ARC untuk pemeriksaan kesiapan, lihat Kunci kondisi untuk Kesiapan Pemulihan Amazon Route 53

Untuk melihat tindakan dan sumber daya yang dapat Anda gunakan dengan kunci kondisi dengan pemeriksaan kesiapan, lihat Tindakan yang ditentukan oleh Kesiapan Pemulihan Amazon Route 53

Untuk melihat contoh kebijakan berbasis identitas ARC untuk pemeriksaan kesiapan, lihat. Contoh kebijakan berbasis identitas untuk pemeriksaan kesiapan di ARC

Daftar kontrol akses (ACLs) dalam pemeriksaan kesiapan

Mendukung ACLs: Tidak

Access control lists (ACLs) mengontrol prinsipal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACLs mirip dengan kebijakan berbasis sumber daya, meskipun mereka tidak menggunakan format dokumen kebijakan JSON.

Kontrol akses berbasis atribut (ABAC) dengan pemeriksaan kesiapan

Mendukung ABAC (tag dalam kebijakan): Sebagian

Kontrol akses berbasis atribut (ABAC) adalah strategi otorisasi yang menentukan izin berdasarkan atribut. Dalam AWS, atribut ini disebut tag. Anda dapat melampirkan tag ke entitas IAM (pengguna atau peran) dan ke banyak AWS sumber daya. Penandaan ke entitas dan sumber daya adalah langkah pertama dari ABAC. Kemudian rancanglah kebijakan ABAC untuk mengizinkan operasi ketika tanda milik prinsipal cocok dengan tanda yang ada di sumber daya yang ingin diakses.

ABAC sangat berguna di lingkungan yang berkembang dengan cepat dan berguna di situasi saat manajemen kebijakan menjadi rumit.

Untuk mengendalikan akses berdasarkan tanda, berikan informasi tentang tanda di elemen kondisi dari kebijakan menggunakan kunci kondisi `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, atau `aws:TagKeys`.

Jika sebuah layanan mendukung ketiga kunci kondisi untuk setiap jenis sumber daya, nilainya adalah Ya untuk layanan tersebut. Jika suatu layanan mendukung ketiga kunci kondisi untuk hanya beberapa jenis sumber daya, nilainya adalah Parsial.

Untuk informasi selengkapnya tentang ABAC, lihat Tentukan izin dengan otorisasi ABAC dalam Panduan Pengguna IAM. Untuk melihat tutorial yang menguraikan langkah-langkah pengaturan ABAC, lihat Menggunakan kontrol akses berbasis atribut (ABAC) dalam Panduan Pengguna IAM.

Kesiapan Pemulihan (pemeriksaan kesiapan) mendukung ABAC.

Menggunakan kredensi sementara dengan pemeriksaan kesiapan

Mendukung kredensial sementara: Ya

Beberapa Layanan AWS tidak berfungsi saat Anda masuk menggunakan kredensi sementara. Untuk informasi tambahan, termasuk yang Layanan AWS bekerja dengan kredensi sementara, lihat Layanan AWS yang bekerja dengan IAM di Panduan Pengguna IAM.

Anda menggunakan kredensi sementara jika Anda masuk AWS Management Console menggunakan metode apa pun kecuali nama pengguna dan kata sandi. Misalnya, ketika Anda mengakses AWS menggunakan tautan masuk tunggal (SSO) perusahaan Anda, proses tersebut secara otomatis membuat kredensi sementara. Anda juga akan secara otomatis membuat kredensial sementara ketika Anda masuk ke konsol sebagai seorang pengguna lalu beralih peran. Untuk informasi selengkapnya tentang peralihan peran, lihat Beralih dari pengguna ke peran IAM (konsol) dalam Panduan Pengguna IAM.

Anda dapat membuat kredenal sementara secara manual menggunakan API AWS CLI atau AWS . Anda kemudian dapat menggunakan kredensi sementara tersebut untuk mengakses. AWS AWS merekomendasikan agar Anda secara dinamis menghasilkan kredensi sementara alih-alih menggunakan kunci akses jangka panjang. Untuk informasi selengkapnya, lihat Kredensial keamanan sementara di IAM.

Izin utama lintas layanan untuk pemeriksaan kesiapan

Mendukung sesi akses maju (FAS): Ya

Saat Anda menggunakan entitas IAM (pengguna atau peran) untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Kebijakan memberikan izin kepada principal. Saat Anda menggunakan beberapa layanan, Anda mungkin melakukan tindakan yang kemudian memicu tindakan lain di layanan yang berbeda. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut.

Untuk melihat apakah tindakan dalam pemeriksaan kesiapan memerlukan tindakan dependen tambahan dalam kebijakan, lihat Kesiapan Pemulihan Amazon Route 53

Peran layanan untuk pemeriksaan kesiapan

Mendukung peran layanan: Tidak

Peran layanan adalah peran IAM yang diambil oleh sebuah layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, mengubah, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat Buat sebuah peran untuk mendelegasikan izin ke Layanan AWS dalam Panduan pengguna IAM.

Peran terkait layanan untuk pemeriksaan kesiapan

Mendukung peran terkait layanan: Ya

Peran terkait layanan adalah jenis peran layanan yang ditautkan ke. Layanan AWS Layanan tersebut dapat menjalankan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Anda Akun AWS dan dimiliki oleh layanan. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.

Untuk detail tentang membuat atau mengelola peran terkait layanan ARC, lihat. Menggunakan peran terkait layanan untuk pemeriksaan kesiapan di ARC

Untuk detail tentang pembuatan atau manajemen peran terkait layanan, lihat Layanan AWS yang berfungsi dengan IAM. Cari layanan dalam tabel yang memiliki `Yes` di kolom Peran terkait layanan. Pilih tautan Ya untuk melihat dokumentasi peran terkait layanan untuk layanan tersebut.

Contoh kebijakan berbasis identitas untuk pemeriksaan kesiapan di ARC

Secara default, pengguna dan peran tidak memiliki izin untuk membuat atau memodifikasi sumber daya ARC. Mereka juga tidak dapat melakukan tugas dengan menggunakan AWS Management Console, AWS Command Line Interface (AWS CLI), atau AWS API. Untuk memberikan izin kepada pengguna untuk melakukan tindakan di sumber daya yang mereka perlukan, administrator IAM dapat

membuat kebijakan IAM. Administrator kemudian dapat menambahkan kebijakan IAM ke peran, dan pengguna dapat mengambil peran.

Untuk mempelajari cara membuat kebijakan berbasis identitas IAM dengan menggunakan contoh dokumen kebijakan JSON ini, lihat Membuat kebijakan IAM (konsol) di Panduan Pengguna IAM.

Untuk detail tentang tindakan dan jenis sumber daya yang ditentukan oleh ARC, termasuk format ARNs untuk setiap jenis sumber daya, lihat Kunci tindakan, sumber daya, dan kondisi untuk Amazon Application Recovery Controller (ARC) di Referensi Otorisasi Layanan.

Topik

- Praktik terbaik kebijakan
- Contoh: Kesiapan memeriksa akses konsol
- Contoh: Kesiapan memeriksa tindakan API untuk pemeriksaan kesiapan

Praktik terbaik kebijakan

Kebijakan berbasis identitas menentukan apakah seseorang dapat membuat, mengakses, atau menghapus sumber daya ARC di akun Anda. Tindakan ini membuat Akun AWS Anda dikenai biaya. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

- Mulailah dengan kebijakan AWS terkelola dan beralih ke izin hak istimewa paling sedikit — Untuk mulai memberikan izin kepada pengguna dan beban kerja Anda, gunakan kebijakan AWS terkelola yang memberikan izin untuk banyak kasus penggunaan umum. Mereka tersedia di Anda Akun AWS. Kami menyarankan Anda mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola AWS pelanggan yang khusus untuk kasus penggunaan Anda. Untuk informasi selengkapnya, lihat Kebijakan yang dikelola AWS atau Kebijakan yang dikelola AWS untuk fungsi tugas dalam Panduan Pengguna IAM.
- Menerapkan izin dengan hak akses paling rendah – Ketika Anda menetapkan izin dengan kebijakan IAM, hanya berikan izin yang diperlukan untuk melakukan tugas. Anda melakukannya dengan mendefinisikan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi tertentu, yang juga dikenal sebagai izin dengan hak akses paling rendah. Untuk informasi selengkapnya tentang cara menggunakan IAM untuk mengajukan izin, lihat Kebijakan dan izin dalam IAM dalam Panduan Pengguna IAM.
- Gunakan kondisi dalam kebijakan IAM untuk membatasi akses lebih lanjut – Anda dapat menambahkan suatu kondisi ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. Sebagai contoh, Anda dapat menulis kondisi kebijakan untuk menentukan bahwa semua

permintaan harus dikirim menggunakan SSL. Anda juga dapat menggunakan ketentuan untuk memberikan akses ke tindakan layanan jika digunakan melalui yang spesifik Layanan AWS, seperti AWS CloudFormation. Untuk informasi selengkapnya, lihat Elemen kebijakan JSON IAM: Kondisi dalam Panduan Pengguna IAM.

- Gunakan IAM Access Analyzer untuk memvalidasi kebijakan IAM Anda untuk memastikan izin yang aman dan fungsional – IAM Access Analyzer memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan tersebut mematuhi bahasa kebijakan IAM (JSON) dan praktik terbaik IAM. IAM Access Analyzer menyediakan lebih dari 100 pemeriksaan kebijakan dan rekomendasi yang dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang aman dan fungsional. Untuk informasi selengkapnya, lihat Validasi kebijakan dengan IAM Access Analyzer dalam Panduan Pengguna IAM.
- Memerlukan otentikasi multi-faktor (MFA) - Jika Anda memiliki skenario yang mengharuskan pengguna IAM atau pengguna root di Anda, Akun AWS aktifkan MFA untuk keamanan tambahan. Untuk meminta MFA ketika operasi API dipanggil, tambahkan kondisi MFA pada kebijakan Anda. Untuk informasi selengkapnya, lihat Amankan akses API dengan MFA dalam Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang praktik terbaik dalam IAM, lihat Praktik terbaik keamanan di IAM dalam Panduan Pengguna IAM.

Contoh: Kesiapan memeriksa akses konsol

Untuk mengakses konsol Amazon Application Recovery Controller (ARC), Anda harus memiliki seperangkat izin minimum. Izin ini harus memungkinkan Anda untuk membuat daftar dan melihat detail tentang sumber daya ARC di Anda Akun AWS. Jika Anda membuat kebijakan berbasis identitas yang lebih ketat daripada izin minimum yang diperlukan, konsol tidak akan berfungsi sebagaimana mestinya untuk entitas (pengguna atau peran) dengan kebijakan tersebut.

Anda tidak perlu mengizinkan izin konsol minimum untuk pengguna yang melakukan panggilan hanya ke AWS CLI atau AWS API. Sebagai gantinya, izinkan akses hanya ke tindakan yang sesuai dengan operasi API yang coba mereka lakukan.

Untuk memastikan bahwa pengguna dan peran masih dapat menggunakan konsol pemeriksaan kesiapan saat Anda mengizinkan akses hanya ke operasi API tertentu, lampirkan juga kebijakan `ReadOnly` AWS terkelola untuk pemeriksaan kesiapan ke entitas. Untuk informasi selengkapnya, lihat pemeriksaan kesiapan Periksa halaman kebijakan terkelola atau Menambahkan izin ke pengguna di Panduan Pengguna IAM.

Untuk melakukan beberapa tugas, pengguna harus memiliki izin untuk membuat peran terkait layanan yang terkait dengan pemeriksaan kesiapan di ARC. Untuk mempelajari selengkapnya, lihat Menggunakan peran terkait layanan untuk pemeriksaan kesiapan di ARC.

Untuk memberi pengguna akses penuh untuk menggunakan fitur pemeriksaan kesiapan melalui konsol, lampirkan kebijakan seperti berikut ini kepada pengguna:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                   "route53-recovery-readiness:CreateCell",
                   "route53-recovery-readiness:CreateCrossAccountAuthorization",
                   "route53-recovery-readiness:CreateReadinessCheck",
                   "route53-recovery-readiness:CreateRecoveryGroup",
                   "route53-recovery-readiness:CreateResourceSet",
                   "route53-recovery-readiness:DeleteCell",
                   "route53-recovery-readiness:DeleteCrossAccountAuthorization",
                   "route53-recovery-readiness:DeleteReadinessCheck",
                   "route53-recovery-readiness:DeleteRecoveryGroup",
                   "route53-recovery-readiness:DeleteResourceSet",
                   "route53-recovery-readiness:GetArchitectureRecommendations",
                   "route53-recovery-readiness:GetCell",
                   "route53-recovery-readiness:GetCellReadinessSummary",
                   "route53-recovery-readiness:GetReadinessCheck",
                   "route53-recovery-readiness:GetReadinessCheckResourceStatus",
                   "route53-recovery-readiness:GetReadinessCheckStatus",
                   "route53-recovery-readiness:GetRecoveryGroup",
                   "route53-recovery-readiness:GetRecoveryGroupReadinessSummary",
                   "route53-recovery-readiness:GetResourceSet",
                   "route53-recovery-readiness:ListCells",
                   "route53-recovery-readiness:ListCrossAccountAuthorizations",
                   "route53-recovery-readiness:ListReadinessChecks",
                   "route53-recovery-readiness:ListRecoveryGroups",
                   "route53-recovery-readiness:ListResourceSets",
                   "route53-recovery-readiness:ListRules",
                   "route53-recovery-readiness:UpdateCell",
                   "route53-recovery-readiness:UpdateReadinessCheck",
                   "route53-recovery-readiness:UpdateRecoveryGroup",
                   "route53-recovery-readiness:UpdateResourceSet"
            ],
```

```
            "Resource": "*"
        }
    ]
}
```

Contoh: Kesiapan memeriksa tindakan API untuk pemeriksaan kesiapan

Untuk memastikan bahwa pengguna dapat menggunakan tindakan ARC API untuk bekerja dengan bidang kontrol pemeriksaan kesiapan ARC — misalnya, untuk membuat grup pemulihan, kumpulan sumber daya, dan pemeriksaan kesiapan — lampirkan kebijakan yang sesuai dengan operasi API yang perlu dikerjakan pengguna, seperti yang dijelaskan di bawah ini.

Untuk melakukan beberapa tugas, pengguna harus memiliki izin untuk membuat peran terkait layanan yang terkait dengan pemeriksaan kesiapan di ARC. Untuk mempelajari selengkapnya, lihat Menggunakan peran terkait layanan untuk pemeriksaan kesiapan di ARC.

Untuk bekerja dengan operasi API untuk pemeriksaan kesiapan, lampirkan kebijakan seperti berikut ini kepada pengguna:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                   "route53-recovery-readiness:CreateCell",
                   "route53-recovery-readiness:CreateCrossAccountAuthorization",
                   "route53-recovery-readiness:CreateReadinessCheck",
                   "route53-recovery-readiness:CreateRecoveryGroup",
                   "route53-recovery-readiness:CreateResourceSet",
                   "route53-recovery-readiness:DeleteCell",
                   "route53-recovery-readiness:DeleteCrossAccountAuthorization",
                   "route53-recovery-readiness:DeleteReadinessCheck",
                   "route53-recovery-readiness:DeleteRecoveryGroup",
                   "route53-recovery-readiness:DeleteResourceSet",
                   "route53-recovery-readiness:GetArchitectureRecommendations",
                   "route53-recovery-readiness:GetCell",
                   "route53-recovery-readiness:GetCellReadinessSummary",
                   "route53-recovery-readiness:GetReadinessCheck",
                   "route53-recovery-readiness:GetReadinessCheckResourceStatus",
                   "route53-recovery-readiness:GetReadinessCheckStatus",
                   "route53-recovery-readiness:GetRecoveryGroup",
                   "route53-recovery-readiness:GetRecoveryGroupReadinessSummary",
```

```
                "route53-recovery-readiness:GetResourceSet",
                "route53-recovery-readiness:ListCells",
                "route53-recovery-readiness:ListCrossAccountAuthorizations",
                "route53-recovery-readiness:ListReadinessChecks",
                "route53-recovery-readiness:ListRecoveryGroups",
                "route53-recovery-readiness:ListResourceSets",
                "route53-recovery-readiness:ListRules",
                "route53-recovery-readiness:ListTagsForResources",
                "route53-recovery-readiness:UpdateCell",
                "route53-recovery-readiness:UpdateReadinessCheck",
                "route53-recovery-readiness:UpdateRecoveryGroup",
                "route53-recovery-readiness:UpdateResourceSet",
                "route53-recovery-readiness:TagResource",
                "route53-recovery-readiness:UntagResource"
            ],
            "Resource": "*"
        }
    ]
}
```

Menggunakan peran terkait layanan untuk pemeriksaan kesiapan di ARC

Amazon Application Recovery Controller menggunakan peran terkait layanan AWS Identity and Access Management (IAM). Peran terkait layanan adalah jenis unik peran IAM yang ditautkan langsung ke layanan — dalam hal ini, ARC. Peran terkait layanan telah ditentukan sebelumnya oleh ARC dan mencakup semua izin yang diperlukan layanan untuk memanggil AWS layanan lain atas nama Anda untuk tujuan tertentu.

Peran terkait layanan membuat pengaturan ARC lebih mudah karena Anda tidak perlu menambahkan izin yang diperlukan secara manual. ARC mendefinisikan izin peran terkait layanan, dan kecuali ditentukan lain, hanya ARC yang dapat mengambil perannya. Izin-izin yang ditentukan mencakup kebijakan kepercayaan dan kebijakan izin, serta bahwa kebijakan izin tidak dapat dilampirkan ke entitas IAM lainnya.

Anda dapat menghapus peran tertaut layanan hanya setelah terlebih dahulu menghapus sumber dayanya yang terkait. Ini melindungi sumber daya ARC Anda karena Anda tidak dapat secara tidak sengaja menghapus izin untuk mengakses sumber daya.

Untuk informasi tentang layanan lain yang mendukung peran terkait layanan, lihat AWS Layanan yang bekerja dengan IAM dan cari layanan yang memiliki Ya di kolom peran terkait layanan. Pilih Ya bersama tautan untuk melihat dokumentasi peran tertaut layanan untuk layanan tersebut.

ARC memiliki peran terkait layanan berikut, yang dijelaskan dalam Bab ini:

- ARC menggunakan peran terkait layanan bernama Route53 RecoveryReadinessServiceRolePolicy untuk mengakses sumber daya dan konfigurasi untuk memeriksa kesiapan.
- ARC menggunakan peran terkait layanan yang dinamai untuk praktik autoshift berjalan, untuk memantau CloudWatch alarm Amazon yang disediakan pelanggan dan AWS Health Dashboard peristiwa pelanggan, dan untuk memulai praktik berjalan.

Izin peran terkait layanan untuk Route53 RecoveryReadinessServiceRolePolicy

ARC menggunakan peran terkait layanan bernama Route53 RecoveryReadinessServiceRolePolicy untuk mengakses sumber daya dan konfigurasi untuk memeriksa kesiapan. Bagian ini menjelaskan izin untuk peran terkait layanan, dan informasi tentang membuat, mengedit, dan menghapus peran.

Izin peran terkait layanan untuk Route53 RecoveryReadinessServiceRolePolicy

Peran terkait layanan ini menggunakan kebijakan terkelola.
`Route53RecoveryReadinessServiceRolePolicy`

Peran RecoveryReadinessServiceRolePolicy terkait layanan Route53 mempercayai layanan berikut untuk mengambil peran:

- `route53-recovery-readiness.amazonaws.com`

Untuk melihat izin kebijakan ini, lihat Route53 RecoveryReadinessServiceRolePolicy di Referensi Kebijakan AWS Terkelola.

Anda harus mengonfigurasi izin untuk mengizinkan entitas IAM (seperti pengguna, grup, atau peran) untuk membuat, mengedit, atau menghapus peran terkait layanan. Untuk informasi selengkapnya, lihat Izin peran tertaut layanan dalam Panduan Pengguna IAM.

Membuat peran terkait RecoveryReadinessServiceRolePolicy layanan Route53 untuk ARC

Anda tidak perlu membuat peran terkait RecoveryReadinessServiceRolePolicy layanan Route53 secara manual. Saat Anda membuat pemeriksaan kesiapan pertama atau otorisasi lintas akun di AWS Management Console, the, atau AWS API AWS CLI, ARC membuat peran terkait layanan untuk Anda.

Jika Anda menghapus peran tertaut layanan ini, dan ingin membuatnya lagi, Anda dapat mengulangi proses yang sama untuk membuat kembali peran tersebut di akun Anda. Saat Anda membuat

pemeriksaan kesiapan pertama atau otorisasi lintas akun, ARC membuat peran terkait layanan untuk Anda lagi.

Mengedit peran terkait RecoveryReadinessServiceRolePolicy layanan Route53 untuk ARC

ARC tidak mengizinkan Anda mengedit peran terkait RecoveryReadinessServiceRolePolicy layanan Route53. Setelah membuat peran terkait layanan, Anda tidak dapat mengubah nama peran karena entitas lain mungkin mereferensikan peran tersebut. Namun, Anda dapat mengedit penjelasan peran menggunakan IAM. Untuk informasi selengkapnya, lihat Mengedit peran tertaut layanan dalam Panduan Pengguna IAM.

Menghapus peran terkait RecoveryReadinessServiceRolePolicy layanan Route53 untuk ARC

Jika Anda tidak perlu lagi menggunakan fitur atau layanan yang memerlukan peran terkait layanan, kami merekomendasikan Anda menghapus peran tersebut. Dengan begitu, Anda tidak memiliki entitas yang tidak digunakan yang tidak dipantau atau dipelihara secara aktif. Tetapi, Anda harus membersihkan sumber daya peran yang terhubung dengan layanan sebelum menghapusnya secara manual.

Setelah Anda menghapus pemeriksaan kesiapan dan otorisasi lintas akun, Anda dapat menghapus peran terkait layanan RecoveryReadinessServiceRolePolicyRoute53. Untuk informasi selengkapnya tentang pemeriksaan kesiapan, lihatPemeriksaan kesiapan di ARC. Untuk informasi selengkapnya tentang otorisasi lintas akun, lihat. Membuat otorisasi lintas akun di ARC

> **Note**
>
> Jika layanan ARC menggunakan peran saat Anda mencoba menghapus sumber daya, penghapusan peran layanan mungkin gagal. Jika itu terjadi, tunggu beberapa menit dan coba lagi untuk menghapus peran.

Untuk menghapus peran terkait layanan secara manual menggunakan IAM

Gunakan konsol IAM, the AWS CLI, atau AWS API untuk menghapus peran terkait layanan Route53RecoveryReadinessServiceRolePolicy. Untuk informasi selengkapnya, lihat Menghapus peran tertaut layanan dalam Panduan Pengguna IAM.

Pembaruan peran terkait layanan ARC untuk pemeriksaan kesiapan

Untuk pembaruan kebijakan AWS terkelola untuk peran terkait layanan ARC, lihat tabel pembaruan kebijakan AWS terkelola untuk ARC. Anda juga dapat berlangganan peringatan RSS otomatis di halaman riwayat Dokumen ARC.

AWS kebijakan terkelola untuk pemeriksaan kesiapan di ARC

Kebijakan AWS terkelola adalah kebijakan mandiri yang dibuat dan dikelola oleh AWS. AWS Kebijakan terkelola dirancang untuk memberikan izin bagi banyak kasus penggunaan umum sehingga Anda dapat mulai menetapkan izin kepada pengguna, grup, dan peran.

Perlu diingat bahwa kebijakan AWS terkelola mungkin tidak memberikan izin hak istimewa paling sedikit untuk kasus penggunaan spesifik Anda karena tersedia untuk digunakan semua pelanggan. AWS Kami menyarankan Anda untuk mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola pelanggan yang khusus untuk kasus penggunaan Anda.

Anda tidak dapat mengubah izin yang ditentukan dalam kebijakan AWS terkelola. Jika AWS memperbarui izin yang ditentukan dalam kebijakan AWS terkelola, pembaruan akan memengaruhi semua identitas utama (pengguna, grup, dan peran) yang dilampirkan kebijakan tersebut. AWS kemungkinan besar akan memperbarui kebijakan AWS terkelola saat baru Layanan AWS diluncurkan atau operasi API baru tersedia untuk layanan yang ada.

Untuk informasi selengkapnya, lihat Kebijakan terkelola AWS dalam Panduan Pengguna IAM.

AWS kebijakan terkelola: Route53 RecoveryReadinessServiceRolePolicy

Anda tidak dapat melampirkan `Route53RecoveryReadinessServiceRolePolicy` ke entitas IAM Anda. Kebijakan ini dilampirkan ke peran terkait layanan yang memungkinkan Amazon Application Recovery Controller (ARC) mengakses AWS layanan dan sumber daya yang digunakan atau dikelola oleh ARC. Untuk informasi selengkapnya, lihat Menggunakan peran terkait layanan untuk pemeriksaan kesiapan di ARC.

AWS kebijakan terkelola: AmazonRoute 53 RecoveryReadinessFullAccess

Anda dapat melampirkan `AmazonRoute53RecoveryReadinessFullAccess` ke entitas IAM Anda. Kebijakan ini memberikan akses penuh ke tindakan untuk bekerja dengan kesiapan pemulihan (pemeriksaan kesiapan) di ARC. Lampirkan ke pengguna IAM dan prinsipal lain yang membutuhkan akses penuh ke tindakan kesiapan pemulihan.

Untuk melihat izin kebijakan ini, lihat AmazonRoute53 RecoveryReadinessFullAccess di Referensi Kebijakan AWS Terkelola.

AWS kebijakan terkelola: AmazonRoute 53 RecoveryReadinessReadOnlyAccess

Anda dapat melampirkan `AmazonRoute53RecoveryReadinessReadOnlyAccess` ke entitas IAM Anda. Kebijakan ini memberikan akses hanya-baca ke tindakan untuk bekerja dengan kesiapan pemulihan di ARC. Ini berguna bagi pengguna yang perlu melihat status kesiapan dan konfigurasi grup pemulihan. Pengguna ini tidak dapat membuat, memperbarui, atau menghapus sumber daya kesiapan pemulihan.

Untuk melihat izin kebijakan ini, lihat AmazonRoute53 RecoveryReadinessReadOnlyAccess di Referensi Kebijakan AWS Terkelola.

Pembaruan untuk kebijakan AWS terkelola untuk kesiapan

Untuk detail tentang pembaruan kebijakan AWS terkelola untuk pemeriksaan kesiapan di ARC sejak layanan ini mulai melacak perubahan ini, lihatPembaruan kebijakan AWS terkelola untuk Amazon Application Recovery Controller (ARC). Untuk peringatan otomatis tentang perubahan pada halaman ini, berlangganan umpan RSS di halaman riwayat Dokumen ARC.

Kuota untuk pemeriksaan kesiapan

Pemeriksaan kesiapan di Amazon Application Recovery Controller (ARC) tunduk pada kuota berikut (sebelumnya disebut sebagai batas).

Entitas	Kuota
Jumlah grup pemulihan per akun	5
Jumlah sel per akun	15
Jumlah sel bersarang per sel	3
Jumlah sel per kelompok pemulihan	3
Jumlah sumber daya per sel	10
Jumlah sumber daya per kelompok pemulihan	10
Jumlah sumber daya per set sumber daya	6

Entitas	Kuota
Jumlah set sumber daya per akun	200
Jumlah cek kesiapan per akun	200
Jumlah otorisasi lintas akun	100

Sakelar wilayah di ARC

Anda dapat menggunakan sakelar Wilayah di ARC untuk mengatur tugas pemulihan skala besar dan kompleks untuk sumber daya aplikasi Anda di seluruh AWS akun, untuk membantu memastikan kelangsungan bisnis dan mengurangi biaya operasional. Sakelar wilayah menyediakan solusi terpusat dan dapat diamati yang dapat Anda lakukan secara manual, atau otomatis dengan menggunakan pemicu alarm Amazon CloudWatch . Jika Wilayah AWS menjadi terganggu, Anda dapat menjalankan rencana yang Anda buat dengan menggunakan sakelar Wilayah untuk gagal atau mengalihkan sumber daya Anda ke Wilayah lain. Ini memastikan bahwa aplikasi Anda dapat terus beroperasi, berjalan dengan sehat Wilayah AWS.

Sakelar wilayah dibangun berdasarkan konsep rencana, yang Anda desain dan konfigurasikan untuk kebutuhan pemulihan spesifik Anda. Setiap rencana mencakup alur kerja yang terdiri dari langkah-langkah. Sebuah langkah menjalankan satu atau lebih blok eksekusi, yang mana sakelar Region berjalan secara paralel atau berurutan, untuk menyelesaikan pemulihan aplikasi. Setiap blok eksekusi menangani tugas yang berbeda, seperti mengalihkan sumber daya atau mengelola pengalihan lalu lintas untuk aplikasi Anda. Untuk fleksibilitas yang lebih besar, Anda dapat membuat rencana bersarang, dengan menambahkan paket anak ke rencana induk secara keseluruhan.

Sakelar wilayah meliputi yang berikut:

- Support untuk active/passive dan active/active konfigurasi. Anda dapat failover dan failback jika memiliki konfigurasi active/passive Multi-region, atau shift-away dan return jika aplikasi Anda diatur seperti di beberapa Wilayah. active/active
- Dukungan lintas akun untuk sumber daya aplikasi yang Anda sertakan dalam pemulihan aplikasi Anda. Anda juga dapat membagikan paket peralihan Wilayah di seluruh akun.
- Failover atau switchover otomatis, dengan memicu eksekusi paket berdasarkan alarm Amazon. CloudWatch Atau, Anda dapat memilih untuk menjalankan rencana peralihan Wilayah secara manual.

- Dasbor berfitur lengkap yang memberi Anda visibilitas waktu nyata ke dalam proses pemulihan.
- Bidang data di masing-masing Wilayah AWS, sehingga Anda dapat menjalankan paket sakelar Wilayah Anda tanpa mengambil ketergantungan pada Wilayah yang Anda nonaktifkan.

Sakelar wilayah sepenuhnya dikelola oleh AWS. Menggunakan sakelar Wilayah memungkinkan Anda mendapatkan keuntungan dari ketahanan platform pemulihan yang berfokus pada persyaratan spesifik aplikasi Anda, alih-alih membangun dan memelihara skrip, dan mengumpulkan data tentang pemulihan secara manual.

Tentang sakelar Wilayah

Dengan sakelar Wilayah, Anda dapat mengatur langkah-langkah spesifik untuk mengganti aplikasi Multi-wilayah Anda berjalan. Wilayah AWS

Sakelar wilayah dibangun berdasarkan konsep rencana, yang Anda desain dan konfigurasikan untuk kebutuhan pemulihan spesifik Anda. Setiap rencana mencakup alur kerja yang terdiri dari langkah-langkah. Sebuah langkah menjalankan satu atau lebih blok eksekusi, yang mana sakelar Region berjalan secara paralel atau berurutan, untuk menyelesaikan pemulihan aplikasi. Setiap blok eksekusi menangani tugas yang berbeda, seperti mengalihkan sumber daya atau mengelola pengalihan lalu lintas untuk aplikasi Anda. Untuk fleksibilitas yang lebih besar, Anda dapat membuat rencana bersarang, dengan menambahkan paket anak.

Setiap kali Anda membuat, atau memperbarui, paket, sakelar Wilayah melakukan evaluasi rencana, untuk memastikan bahwa tidak ada masalah dengan izin IAM, konfigurasi sumber daya, atau kapasitas berjalan. Sakelar wilayah menjalankan evaluasi ini secara teratur, dan menghasilkan peringatan untuk masalah apa pun yang ditemukannya.

Sakelar wilayah juga menghitung nilai waktu pemulihan aktual untuk setiap pelaksanaan rencana, untuk membantu Anda mengevaluasi apakah rencana tersebut memenuhi tujuan Anda. Anda dapat melihat waktu pemulihan dan detail lainnya tentang eksekusi rencana di dasbor sakelar Wilayah di. AWS Management Console Untuk informasi selengkapnya, lihat Dasbor sakelar wilayah.

Untuk mempelajari lebih lanjut tentang masing-masing area ini di sakelar Wilayah, lihat bagian berikut.

Rencana peralihan wilayah

Rencana peralihan Wilayah adalah sumber daya tingkat atas di sakelar Wilayah. Anda harus mencakup rencana Anda ke aplikasi Multi-wilayah tertentu. Sebuah rencana memungkinkan Anda

untuk membangun alur kerja untuk memulihkan aplikasi Anda dengan menjalankan serangkaian blok eksekusi peralihan Wilayah yang mengaktifkan atau menonaktifkan aplikasi Anda dan sumber dayanya, termasuk sumber daya lintas akun, dalam Wilayah AWS yang Anda tentukan.

Rencana terdiri dari satu atau beberapa alur kerja, untuk memungkinkan Anda mengaktifkan atau menonaktifkan yang spesifik. Wilayah AWS Anda dapat mengonfigurasi blok eksekusi dalam alur kerja untuk dijalankan secara berurutan, atau Anda dapat menentukan bahwa beberapa blok berjalan secara paralel.

Untuk paket yang Anda konfigurasikan untuk pendekatan active/passive Multi-wilayah, Anda membuat salah satu alur kerja yang dapat digunakan untuk mengaktifkan salah satu Wilayah Anda, atau dua alur kerja aktivasi terpisah, satu untuk setiap Wilayah. Untuk paket yang Anda konfigurasikan untuk pendekatan aktif/aktif, Anda membuat satu alur kerja untuk mengaktifkan Wilayah dan satu alur kerja untuk menonaktifkan Wilayah Anda.

Wilayah AWS adalah lokasi geografis di seluruh dunia di mana AWS cluster data center. Setiap Wilayah dirancang untuk sepenuhnya terisolasi dari Wilayah lain, memberikan toleransi dan stabilitas kesalahan. Saat Anda menggunakan sakelar Region, Anda perlu mempertimbangkan Wilayah mana aplikasi Anda digunakan dan Wilayah mana yang ingin Anda gunakan untuk pemulihan.

Sakelar wilayah mendukung pemulihan antara dua Wilayah AWS tempat layanan tersedia. Saat mengonfigurasi paket peralihan Wilayah, Anda menentukan Wilayah tempat aplikasi Anda digunakan dan pendekatan pemulihan yang ingin Anda gunakan: active/passive atau aktif/aktif.

Misalnya, Anda mungkin memiliki pendekatan active/passive Multi-wilayah dengan us-east-1 sebagai Region utama dan us-west-2 sebagai Region siaga. Untuk memulihkan aplikasi Anda dari masalah operasional yang memengaruhi aplikasi di us-east-1, Anda dapat menjalankan rencana peralihan Wilayah untuk mengaktifkan us-west-2. Ini akan mengakibatkan aplikasi beralih dari sumber daya di us-east-1 ke sumber daya di us-west-2.

Rencana peralihan wilayah dijalankan menggunakan izin yang terkait dengan peran IAM yang Anda tentukan saat membuat paket.

Anda dapat membuat beberapa paket, satu untuk setiap aplikasi Multi-wilayah Anda, dan kemudian mengatur pemulihan di seluruh paket ini dalam urutan yang Anda butuhkan dengan membuat paket induk. Rencana induk adalah rencana yang menggunakan blok eksekusi rencana sakelar Wilayah sebagai langkah. Hirarki rencana terbatas pada dua tingkatan (orang tua dan anak), tetapi Anda dapat menyertakan beberapa paket anak di bawah rencana induk yang sama.

Alur kerja dan blok eksekusi

Setelah membuat rencana peralihan Wilayah, Anda harus menambahkan satu atau beberapa alur kerja ke paket, untuk menentukan langkah-langkah yang ingin dilakukan rencana untuk pemulihan aplikasi Anda. Untuk setiap alur kerja, Anda menambahkan blok eksekusi untuk menyelesaikan tugas tertentu, seperti meningkatkan sumber daya atau memperbarui kontrol perutean untuk mengubah rute lalu lintas. Blok eksekusi memungkinkan Anda menentukan tugas-tugas ini dan urutan penyelesaiannya. Dengan membuat paket bersarang, Anda juga dapat mengatur urutan pemulihan beberapa aplikasi ke Wilayah yang Anda aktifkan.

Anda dapat menambahkan blok eksekusi dalam alur kerja secara berurutan, atau Anda dapat menambahkan satu atau beberapa blok eksekusi secara paralel. Selain itu, tergantung pada sumber daya, Anda dapat memiliki opsi untuk menjalankan blok eksekusi dengan eksekusi yang anggun (terencana) atau tidak teratur (tidak direncanakan).

- Eksekusi anggun: Alur kerja eksekusi yang direncanakan. Ketika lingkungan Anda sehat, Anda dapat menggunakan alur kerja yang anggun untuk menjalankan semua langkah untuk pelaksanaan rencana yang teratur.
- Eksekusi yang tidak menyenangkan: Eksekusi yang tidak direncanakan. Mode alur kerja yang tidak sopan hanya menggunakan langkah dan tindakan yang diperlukan. Mode ini mengubah perilaku blok eksekusi dalam alur kerja atau melewatkan blok eksekusi tertentu.

Terakhir, Anda juga dapat mengonfigurasi sumber daya lintas akun untuk blok eksekusi. Pertama, Anda harus mengonfigurasi izin, dengan mengikuti panduan diDukungan lintas akun di sakelar Wilayah. Setelah menyiapkan peran IAM yang diperlukan, Anda dapat menambahkan sumber daya lintas akun di blok eksekusi dalam alur kerja rencana Anda. Untuk menambahkan sumber daya lintas akun, saat menambahkan blok eksekusi, Anda menentukan peran IAM target yang memiliki izin ke sumber daya lainnya. Akun AWS Anda juga harus menentukan ID eksternal yang Anda berikan dalam kebijakan kepercayaan untuk peran lintas akun. Untuk detail tentang membuat peran IAM yang diperlukan, lihatAkses sumber daya lintas akun.

Untuk mempelajari lebih lanjut tentang alur kerja, lihatBuat alur kerja rencana peralihan Wilayah. Untuk detail tentang setiap jenis blok eksekusi, termasuk langkah-langkah konfigurasi, cara kerjanya, dan apa yang dievaluasi sebagai bagian dari evaluasi rencana, lihatTambahkan blok eksekusi.

Evaluasi rencana

Evaluasi rencana adalah proses otomatis yang dijalankan sakelar Wilayah saat rencana dibuat atau diperbarui, dan kemudian setiap 30 menit setelah itu, selama kondisi tunak. Proses evaluasi memverifikasi beberapa aspek penting dari konfigurasi rencana dan konfigurasi sumber daya. Evaluasi termasuk memverifikasi izin IAM, konfigurasi sumber daya, dan kapasitas berjalan.

Jika Region switch menemukan masalah yang mungkin mencegah eksekusi rencana berhasil, itu akan menghasilkan peringatan evaluasi rencana, yang disorot pada halaman detail paket di konsol. Anda juga dapat menggunakan peringatan evaluasi paket dengan Amazon EventBridge, atau Anda dapat melihat peringatan menggunakan API sakelar Wilayah.

Anda dapat melihat detail dan perbaikan yang disarankan untuk masalah yang muncul dalam evaluasi rencana tab pada halaman detail rencana. Kami menyarankan Anda juga menguji pemulihan aplikasi dengan menjalankan rencana peralihan Wilayah Anda, dan Anda tidak hanya mengandalkan evaluasi rencana peralihan Wilayah untuk menguji apakah rencana pemulihan Anda akan berfungsi seperti yang Anda harapkan.

Alarm regional dan waktu pemulihan aktual

Sakelar wilayah menghitung nilai waktu pemulihan aktual untuk setiap eksekusi paket, yang dapat Anda lihat setelah eksekusi rencana. Waktu pemulihan aktual ditampilkan pada halaman detail eksekusi rencana, sehingga Anda dapat membandingkan waktu aktual dengan tujuan waktu pemulihan yang Anda tentukan saat membuat rencana.

Waktu pemulihan aktual dihitung karena total waktu yang diperlukan untuk menyelesaikan eksekusi rencana, dan waktu tambahan apa pun yang berlalu sebelum CloudWatch alarm Amazon tertentu yang Anda konfigurasikan kembali ke status hijau.

Untuk mendukung penghitungan waktu pemulihan aktual yang akurat untuk pelaksanaan rencana, tambahkan CloudWatch alarm Amazon Regional ke paket peralihan Wilayah yang memberikan sinyal tentang kesehatan aplikasi Anda di setiap Wilayah. Ketika rencana dijalankan, sakelar Wilayah menggunakan alarm kesehatan aplikasi ini untuk menentukan kapan aplikasi Anda sehat kembali. Kemudian, sakelar Wilayah menghitung waktu pemulihan aktual berdasarkan waktu yang diperlukan untuk mengeksekusi rencana Anda ditambahkan ke waktu yang diperlukan agar aplikasi Anda kembali sehat, berdasarkan alarm kesehatan aplikasi yang Anda tentukan.

Wilayah AWS

Sakelar wilayah tersedia di semua komersial Wilayah AWS.

Untuk informasi terperinci tentang dukungan Regional dan titik akhir layanan untuk Amazon Application Recovery Controller (ARC), lihat titik akhir dan kuota Amazon Application Recovery Controller (ARC) di Referensi Umum Amazon Web Services.

Komponen sakelar wilayah

Berikut ini adalah komponen, dan konsep tentang, fitur sakelar Wilayah di Amazon Application Recovery Controller (ARC).

Rencana

Rencana adalah proses pemulihan mendasar untuk aplikasi Anda. Anda membuat rencana dengan membangun satu atau beberapa alur kerja dengan blok eksekusi untuk dijalankan secara berurutan atau paralel. Kemudian, ketika ada gangguan Regional, Anda menjalankan rencana untuk menyelesaikan pemulihan aplikasi Anda dengan menggeser aplikasi untuk berjalan di Wilayah yang sehat.

Rencana anak

Rencana anak adalah rencana mandiri yang dapat dijalankan dari dalam rencana induk, untuk mengoordinasikan skenario pemulihan aplikasi yang lebih kompleks. Anda dapat membuat sarang paket peralihan Wilayah satu tingkat.

Alur kerja

Rencana peralihan Wilayah mencakup satu atau beberapa alur kerja. Alur kerja terdiri dari blok eksekusi yang Anda tentukan untuk dijalankan secara paralel atau berurutan, yang menyelesaikan aktivasi atau penonaktifan Wilayah sebagai bagian dari rencana pemulihan. Untuk paket yang Anda konfigurasikan agar memiliki active/passive pendekatan, Anda membuat salah satu alur kerja yang dapat digunakan untuk mengaktifkan salah satu Wilayah Anda, atau alur kerja aktivasi terpisah, satu untuk setiap Wilayah. Untuk paket yang Anda konfigurasikan untuk suatu active/active pendekatan, Anda membuat satu alur kerja untuk mengaktifkan Wilayah dan satu alur kerja untuk menonaktifkan Wilayah Anda.

Blok eksekusi

Anda menambahkan blok eksekusi sakelar Wilayah ke alur kerja rencana sakelar Wilayah Anda. Blok eksekusi memungkinkan Anda menentukan pemulihan untuk beberapa aplikasi atau sumber daya ke Wilayah pengaktifan. Saat Anda menambahkan blok eksekusi ke alur kerja, Anda dapat menambahkannya secara berurutan dengan blok lain, atau secara paralel dengan satu atau beberapa blok lainnya.

Konfigurasi anggun dan tidak sopan

Anda dapat memilih untuk menjalankan blok eksekusi tertentu dengan eksekusi yang anggun (direncanakan) atau tidak pantas (tidak direncanakan). Ketika lingkungan Anda sehat, Anda dapat menggunakan alur kerja yang anggun untuk menjalankan semua langkah untuk pelaksanaan rencana yang teratur. Mode alur kerja yang tidak sopan hanya menggunakan langkah dan tindakan yang diperlukan. Saat Anda menjalankan rencana dalam mode tidak pantas, rencana tersebut mengubah perilaku blok eksekusi dalam alur kerja atau melewatkan blok eksekusi tertentu, tergantung pada jenis blok eksekusi.

Jenis blok eksekusi tertentu memiliki perilaku yang berbeda ketika mereka berjalan dengan tidak sopan. Rincian tentang perbedaan ini dijelaskan di bagian yang mencakup rincian tentang setiap jenis blok eksekusi. Untuk informasi selengkapnya, lihat Tambahkan blok eksekusi.

Konfigurasi-konfigurasi Active/active and active/passive

Ada dua pendekatan utama untuk membuat konfigurasi tangguh untuk aplikasi di beberapa Wilayah: active/passive dan aktif/aktif. Sakelar wilayah mendukung pemulihan aplikasi untuk kedua pendekatan ini.

Dengan active/passive konfigurasi, Anda menerapkan dua replika aplikasi Anda di dua Wilayah berbeda, dengan lalu lintas pelanggan hanya menuju satu Wilayah.

Dengan active/active konfigurasi, Anda menyebarkan dua replika ke dua Wilayah yang berbeda, tetapi kedua replika memproses pekerjaan atau menerima lalu lintas.

Rencana eksekusi

Saat paket peralihan Wilayah dijalankan, paket ini mengimplementasikan pemulihan untuk aplikasi saat Wilayah mengalami gangguan dengan mengaktifkan Wilayah yang sehat untuk aplikasi dan lalu lintas yang diterimanya. Dengan active/active konfigurasi, Anda juga menjalankan eksekusi rencana untuk menonaktifkan Wilayah yang rusak.

Alarm kesehatan aplikasi

Alarm kesehatan aplikasi adalah CloudWatch alarm yang Anda tentukan untuk rencana untuk menunjukkan kesehatan aplikasi Anda di setiap Wilayah. Sakelar wilayah menggunakan alarm kesehatan aplikasi untuk membantu menentukan waktu pemulihan aktual setelah Anda beralih Wilayah untuk menerapkan pemulihan.

Pemicu

Anda dapat menggunakan pemicu di sakelar Wilayah untuk mengotomatiskan pemulihan aplikasi. Saat membuat pemicu, Anda menentukan satu atau beberapa CloudWatch alarm Amazon yang menunjukkan kesehatan aplikasi Anda. Ketika alarm masuk ke status alarm, sakelar Wilayah secara otomatis mengeksekusi rencana pemulihan yang sesuai.

Dasbor

Sakelar wilayah mencakup dasbor tempat Anda dapat melacak detail tentang eksekusi rencana secara real time.

Bidang data dan kontrol untuk sakelar Wilayah

Saat Anda merencanakan kegagalan dan pemulihan bencana, pertimbangkan seberapa tangguh mekanisme failover Anda. Kami menyarankan Anda memastikan bahwa mekanisme yang Anda andalkan selama failover sangat tersedia, sehingga Anda dapat menggunakannya saat Anda membutuhkannya dalam skenario bencana. Biasanya, Anda harus menggunakan fungsi bidang data untuk mekanisme Anda kapan pun Anda bisa, untuk keandalan dan toleransi kesalahan terbesar. Dengan mengingat hal itu, penting untuk memahami bagaimana fungsionalitas layanan dibagi antara pesawat kontrol dan pesawat data, dan kapan Anda dapat mengandalkan ekspektasi keandalan ekstrim dengan bidang data layanan.

Seperti banyak AWS layanan lainnya, fungsionalitas untuk kemampuan sakelar Wilayah didukung oleh bidang kontrol dan pesawat data. Sementara kedua jenis dibangun agar dapat diandalkan, bidang kontrol dioptimalkan untuk konsistensi data, sementara bidang data dioptimalkan untuk ketersediaan. Pesawat data dirancang untuk ketahanan sehingga dapat mempertahankan ketersediaan bahkan selama peristiwa yang mengganggu, ketika pesawat kontrol mungkin menjadi tidak tersedia.

Secara umum, bidang kontrol memungkinkan Anda melakukan fungsi manajemen dasar, seperti membuat, memperbarui, dan menghapus sumber daya dalam layanan. Pesawat data menyediakan fungsionalitas inti layanan. Karena itu, kami menyarankan Anda menggunakan operasi pesawat data ketika ketersediaan penting, misalnya, ketika Anda perlu mendapatkan informasi tentang rencana peralihan Wilayah selama pemadaman.

Untuk sakelar Wilayah, bidang kontrol dan bidang data dibagi sebagai berikut:

- Pesawat kontrol untuk sakelar Wilayah terletak di Wilayah AS Timur (Virginia N.) (us-east-1) dan dimaksudkan untuk hanya digunakan untuk manajemen layanan, yaitu, membuat dan

memperbarui rencana, bukan untuk pemulihan, yaitu, melaksanakan rencana. Operasi API bidang kontrol konfigurasi sakelar Wilayah sangat tidak tersedia.

- Sakelar wilayah memiliki bidang data independen di masing-masing Wilayah AWS. Anda harus menggunakan bidang data untuk tindakan pemulihan, yaitu, untuk menjalankan rencana peralihan Wilayah. Untuk daftar operasi paket data, lihat Operasi API sakelar wilayah. Operasi bidang data sakelar Wilayah ini sangat tersedia.

Sakelar wilayah menyediakan konsol independen di masing-masing Wilayah AWS, yang memanggil operasi API bidang data untuk tugas pemulihan, sehingga Anda dapat menggunakan konsol di Wilayah yang Anda aktifkan untuk menjalankan rencana pemulihan aplikasi. Untuk informasi selengkapnya tentang pertimbangan utama saat Anda mempersiapkan dan menyelesaikan operasi pemulihan dengan sakelar Wilayah, lihatPraktik terbaik untuk peralihan Wilayah di ARC.

Untuk informasi selengkapnya tentang bidang data, pesawat kontrol, dan cara AWS membangun layanan untuk memenuhi target ketersediaan tinggi, lihat paper Stabilitas statis menggunakan Availability Zones di Amazon Builders' Library.

Penandaan untuk sakelar Wilayah ARC;

Tag adalah kata atau frasa (meta data) yang Anda gunakan untuk mengidentifikasi dan mengatur AWS sumber daya Anda. Anda dapat menambahkan beberapa tag ke setiap sumber daya, dan setiap tag mencakup kunci dan nilai yang Anda tentukan. Misalnya, kuncinya mungkin lingkungan dan nilainya mungkin produksi. Anda dapat mencari dan memfilter sumber daya Anda berdasarkan tanda yang Anda tambahkan.

Anda dapat menandai sumber daya berikut di sakelar Wilayah di ARC:

- Rencana

Penandaan di ARC hanya tersedia melalui API, misalnya, dengan menggunakan file. AWS CLI

Berikut ini adalah contoh penandaan di sakelar Wilayah dengan menggunakan. AWS CLI

```
aws arc-region-switch --region us-east-1 create-plan --plan-name example-plan --tags Region=IAD,Stage=Prod
```

Untuk informasi selengkapnya, lihat TagResourcedi Panduan Referensi API Region Switch untuk Amazon Application Recovery Controller (ARC).

Harga

Anda membayar biaya bulanan tetap per paket peralihan Wilayah yang Anda konfigurasikan.

Untuk informasi harga terperinci untuk ARC dan contoh harga, lihat Harga ARC.

Praktik terbaik untuk peralihan Wilayah di ARC

Kami merekomendasikan praktik terbaik berikut untuk pemulihan dan kesiapan failover dengan sakelar Wilayah di Amazon Application Recovery Controller (ARC).

Topik

- Jaga agar AWS kredensil yang dibangun khusus dan berumur panjang tetap aman dan selalu dapat diakses
- Pilih nilai TTL yang lebih rendah untuk catatan DNS yang terlibat dalam failover
- Cadangan kapasitas yang diperlukan untuk aplikasi penting
- Gunakan operasi API bidang data yang sangat andal untuk membuat daftar dan mendapatkan informasi tentang rencana peralihan Wilayah
- Uji failover dengan ARC

Jaga agar AWS kredensil yang dibangun khusus dan berumur panjang tetap aman dan selalu dapat diakses

> Dalam skenario pemulihan bencana (DR), pertahankan ketergantungan sistem seminimal mungkin dengan menggunakan pendekatan sederhana untuk mengakses AWS dan melakukan tugas pemulihan. Buat kredensil IAM yang berumur panjang khusus untuk tugas DR, dan simpan kredensialnya dengan aman di brankas fisik lokal atau brankas virtual, untuk diakses bila diperlukan. Dengan IAM, Anda dapat mengelola kredenal keamanan secara terpusat, seperti kunci akses, dan izin untuk akses ke sumber daya. AWS Untuk tugas non-DR, kami menyarankan Anda untuk terus menggunakan akses federasi, menggunakan AWS layanan seperti AWS Single Sign-On.

Pilih nilai TTL yang lebih rendah untuk catatan DNS yang terlibat dalam failover

> Untuk catatan DNS yang mungkin perlu Anda ubah sebagai bagian dari mekanisme failover Anda, terutama catatan yang diperiksa kesehatan, menggunakan nilai TTL yang lebih rendah adalah tepat. Mengatur TTL 60 atau 120 detik adalah pilihan umum untuk skenario ini.

Pengaturan DNS TTL (time to live) memberi tahu resolver DNS berapa lama untuk menyimpan rekaman sebelum meminta yang baru. Ketika Anda memilih TTL, Anda membuat trade-off antara latensi dan keandalan, dan responsif terhadap perubahan. Dengan TTL yang lebih pendek pada catatan, penyelesai DNS melihat pembaruan ke catatan lebih cepat karena TTL menentukan bahwa mereka harus melakukan kueri lebih sering.

Untuk informasi selengkapnya, lihat Memilih nilai TTL untuk catatan DNS dalam Praktik terbaik untuk Amazon Route 53 DNS.

Cadangan kapasitas yang diperlukan untuk aplikasi penting

Sakelar wilayah mencakup jenis blok eksekusi yang membantu menskalakan sumber daya komputasi sebagai bagian dari pemulihan. Jika Anda menggunakan blok eksekusi ini dalam rencana, sakelar Wilayah tidak menjamin bahwa kapasitas komputasi yang diinginkan tercapai. Jika Anda memiliki aplikasi penting dan perlu menjamin akses ke kapasitas, kami sarankan Anda memesan kapasitas.

Ada strategi yang dapat Anda ikuti untuk memesan kapasitas komputasi di Wilayah sekunder sementara juga membatasi biaya. Untuk mempelajari lebih lanjut, lihat Lampu pilot dengan kapasitas cadangan: Cara mengoptimalkan biaya DR menggunakan Pemesanan Kapasitas Sesuai Permintaan.

Gunakan operasi API bidang data yang sangat andal untuk membuat daftar dan mendapatkan informasi tentang rencana peralihan Wilayah

Gunakan operasi API bidang data untuk bekerja dengan dan menjalankan rencana peralihan Wilayah Anda selama acara berlangsung. Untuk daftar operasi bidang data sakelar Wilayah, lihat Operasi API sakelar wilayah.

Konsol sakelar Wilayah di setiap Wilayah menggunakan operasi bidang data untuk menjalankan rencana peralihan Wilayah. Anda juga dapat memanggil operasi API bidang data dengan menggunakan AWS CLI atau dengan menjalankan kode yang Anda tulis menggunakan salah satu AWS SDKs. ARC menawarkan keandalan ekstrim dengan API di bidang data.

Uji pemulihan aplikasi dengan ARC

Uji pemulihan aplikasi secara teratur dengan sakelar Wilayah ARC, untuk mengaktifkan tumpukan aplikasi sekunder di tempat lain Wilayah AWS, atau untuk mengalihkan konfigurasi aktif-aktif dengan menjalankan rencana sakelar Wilayah untuk menonaktifkan salah satu Wilayah.

Penting untuk memastikan bahwa paket peralihan Wilayah yang telah Anda buat selaras dengan sumber daya yang benar di tumpukan Anda, dan semuanya berfungsi seperti yang Anda

harapkan. Anda harus menguji ini setelah Anda mengatur sakelar Wilayah untuk lingkungan Anda, dan terus menguji secara berkala, sehingga Anda memvalidasi bahwa proses pemulihan Anda bekerja dengan benar. Lakukan pengujian ini secara teratur, sebelum Anda mengalami situasi kegagalan, untuk membantu menghindari downtime bagi pengguna Anda.

Tutorial: Buat rencana peralihan active/passive Wilayah

Tutorial ini memandu Anda melalui pembuatan rencana peralihan active/passive Wilayah untuk aplikasi yang berjalan di us-east-1 dan memulihkan ke us-west-2. Contohnya termasuk EC2 instans Amazon untuk komputasi, Amazon Aurora Global Database untuk penyimpanan, dan Amazon Route 53 untuk DNS.

Dalam tutorial ini, Anda akan menyelesaikan langkah-langkah berikut:

- Buat paket sakelar Wilayah
- Membangun alur kerja rencana dan blok eksekusi
- Membangun blok eksekusi grup EC2 Auto Scaling
- Bangun dua blok eksekusi persetujuan manual
- Bangun dua blok eksekusi Lambda tindakan khusus
- Membangun blok eksekusi Amazon Aurora Global Database
- Membangun blok kontrol perutean ARC
- Jalankan rencana sakelar Wilayah

Prasyarat

Sebelum Anda memulai tutorial ini, verifikasi bahwa Anda memiliki prasyarat berikut di kedua Wilayah:

- Peran IAM dengan izin yang sesuai
- EC2 Grup Auto Scaling
- Fungsi Lambda untuk halaman pemeliharaan dan pagar
- Basis Data Global Aurora
- Kontrol perutean ARC

Langkah 1: Buat paket sakelar Wilayah

1. Dari konsol switch Region, pilih Create Region switch plan.
2. Berikan detail berikut:
   - Wilayah Utama: Pilih us-east-1
   - Wilayah Siaga: Pilih us-west-2
   - Tujuan waktu pemulihan yang diinginkan (RTO) (opsional)
   - Peran IAM: Masukkan peran IAM eksekusi rencana. Peran IAM ini memungkinkan Region beralih ke AWS layanan panggilan selama eksekusi.
3. Pilih Buat.

(Opsional) Tambahkan sumber daya dari AWS akun yang berbeda ke paket peralihan Wilayah Anda:

1. Buat peran lintas akun:
   - Di akun yang menghosting sumber daya, buat peran IAM.
   - Tambahkan izin untuk sumber daya tertentu yang akan diakses paket.
   - Tambahkan kebijakan kepercayaan yang memungkinkan peran eksekusi untuk mengambil peran baru.
   - Masukkan dan catat ID eksternal yang akan Anda gunakan sebagai rahasia bersama.
2. Konfigurasikan sumber daya dalam paket Anda:
   - Saat Anda menambahkan sumber daya ke paket Anda, tentukan dua bidang tambahan:
     - crossAccountRole: ARN dari peran yang Anda buat di langkah 1
     - ExternalId: ID eksternal yang Anda masukkan pada langkah 1

Contoh konfigurasi untuk blok eksekusi EC2 Auto Scaling yang mengakses sumber daya di akun 987654321:

```
{
  "executionBlock": "EC2AutoScaling",
  "name": "ASG",
  "crossAccountRole": "arn:aws:iam::987654321:role/RegionSwitchCrossAccountRole",
  "externalId": "unique-external-id-123",
```

```
  "autoScalingGroupArn": "arn:aws:autoscaling:us-
west-2:987654321:autoScalingGroup:*:autoScalingGroupName/CrossAccountASG"
}
```

Izin yang diperlukan:

- Peran eksekusi harus memiliki sts: AssumeRole izin untuk peran lintas akun.
- Peran lintas akun harus memiliki izin hanya untuk sumber daya tertentu yang diakses.
- Kebijakan kepercayaan peran lintas akun harus mencakup:
  - Akun peran eksekusi sebagai entitas tepercaya.
  - Kondisi ID eksternal.

Sebelum menjalankan rencana, sakelar Wilayah akan memverifikasi hal berikut:

- Peran eksekusi dapat mengasumsikan peran lintas akun.
- Peran lintas akun memiliki izin yang diperlukan.
- ID eksternal cocok dengan kebijakan kepercayaan.

Langkah 2: Bangun alur kerja rencana dan blok eksekusi

1. Dari halaman detail rencana peralihan Wilayah, pilih Buat alur kerja.
2. Pilih Buat alur kerja aktivasi yang sama untuk semua Wilayah.
3. Masukkan deskripsi alur kerja aktivasi Wilayah (opsional). Ini akan digunakan untuk dengan mudah mengidentifikasi alur kerja saat menjalankan rencana.
4. Jangan pilih Save and continue (Simpan dan lanjutkan).
5. Pilih Tambahkan langkah, lalu pilih Jalankan secara berurutan.
6. Pilih blok eksekusi EC2 Auto Scaling, lalu pilih Tambah dan edit. Blok ini akan memungkinkan Anda untuk mulai meningkatkan kapasitas di Wilayah pasif.
7. Di panel kanan, konfigurasikan blok:
   - Nama langkah: Masukkan “Skala”
   - Deskripsi langkah (opsional)
   - Grup Auto Scaling ARN untuk us-east-1: ARN ASG Anda di us-east-1
   - Grup Auto Scaling ARN untuk us-west-2: ARN ASG Anda di us-west-2

- Persen untuk mencocokkan sumber Kapasitas Wilayah: Masukkan 100
- Pendekatan pemantauan kapasitas: Tinggalkan sebagai “Terbaru”
- Batas waktu (opsional)

8. Pilih Simpan langkah.
9. Pilih Tambahkan langkah.
10. Pilih blok eksekusi persetujuan manual dan tambahkan ke jendela desain. Blok ini memungkinkan verifikasi manusia sebelum melanjutkan.
11. Di panel kanan, konfigurasikan blok:

    - Nama langkah: Masukkan “Persetujuan manual sebelum pengaturan”
    - Deskripsi langkah (opsional)
    - Peran persetujuan IAM: Peran yang harus diambil pengguna untuk menyetujui eksekusi
    - Batas waktu (opsional). Setelah batas waktu, eksekusi berhenti dan Anda dapat memilih untuk mencoba lagi, melewati, atau membatalkan.

12. Pilih Simpan langkah.
13. Pilih Tambahkan langkah.
14. Pilih blok eksekusi Lambda tindakan kustom, lalu pilih Tambah dan edit. Blok ini menerbitkan halaman pemeliharaan di Wilayah yang sedang aktif.
15. Di panel kanan, konfigurasikan blok:

    - Nama langkah: Masukkan “Tampilan halaman pemeliharaan”
    - Deskripsi langkah (opsional)
    - Lambda ARN untuk mengaktifkan us-east-1: ARN dari halaman pemeliharaan Fungsi Lambda diterapkan di us-east-1
    - Lambda ARN untuk mengaktifkan us-west-2: ARN dari halaman pemeliharaan Fungsi Lambda diterapkan di us-west-2
    - Wilayah untuk menjalankan fungsi Lambda: Pilih Jalankan dalam mengaktifkan Wilayah
    - Batas waktu (opsional)
    - Coba lagi interval (opsional)

16. Pilih Simpan langkah.
17. Pilih Tambahkan langkah.

18. Pilih blok eksekusi Lambda tindakan kustom kedua, lalu pilih Tambah dan edit. Blok ini memicu mekanisme pagar di Wilayah aktif yang memastikan bahwa Wilayah yang menonaktifkan tidak dapat lagi menerima lalu lintas.
19. Di panel kanan, konfigurasikan blok:
    - Nama langkah: Masukkan “Anggar”
    - Deskripsi langkah (opsional)
    - Lambda ARN untuk mengaktifkan us-east-1: ARN dari fungsi pagar Lambda dikerahkan di us-east-1
    - Lambda ARN untuk mengaktifkan us-west-2: ARN dari fungsi pagar Lambda digunakan di us-west-2
    - Wilayah untuk menjalankan fungsi Lambda: Pilih Jalankan di menonaktifkan Wilayah
    - Batas waktu (opsional)
    - Coba lagi interval (opsional)
20. Pilih Simpan langkah.
21. Pilih Tambahkan langkah.
22. Pilih Blok eksekusi persetujuan manual, lalu pilih Tambah dan edit. Blok ini meminta persetujuan dari anggota tim.
23. Di panel kanan, konfigurasikan blok:
    - Nama langkah: Masukkan persetujuan Manual sebelum Database dan DNS berubah
    - Deskripsi langkah (opsional)
    - Peran persetujuan IAM: Peran yang harus diasumsikan pengguna sehingga mereka dapat menyetujui eksekusi
    - Batas waktu (opsional)
24. Pilih Simpan langkah.
25. Pilih Tambahkan langkah.
26. Pilih blok eksekusi Aurora Global Database, lalu pilih Tambah dan edit. Blok ini memicu peralihan basis data global Aurora (tidak ada kehilangan data). Untuk informasi selengkapnya, lihat Menggunakan switchover atau failover untuk Aurora Global Database di Panduan Pengguna Aurora.
27. Di panel kanan, konfigurasikan blok:
    - Nama langkah: Masukkan peralihan Aurora

- Deskripsi langkah (opsional)
- Pengidentifikasi basis data global Aurora: Nama cluster Aurora
- Cluster ARN digunakan untuk mengaktifkan us-east-1: ARN cluster Aurora di us-east-1
- Cluster ARN digunakan untuk mengaktifkan us-west-2: ARN cluster Aurora di us-west-2
- Pilih opsi untuk database Aurora: Pilih Switchover
- Batas waktu (opsional)

28. Pilih Simpan langkah.
29. Pilih Tambahkan langkah.
30. Pilih blok eksekusi kontrol perutean ARC, lalu pilih Tambah dan edit. Blok ini melakukan failover DNS untuk mengalihkan lalu lintas ke Wilayah pasif.
31. Di panel kanan, konfigurasikan blok:

    - Nama langkah: Masukkan Toggle DNS
    - Deskripsi langkah (opsional)
    - Kontrol perutean yang digunakan dalam mengaktifkan us-east-1: Pilih Tambahkan kontrol perutean
    - Timeout: Masukkan nilai batas waktu.

32. Pilih Tambahkan kontrol perutean:

    - Kontrol perutean ARN: ARN dari kontrol perutean yang mengontrol us-east-1
    - Status kontrol perutean: Pilih Aktif

33. Pilih Tambahkan kontrol perutean lagi:

    - Kontrol perutean ARN: ARN dari kontrol perutean yang mengontrol us-west-2
    - Status kontrol perutean: Pilih Nonaktif

34. Pilih Simpan.
35. Kontrol perutean yang digunakan dalam mengaktifkan us-west-2: Pilih Tambahkan kontrol perutean
36. Pilih Tambahkan kontrol perutean:

    - Kontrol perutean ARN: ARN dari kontrol perutean yang mengontrol us-west-2
    - Status kontrol perutean: Pilih Aktif

37. Pilih Tambahkan kontrol perutean lagi:

- Kontrol perutean ARN: ARN dari kontrol perutean yang mengontrol us-east-1
- Status kontrol perutean: Pilih Nonaktif

38. Pilih Simpan.
39. Pilih Simpan langkah.
40. Pilih Simpan.

Langkah 3: Jalankan rencana

1. Pada halaman Detail rencana peralihan Wilayah, di kanan atas, pilih Jalankan.
2. Masukkan detail eksekusi:
   - Pilih Wilayah yang akan diaktifkan.
   - Pilih mode eksekusi rencana.
   - (Opsional) Lihat langkah-langkah eksekusi.
   - Akui pelaksanaan rencana.
3. Pilih Mulai.
4. Anda dapat melihat langkah-langkah terperinci saat rencana dijalankan di halaman detail eksekusi. Anda dapat melihat setiap langkah dalam eksekusi rencana, termasuk waktu mulai, waktu akhir, ARN sumber daya, dan pesan log.

Ketika Region yang rusak telah pulih, Anda dapat menjalankan rencana lagi (mengubah parameter yang Anda berikan) untuk mengaktifkan Region asli, untuk mengalihkan kembali operasi aplikasi Anda ke Region primer asli.

Operasi API sakelar wilayah

Tabel berikut mencantumkan operasi ARC yang dapat Anda gunakan untuk sakelar Wilayah, dengan tautan ke dokumentasi yang relevan.

Tindakan	Menggunakan konsol ARC	Menggunakan ARC API	API bidang data
Menyetujui atau menolak langkah eksekusi rencana	Lihat Blok eksekusi persetujuan manual	Lihat ApprovePlanExecutionStep	Ya
Batalkan eksekusi rencana	Lihat Buat paket sakelar Wilayah	Lihat CancelPlanExecution	Ya
Buat rencana	Lihat Buat paket sakelar Wilayah	Lihat CreatePlan	Tidak
Hapus paket	Lihat Bekerja dengan sakelar Wilayah	Lihat DeletePlan	Tidak
Dapatkan rencana	Lihat Bekerja dengan sakelar Wilayah	Lihat GetPlan	Tidak
Dapatkan status evaluasi rencana	Lihat Evaluasi rencana	Lihat GetPlanEvaluationStatus	Ya
Dapatkan eksekusi rencana	Lihat Dasbor sakelar wilayah	Lihat GetPlanExecution	Ya
Dapatkan rencana di Region	Lihat Bekerja dengan sakelar Wilayah	Lihat GetPlanInRegion	Ya
Buat daftar pemeriksaan kesehatan untuk sebuah rencana	Lihat Blok eksekusi pemeriksaan kesehatan Amazon Route 53	Lihat ListHealthChecksForPlan	Tidak
Daftar acara eksekusi rencana	Lihat Jalankan rencana peralihan Wilayah untuk memulihkan aplikasi	Lihat ListPlanExecutionEvents	Ya

Tindakan	Menggunakan konsol ARC	Menggunakan ARC API	API bidang data
Daftar eksekusi rencana	Lihat Jalankan rencana peralihan Wilayah untuk memulihkan aplikasi	Lihat ListPlanExecutions	Ya
Daftar rencana	Lihat Bekerja dengan sakelar Wilayah	Lihat ListPlans	Tidak
Daftar rencana di Wilayah	Lihat Bekerja dengan sakelar Wilayah	Lihat ListPlansInRegion	Ya
Membuat daftar tanda untuk sumber daya	Lihat Penandaan untuk sakelar Wilayah ARC;	Lihat ListTagsForResource	Tidak
Mulai eksekusi rencana	Lihat Jalankan rencana peralihan Wilayah untuk memulihkan aplikasi	Lihat StartPlanExecution	Ya
Menandai sumber daya	Lihat Buat paket sakelar Wilayah	Lihat TagResource	Tidak
Hapus tag dari sumber daya	Lihat Penandaan untuk sakelar Wilayah ARC;	Lihat UntagResource	Tidak
Perbarui rencana	Lihat Buat paket sakelar Wilayah	Lihat UpdatePlan	Tidak
Perbarui eksekusi rencana	Lihat Buat paket sakelar Wilayah	Lihat UpdatePlanExecution	Ya
Perbarui langkah eksekusi rencana	Lihat Buat paket sakelar Wilayah	Lihat UpdatePlanExecutionStep	Ya

Bekerja dengan sakelar Wilayah

Bagian ini memberikan step-by-step petunjuk untuk bekerja dengan rencana peralihan Wilayah, yang dapat Anda gunakan untuk memulihkan aplikasi Multi-wilayah. Peralihan wilayah memungkinkan Anda membuat rencana untuk keduanya active/passive dan pendekatan active/active pemulihan.

Untuk membuat rencana pemulihan untuk aplikasi Anda, Anda melakukan hal berikut:

1. Buat rencana peralihan Wilayah. Rencana adalah struktur dengan atribut tertentu, seperti spesifik Wilayah AWS yang dijalankan aplikasi Anda. Setiap rencana mencakup satu atau lebih alur kerja.

   Secara opsional, Anda dapat membuat beberapa rencana, dan menyusun rencana anak tersebut dalam rencana pemulihan secara keseluruhan.
2. Buat alur kerja untuk rencana tersebut. Anda tidak dapat menjalankan rencana tanpa membuat alur kerja terlebih dahulu.
3. Dalam alur kerja, tambahkan satu atau beberapa langkah yang masing-masing merupakan blok eksekusi.

   Misalnya, Anda dapat menambahkan blok eksekusi untuk meningkatkan grup EC2 Auto Scaling di Wilayah tujuan.
4. Setelah Anda menambahkan blok eksekusi ke alur kerja Anda, langkah-langkah tambahan mungkin diperlukan, seperti mengonfigurasi pemeriksaan kesehatan di Amazon Route 53. Setiap bagian blok eksekusi menyertakan informasi konfigurasi yang Anda butuhkan. Untuk informasi selengkapnya, lihat Tambahkan blok eksekusi.
5. Untuk memulihkan aplikasi Anda ketika sedang berjalan dalam gangguan Wilayah AWS, jalankan paket.

   Anda dapat melacak kemajuan pelaksanaan rencana dengan melihat informasi di dasbor global atau dasbor Regional.

Bagian berikut memberikan informasi rinci dan langkah-langkah untuk membuat rencana dan alur kerja, dan menambahkan langkah-langkah blok eksekusi dalam alur kerja Anda.

Daftar Isi

- Buat rencana anak
- Buat pemicu untuk paket sakelar Wilayah
- Jalankan rencana peralihan Wilayah untuk memulihkan aplikasi

Prosedur di bagian ini menggambarkan cara bekerja dengan rencana, alur kerja, blok eksekusi, dan pemicu dengan menggunakan. AWS Management Console Untuk bekerja dengan operasi API sakelar Wilayah, lihat Operasi API sakelar wilayah.

Buat paket sakelar Wilayah

Anda dapat membuat dua jenis rencana berbeda di sakelar Wilayah: active/active rencana atau active/passive rencana. Saat Anda membuat rencana, tentukan jenis yang berlaku untuk cara Anda ingin mengelola failover.

- Pendekatan aktif/pasif menyebarkan dua replika aplikasi ke dalam dua Wilayah, dengan lalu lintas diarahkan ke Wilayah aktif saja. Anda dapat mengaktifkan replika di Wilayah pasif dengan menjalankan rencana sakelar Wilayah.
- Pendekatan aktif/aktif menyebarkan dua replika aplikasi ke dalam dua Wilayah, dan kedua replika memproses pekerjaan atau menerima lalu lintas.

Untuk membuat rencana peralihan Wilayah

1. Dari konsol sakelar Region, pilih Create Region switch plan with active/passive approach.
2. Berikan detail berikut:
    - Nama paket - Masukkan nama deskriptif untuk paket Anda.
    - Pendekatan Multi-Region - Pilih Aktif/Pasif atau Aktif/Aktif. Pendekatan ini berarti dua replika aplikasi dikerahkan ke dua Wilayah, dengan lalu lintas diarahkan ke Wilayah aktif saja. Anda dapat mengaktifkan replika di Wilayah pasif dengan menjalankan rencana sakelar Wilayah.
        - Pilih aktif/pasif jika Anda telah menerapkan dua replika aplikasi ke dalam dua Wilayah, dengan lalu lintas dirutekan ke Wilayah aktif saja. Kemudian, Anda dapat mengaktifkan replika di Wilayah pasif dengan menjalankan rencana sakelar Wilayah yang menentukan aktif/pasif.
        - Pilih Aktif/aktif jika Anda telah menerapkan dua replika aplikasi ke dalam dua Wilayah, dan kedua replika sedang memproses pekerjaan atau menerima lalu lintas.

- Wilayah atau Wilayah Primer dan siaga - Pilih Wilayah utama dan siaga untuk aplikasi Anda. Untuk active/active penerapan, pilih Wilayah tempat replika digunakan.
- Tujuan waktu pemulihan (RTO) - Masukkan RTO yang Anda inginkan. Sakelar wilayah menggunakan ini untuk memberikan wawasan tentang berapa lama eksekusi rencana peralihan Wilayah harus diselesaikan dibandingkan dengan RTO yang Anda inginkan.
- Peran IAM - Berikan peran IAM untuk sakelar Wilayah untuk digunakan untuk menjalankan rencana. Untuk informasi selengkapnya tentang izin, lihat Pengalihan Identity and Access Management untuk Wilayah di ARC.
- CloudWatch Alarm Amazon - Menyediakan alarm kesehatan aplikasi yang telah Anda buat dengan Amazon CloudWatch, untuk menunjukkan kesehatan aplikasi Anda di setiap Wilayah. Sakelar wilayah menggunakan alarm kesehatan aplikasi ini untuk membantu menentukan waktu pemulihan aktual setelah Anda beralih Wilayah untuk menerapkan pemulihan.
- Tag - Secara opsional, tambahkan satu atau lebih tag ke paket Anda.

Buat alur kerja rencana peralihan Wilayah

Setelah membuat rencana peralihan Wilayah, Anda perlu menentukan dan membuat alur kerja yang menentukan proses pemulihan untuk aplikasi Anda. Untuk setiap paket, Anda menentukan satu atau beberapa alur kerja yang menyelesaikan pemulihan untuk aplikasi Anda. Di setiap alur kerja, Anda menambahkan langkah-langkah yang menyertakan blok eksekusi yang menentukan setiap tindakan yang Anda inginkan untuk beralih Wilayah untuk pemulihan aplikasi Anda.

Jumlah alur kerja yang Anda buat bergantung pada skenario penerapan aplikasi dan preferensi Anda untuk mengelola pemulihan. Misalnya:

- Jika rencana peralihan Wilayah Anda adalah untuk active/active application deployment, you also need to create a deactivation workflow. This means that for or active/active penerapan, Anda akan memiliki minimal dua alur kerja: alur kerja aktivasi dan alur kerja penonaktifan.
- Jika paket peralihan Wilayah Anda adalah untuk active/passive penerapan aplikasi, Anda memiliki Region primer dan sekunder. Jika Anda memilih untuk memiliki alur kerja aktivasi terpisah untuk setiap Wilayah, Anda akan membuat dua alur kerja: satu untuk setiap Wilayah.

Untuk membuat alur kerja rencana peralihan Wilayah

1. Di daftar sakelar Wilayah yang Anda buat, pilih Buat alur kerja.
2. Pilih salah satu opsi alur kerja berikut:

- Buat alur kerja aktivasi yang sama untuk semua Wilayah - Memungkinkan Anda menggunakan alur kerja aktivasi yang sama di seluruh Wilayah.
- Buat alur kerja secara terpisah untuk setiap Wilayah - Membangun alur kerja aktivasi individual untuk setiap Wilayah.

3. Secara opsional, berikan deskripsi untuk setiap alur kerja.
4. Tentukan alur kerja yang diperlukan untuk memulihkan aplikasi Anda. Dalam alur kerja Anda, Anda menambahkan blok eksekusi untuk menentukan langkah-langkah yang Anda inginkan untuk beralih Wilayah untuk pemulihan Anda. Setiap blok eksekusi mendefinisikan tindakan, seperti pengalihan lalu lintas aplikasi atau pemulihan basis data di Wilayah pengaktifan, dan mendukung sumber daya di wilayah lain. Akun AWS Anda dapat memilih untuk menjalankan blok eksekusi secara paralel atau berurutan. Untuk informasi rinci tentang blok eksekusi tertentu yang dapat Anda tambahkan ke alur kerja, lihatTambahkan blok eksekusi.
5. Bergantung pada opsi alur kerja yang Anda pilih, lakukan hal berikut:

   - Jika Anda memilih Membangun alur kerja aktivasi yang sama untuk semua Wilayah, diperlukan satu alur kerja aktivasi.
   - Jika Anda memilih alur kerja Build secara terpisah untuk setiap Wilayah, diperlukan dua alur kerja aktivasi.

   Untuk active/active rencana, Anda harus menentukan alur kerja aktivasi dan alur kerja penonaktifan.

Tambahkan blok eksekusi

Anda menambahkan blok eksekusi ke alur kerja dalam paket peralihan Wilayah, untuk melakukan langkah-langkah individual guna menyelesaikan failover atau peralihan aplikasi Anda. Untuk detail tentang fungsionalitas dan perilaku setiap jenis blok eksekusi, lihat deskripsi berikut.

Sakelar wilayah menjalankan evaluasi rencana segera setelah Anda membuat rencana atau memperbaruinya, dan kemudian setiap 30 menit selama kondisi tunak. Peralihan wilayah menyimpan informasi tentang evaluasi rencana di semua Wilayah tempat paket Anda dikonfigurasi. Setiap bagian blok eksekusi di sini mencakup informasi tentang apa yang dievaluasi saat sakelar Wilayah menjalankan evaluasi rencana.

Sakelar wilayah mencakup jenis blok eksekusi yang membantu menskalakan sumber daya komputasi sebagai bagian dari pemulihan. Jika Anda menggunakan blok eksekusi ini dalam sebuah

rencana, ketahuilah bahwa sakelar Wilayah tidak menjamin bahwa kapasitas komputasi yang diinginkan akan tercapai. Jika Anda memiliki aplikasi penting dan perlu menjamin akses ke kapasitas, kami sarankan Anda memesan kapasitas. Ada strategi yang dapat Anda ikuti untuk memesan kapasitas komputasi di Wilayah sekunder sementara juga membatasi biaya. Untuk mempelajari lebih lanjut, lihat Lampu pilot dengan kapasitas cadangan: Cara mengoptimalkan biaya DR menggunakan Pemesanan Kapasitas Sesuai Permintaan.

Sakelar wilayah mendukung blok eksekusi berikut.

Blok eksekusi	Fungsi	Konfigurasi yang tidak menyenang kan
Blok eksekusi rencana sakelar Wilayah ARC	Mengatur pemulihan untuk beberapa aplikasi dalam satu eksekusi dengan menentukan rencana anak untuk dieksekusi.	Mulai rencana anak dengan konfigurasi mereka yang tidak menyenangkan.
Blok eksekusi grup EC2 Auto Scaling Amazon	EC2 Skalakan sumber daya komputasi yang ada dalam grup Auto Scaling sebagai bagian dari eksekusi paket Anda.	Tentukan persentase minimum kapasitas komputasi yang harus dicocokkan di Wilayah yang Anda aktifkan.
Blok eksekusi penskalaan sumber daya Amazon EKS	Skala pod klaster Amazon EKS sebagai bagian dari eksekusi paket Anda.	N/A
Blok eksekusi penskalaan layanan Amazon ECS	Skalakan tugas layanan Amazon ECS sebagai bagian dari eksekusi paket Anda.	N/A
Blok eksekusi kontrol perutean ARC	Tambahkan langkah untuk mengubah status satu atau beberapa kontrol perutean ARC, untuk mengarahkan lalu lintas aplikasi Anda ke target. Wilayah AWS	N/A

Blok eksekusi	Fungsi	Konfigurasi yang tidak menyenang kan
Blok eksekusi Database Global Amazon Aurora	Lakukan alur kerja pemulihan untuk database global Aurora.	Lakukan failover database global Aurora (berpotensi menyebabkan kehilangan data).
Blok eksekusi persetujuan manual	Masukkan langkah persetujuan, untuk meminta persetujuan atau pembatalan eksekusi sebelum melanjutkan.	N/A
Tindakan kustom Blok eksekusi Lambda	Tambahkan langkah khusus untuk menjalankan fungsi Lambda, untuk mengaktifkan tindakan kustom.	Lewati langkahnya.
Blok eksekusi pemeriksa an kesehatan Amazon Route 53	Menentukan Wilayah tempat lalu lintas aplikasi Anda akan diarahkan selama failover.	N/A

Blok eksekusi rencana sakelar Wilayah ARC

Blok eksekusi rencana sakelar Wilayah memungkinkan Anda mengatur urutan di mana beberapa aplikasi beralih ke Wilayah yang ingin Anda aktifkan, dengan mereferensikan rencana peralihan Wilayah turunan lainnya. Dengan menggunakan hubungan orangtua/anak ini, Anda dapat membuat proses pemulihan yang kompleks dan terkoordinasi yang mengelola beberapa sumber daya dan dependensi di seluruh infrastruktur Anda.

Konfigurasi

Saat Anda menggunakan blok eksekusi rencana sakelar Wilayah, Anda memilih paket sakelar Wilayah tertentu yang ingin dieksekusi dalam alur kerja paket yang Anda buat.

Untuk mengonfigurasi blok eksekusi rencana sakelar Wilayah, masukkan nilai berikut:

1. Nama langkah: Masukkan nama.
2. Deskripsi langkah (opsional): Masukkan deskripsi langkah.

3. Rencana peralihan wilayah: Pilih rencana untuk dijalankan dalam alur kerja untuk rencana saat ini.

Kemudian, pilih Save step.

Cara kerjanya

Gunakan blok eksekusi rencana sakelar Wilayah untuk membuat alur kerja bersarang dengan parent/child relasi. Perhatikan bahwa blok eksekusi ini tidak mendukung level tambahan paket anak, dan membatasi jumlah paket anak bersarang. Rencana anak harus mendukung Wilayah yang sama yang didukung oleh rencana induk, dan harus memiliki pendekatan pemulihan yang sama dengan rencana induk (yaitu, active/active atau aktif/pasif).

Blok ini mendukung mode eksekusi yang anggun dan tidak menyenangkan. Pengaturan yang tidak menyenangkan akan memulai rencana anak dengan konfigurasi tidak senonoh mereka. Jika blok sakelar Wilayah dijalankan dengan anggun, dan kemudian beralih ke mode eksekusi yang tidak beraturan, paket anak apa pun juga akan beralih ke mode eksekusi yang tidak pantas.

Apa yang dievaluasi sebagai bagian dari evaluasi rencana

Jika Anda membagikan paket di seluruh akun, dan paket tidak lagi dibagikan dengan akun paket induk, evaluasi peralihan Wilayah akan menampilkan peringatan bahwa paket tersebut tidak valid.

Blok eksekusi grup EC2 Auto Scaling Amazon

Blok eksekusi grup EC2 Auto Scaling memungkinkan Anda menskalakan EC2 instans sebagai bagian dari proses pemulihan Multi-wilayah Anda. Anda dapat menentukan persentase kapasitas, relatif terhadap Wilayah yang Anda tinggalkan (sumber dan tujuan).

Konfigurasi

Saat Anda mengonfigurasi blok eksekusi grup EC2 Auto Scaling, Anda memasukkan Auto EC2 ARNs Scaling untuk Wilayah tertentu yang terkait dengan paket Anda. Anda harus memasukkan EC2 Auto Scaling ARNs di setiap Wilayah yang ingin ditingkatkan selama eksekusi rencana.

Untuk mengonfigurasi blok eksekusi grup EC2 Auto Scaling, masukkan nilai berikut:

1. Nama langkah: Masukkan nama.
2. Deskripsi langkah (opsional): Masukkan deskripsi langkah.
3. EC2 Grup Auto Scaling ARN untuk Wilayah: Masukkan ARN untuk Auto EC2 Scaling di setiap Wilayah untuk paket Anda.

4. Persentase yang sesuai dengan kapasitas Wilayah yang diaktifkan: Masukkan persentase yang diinginkan dari jumlah instans yang berjalan di grup Auto Scaling agar sesuai dengan Wilayah yang diaktifkan.
5. Pendekatan pemantauan kapasitas: Di menu tarik-turun, pilih pendekatan pemantauan Anda untuk grup EC2 Auto Scaling Anda.
6. Timeout: Masukkan nilai batas waktu.

Kemudian, pilih Save step.

Cara kerjanya

Setelah Anda mengonfigurasi blok eksekusi EC2 Auto Scaling, sakelar Wilayah mengonfirmasi bahwa hanya ada satu grup Auto Scaling sumber dan satu grup Auto Scaling tujuan. Jika ada beberapa grup Auto Scaling, blok eksekusi gagal selama evaluasi rencana. Kapasitas target didefinisikan sebagai jumlah instance memiliki status yang diatur ke`InService`. Untuk informasi selengkapnya, lihat Siklus hidup instance EC2 Auto Scaling.

Berdasarkan nilai yang Anda tentukan (saat Anda mengonfigurasi blok eksekusi Auto Scaling) untuk persentase yang cocok, sakelar Wilayah menghitung kapasitas baru yang diinginkan untuk grup Auto Scaling tujuan. Kapasitas baru yang diinginkan dibandingkan dengan kapasitas yang diinginkan grup Auto Scaling tujuan. Rumus yang digunakan sakelar Region untuk menghitung kapasitas yang diinginkan adalah sebagai berikut:`ceil(percentToMatch * Source Auto Scaling group capacity)`, di mana ceil () adalah fungsi yang membulatkan hasil fraksional apa pun. Jika kapasitas yang diinginkan saat ini dari grup Auto Scaling tujuan lebih besar dari atau sama dengan kapasitas yang diinginkan dari grup Auto Scaling baru yang dihitung oleh sakelar Wilayah, blok eksekusi akan dilanjutkan. Perhatikan bahwa sakelar Wilayah tidak menurunkan kapasitas grup Auto Scaling.

Saat sakelar Wilayah mengeksekusi blok Auto Scaling, sakelar Wilayah mencoba meningkatkan kapasitas grup Auto Scaling Wilayah target agar sesuai dengan kapasitas yang diinginkan. Kemudian, sakelar Wilayah menunggu hingga kapasitas grup Auto Scaling yang diminta terpenuhi di grup Auto Scaling Wilayah target sebelum peralihan Wilayah melanjutkan ke langkah berikutnya dalam rencana.

Jika Anda menggunakan active/active pendekatan, sakelar Wilayah menggunakan Wilayah lain yang dikonfigurasi sebagai sumbernya. Artinya, jika Wilayah dinonaktifkan, sakelar Wilayah menggunakan Wilayah aktif lainnya sebagai sumber untuk mencocokkan persentase untuk skala.

Blok ini mendukung mode eksekusi yang anggun dan tidak menyenangkan. Anda dapat mengonfigurasi eksekusi yang tidak pantas dengan menentukan persentase minimum kapasitas

komputasi yang akan dicocokkan di Wilayah target sebelum peralihan Wilayah melanjutkan ke langkah berikutnya dalam rencana.

Apa yang dievaluasi sebagai bagian dari evaluasi rencana

Saat sakelar Wilayah mengevaluasi paket Anda, sakelar Wilayah melakukan beberapa pemeriksaan penting pada konfigurasi dan izin blok eksekusi grup EC2 Auto Scaling Anda. Evaluasi sakelar wilayah memverifikasi bahwa grup Auto Scaling ada di kedua Wilayah, memastikan bahwa grup tersebut dikonfigurasi dan diakses dengan benar, serta mencatat jumlah instans yang berjalan di setiap Wilayah. Ini juga menegaskan bahwa kapasitas maksimum dalam kelompok Auto Scaling Wilayah target cukup untuk menangani kecocokan persentase skala yang ditentukan untuk kapasitas yang diperlukan.

Sakelar wilayah juga memvalidasi bahwa peran IAM paket memiliki izin yang benar untuk Auto Scaling. Untuk informasi selengkapnya tentang izin yang diperlukan untuk blok eksekusi peralihan Wilayah, lihatContoh kebijakan berbasis identitas untuk peralihan Wilayah di ARC. Jika salah satu pemeriksaan gagal, sakelar Wilayah mengembalikan pesan peringatan, yang dapat Anda lihat di konsol. Atau, Anda dapat menerima peringatan validasi melalui EventBridge atau dengan menggunakan operasi API.

Blok eksekusi penskalaan sumber daya Amazon EKS

Blok eksekusi penskalaan sumber daya EKS memungkinkan Anda menskalakan sumber daya EKS sebagai bagian dari proses pemulihan Multi-wilayah Anda. Ketika Anda mengkonfigurasi blok eksekusi, Anda menentukan persentase kapasitas untuk skala, relatif terhadap kapasitas di Wilayah yang sedang dinonaktifkan.

Konfigurasikan izin entri akses EKS

Sebelum Anda dapat menambahkan blok eksekusi untuk penskalaan sumber daya EKS, Anda harus menyediakan switch Region dengan izin yang diperlukan untuk mengambil tindakan dengan sumber daya Kubernetes di kluster EKS Anda. Untuk menyediakan akses bagi sakelar Wilayah, Anda harus membuat entri akses EKS untuk peran IAM yang digunakan sakelar Region untuk eksekusi paket, dengan menggunakan kebijakan akses sakelar Wilayah berikut: `arn:aws:eks::aws:cluster-access-policy/AmazonARCRegionSwitchScalingPolicy`

Kebijakan akses EKS beralih wilayah

Informasi berikut memberikan rincian tentang kebijakan akses EKS.

Nama: `AmazonARCRegionSwitchScalingPolicy`

Kebijakan ARN: `arn:aws:eks::aws:cluster-access-policy/AmazonARCRegionSwitchScalingPolicy`

Grup API Kubernetes	Sumber daya Kubernetes	Kata kerja Kubernetes (izin)
*	*/skala	dapatkan, perbarui
*	*/status	memperoleh
penskalaan otomatis	horizontalpodautoscalers	dapatkan, tambal

Buat entri akses EKS untuk sakelar Wilayah

Contoh berikut menjelaskan cara membuat entri akses yang diperlukan dan asosiasi kebijakan akses sehingga peralihan Wilayah dapat mengambil tindakan spesifik untuk sumber daya Kubernetes Anda. Dalam contoh ini, izin berlaku untuk namespace *`my-namespace1`* di kluster EKS *`my-cluster`* untuk peran IAM. `arn:aws:iam::555555555555:role/`*`my-role`*

Saat Anda mengonfigurasi izin ini, pastikan Anda mengambil langkah-langkah ini untuk kedua kluster EKS di blok eksekusi Anda.

Prasyarat

Sebelum Anda memulai, ubah mode otentikasi cluster menjadi salah satu `API_AND_CONFIG_MAP` atau`API`. Mengubah mode otorisasi menambahkan API untuk entri akses. Untuk informasi selengkapnya, lihat Mengubah mode autentikasi untuk menggunakan entri akses di Panduan Pengguna Amazon EKS.

Buat entri akses

Langkah pertama adalah membuat entri akses dengan menggunakan AWS CLI perintah yang mirip dengan yang berikut ini:

```
aws eks create-access-entry --cluster-name my-cluster --principal-arn
        arn:aws:iam::555555555555:user/my-user --type STANDARD
```

Untuk informasi selengkapnya, lihat Membuat entri akses di Panduan Pengguna Amazon EKS.

Buat asosiasi entri akses

Selanjutnya, buat asosiasi ke kebijakan akses sakelar Wilayah dengan menggunakan AWS CLI perintah yang mirip dengan berikut ini:

```
aws eks associate-access-policy --cluster-name my-cluster --principal-arn
 arn:aws:iam::555555555555:role/my-role \
          --access-scope type=namespace,namespaces=my-namespace1 --policy-arn
          arn:aws:eks::aws:cluster-access-policy/AmazonARCRegionSwitchScalingPolicy
```

Untuk informasi selengkapnya, lihat Mengaitkan kebijakan akses dengan entri akses di Panduan Pengguna Amazon EKS.

Pastikan untuk mengulangi langkah-langkah ini dengan cluster EKS kedua di blok eksekusi Anda, di Wilayah lain, untuk memastikan bahwa kedua cluster dapat diakses oleh sakelar Region.

Konfigurasi

Untuk mengonfigurasi blok eksekusi penskalaan sumber daya EKS, pertama, pastikan Anda memiliki izin yang benar. Untuk informasi selengkapnya, lihat Konfigurasikan izin entri akses EKS.

Perhatikan bahwa sakelar Wilayah saat ini mendukung ReplicaSet sumber daya berikut: apps/v1, Deployment, and apps/v 1.

Kemudian, untuk konfigurasi blok eksekusi, masukkan nilai berikut.

1. Nama langkah: Masukkan nama.
2. Deskripsi langkah (opsional): Masukkan deskripsi langkah.
3. Nama aplikasi: Masukkan nama aplikasi EKS Anda misalnya, MyApplication.
4. Jenis sumber daya Kubernetes: Masukkan jenis sumber daya untuk aplikasi, misalnya, Deployment.
5. Sumber Daya untuk Wilayah: Untuk setiap Wilayah, masukkan informasi untuk kluster EKS, termasuk ARN cluster EKS, namespace sumber daya, dan sebagainya.
6. Persentase untuk mencocokkan kapasitas Region yang diaktifkan: Masukkan persentase yang diinginkan dari pod yang sedang berjalan di Wilayah sumber agar sesuai dengan Region yang diaktifkan.
7. Pendekatan pemantauan kapasitas: Di menu tarik-turun, pilih pendekatan pemantauan untuk sumber daya EKS Anda.

8. Timeout: Masukkan nilai batas waktu.

Kemudian, pilih Save step.

Cara kerjanya

Selama eksekusi rencana, sakelar Wilayah mengambil jumlah maksimum sampel replika selama 24 jam sebelumnya untuk sumber daya target di Wilayah yang Anda aktifkan. Kemudian, menghitung jumlah replika yang diinginkan untuk sumber daya tujuan dengan menggunakan rumus berikut: `ceil(percentToMatch * Source replica count)`

Jika jumlah replika siap tujuan lebih rendah dari nilai yang diinginkan, sakelar Wilayah menskalakan nilai replika sumber daya tujuan ke kapasitas yang diinginkan. Ini menunggu replika siap, memanfaatkan penskalaan otomatis node Anda untuk meningkatkan kapasitas node jika perlu.

Jika `hpaName` bidang opsional tidak kosong, sakelar Region menambal HorizontalPodAutoscaler untuk mencegah penskalaan otomatis selama atau setelah eksekusi dengan menggunakan tambalan berikut: `{"spec":{"behavior":{"scaleDown":{"selectPolicy":"Disabled"}}}}`

Pastikan untuk mengonfigurasi alat pengoreksi drift apa pun, seperti GitOps perkakas, untuk mengabaikan bidang replika untuk sumber daya di tambalan, serta bidang. HorizontalPodAutoscaler

Apa yang dievaluasi sebagai bagian dari evaluasi rencana

Saat sakelar Wilayah mengevaluasi paket Anda, sakelar Wilayah melakukan beberapa pemeriksaan pada blok eksekusi EKS dan izin yang dikonfigurasi. Sakelar wilayah memverifikasi bahwa peran IAM paket memiliki izin yang benar untuk mendeskripsikan kluster EKS dan mencantumkan kebijakan Entri Akses terkait. Region switch juga memvalidasi bahwa peran IAM dikaitkan dengan kebijakan Access Entry yang benar, sehingga switch Region memiliki izin yang diperlukan untuk bertindak pada resource Kubernetes. Terakhir, Region switch mengonfirmasi bahwa kluster EKS yang dikonfigurasi dan sumber daya Kubernetes ada.

Selain itu, Region switch memeriksa apakah ia telah berhasil mengumpulkan dan menyimpan data pemantauan yang diperlukan (jumlah replika Kubernetes) dan menangkap jumlah pod yang sedang berjalan yang diperlukan untuk menjalankan rencana switch Region.

Blok eksekusi penskalaan layanan Amazon ECS

Blok eksekusi penskalaan layanan ECS memungkinkan Anda untuk menskalakan layanan ECS Anda di Wilayah tujuan sebagai bagian dari proses pemulihan Multi-wilayah Anda. Anda dapat menentukan persentase kapasitas, relatif terhadap Wilayah tempat peralihan Wilayah gagal atau dinonaktifkan.

Konfigurasi

Untuk mengkonfigurasi blok eksekusi penskalaan layanan ECS, masukkan nilai berikut.

1. Nama langkah: Masukkan nama.
2. Deskripsi langkah (opsional): Masukkan deskripsi langkah.
3. Sumber Daya untuk Wilayah: Untuk setiap Wilayah, masukkan ARN cluster ECS dan ARN layanan ECS.
4. Persentase untuk mencocokkan jumlah tugas Wilayah sumber: Masukkan persentase tugas yang diinginkan di Wilayah sumber untuk dicocokkan di Wilayah yang diaktifkan.
5. Pendekatan pemantauan kapasitas: Di menu tarik-turun, pilih pendekatan pemantauan untuk sumber daya ECS Anda.
6. Timeout: Masukkan nilai batas waktu.

Kemudian, pilih Save step.

Cara kerjanya

Setelah Anda mengonfigurasi blok eksekusi dalam paket Anda, sakelar Wilayah mengonfirmasi bahwa hanya ada satu layanan ECS sumber dan satu layanan tujuan. Jika ada beberapa layanan, sakelar Wilayah mengembalikan peringatan untuk blok eksekusi. Peralihan wilayah menyimpan data ini di semua Wilayah yang dikonfigurasi untuk paket Anda. Kapasitas target didefinisikan sebagai hitungan yang diinginkan yang ditetapkan pada layanan ECS Anda.

Untuk active/passive pendekatan, sakelar Wilayah menghitung kapasitas baru yang diinginkan untuk layanan ECS di Wilayah tujuan (mengaktifkan). Kapasitas baru yang diinginkan dibandingkan dengan kapasitas yang diinginkan layanan ECS tujuan. Rumus yang digunakan sakelar Region untuk menghitung kapasitas yang diinginkan adalah sebagai berikut:`ceil(percentToMatch * Source Auto Scaling group capacity)`, di mana ceil () adalah fungsi yang membulatkan hasil fraksional apa pun. Jika jumlah yang diinginkan saat ini untuk layanan ECS tujuan lebih tinggi dari kapasitas baru yang diinginkan yang dihitung untuk layanan ECS, pelaksanaan rencana berlangsung. Perhatikan bahwa sakelar Wilayah tidak menurunkan kapasitas layanan ECS.

Jika layanan ECS mengaktifkan Application Autoscaling, Region switch memperbarui kapasitas minimum dalam Application Autoscaling, dan juga memperbarui jumlah yang diinginkan dalam layanan ECS.

Ketika sakelar Wilayah mengeksekusi blok layanan ECS, sakelar Wilayah mencoba meningkatkan kapasitas ECS Wilayah target agar sesuai dengan kapasitas yang diinginkan. Kemudian, sakelar Wilayah menunggu hingga kapasitas layanan ECS yang diminta terpenuhi dalam layanan ECS Wilayah target sebelum peralihan Wilayah melanjutkan ke langkah berikutnya dalam rencana. Jika mau, Anda dapat mengonfigurasi langkah yang harus diselesaikan sebelum pemenuhan selesai dengan menetapkan batas waktu untuk berapa lama sakelar Wilayah menunggu pemenuhan kapasitas.

Jika Anda menggunakan active/active pendekatan, sakelar Wilayah menggunakan Wilayah lain yang dikonfigurasi sebagai sumbernya. Artinya, jika Wilayah dinonaktifkan, sakelar Wilayah menggunakan Wilayah aktif lainnya sebagai sumber untuk mencocokkan persentase untuk skala.

Apa yang dievaluasi sebagai bagian dari evaluasi rencana

Saat sakelar Wilayah mengevaluasi paket Anda, sakelar Wilayah melakukan beberapa pemeriksaan pada konfigurasi dan izin blok eksekusi layanan ECS Anda. Sakelar wilayah memverifikasi bahwa layanan ECS ada di Wilayah sumber dan target, dan memeriksa untuk memastikan bahwa kapasitas maksimum yang ditetapkan untuk layanan ECS Wilayah target cukup untuk menangani persentase kecocokan yang ditentukan dari kapasitas Wilayah target. Peralihan wilayah juga memvalidasi bahwa peran IAM paket memiliki izin yang benar untuk layanan ECS. Untuk informasi selengkapnya tentang izin yang diperlukan untuk blok eksekusi peralihan Wilayah, lihatContoh kebijakan berbasis identitas untuk peralihan Wilayah di ARC.

Selain itu, Region switch memeriksa bahwa `ResourceMonitor` telah berhasil mengumpulkan dan menyimpan data pemantauan yang diperlukan untuk layanan ECS, dan menangkap hitungan jumlah tugas yang berjalan.

Jika salah satu pemeriksaan gagal, sakelar Wilayah mengembalikan pesan peringatan, yang dapat Anda lihat di konsol. Atau, Anda dapat menerima peringatan validasi melalui EventBridge atau dengan menggunakan operasi API.

Blok eksekusi kontrol perutean ARC

Jika Anda telah mengonfigurasi kontrol perutean Amazon Application Recovery Controller (ARC) untuk aplikasi Anda, Anda dapat menambahkan blok eksekusi kontrol perutean ARC untuk mengarahkan lalu lintas aplikasi. Blok eksekusi ini memungkinkan Anda mengubah status satu atau lebih kontrol perutean ARC untuk mengarahkan lalu lintas aplikasi Anda ke tujuan. Wilayah AWS Kontrol perutean ARC mengalihkan lalu lintas dengan menggunakan pemeriksaan kesehatan di Amazon Route 53 yang dikonfigurasi dengan catatan DNS yang terkait dengan kontrol perutean.

Konfigurasi

Untuk mengkonfigurasi blok eksekusi kontrol routing, masukkan nilai-nilai berikut:

1. Nama langkah: Masukkan nama.
2. Deskripsi langkah (opsional): Masukkan deskripsi langkah.
3. Kontrol perutean yang diinginkan: Untuk setiap Wilayah yang ingin Anda aktifkan atau nonaktifkan, masukkan ARN kontrol perutean dan status awal untuk kontrol perutean, Aktif atau Mati.
4. Timeout: Masukkan nilai batas waktu.

Kemudian, pilih Save step.

Pola yang diharapkan untuk blok eksekusi ini adalah untuk menentukan kontrol routing dan status awal yang selaras dengan cara Anda mengatur aplikasi Anda secara spesifik. Wilayah AWS Misalnya, jika Anda memiliki paket yang memungkinkan Anda mengaktifkan Wilayah A dan Wilayah B untuk aplikasi Anda, maka Anda mungkin memiliki kontrol perutean untuk Wilayah A tempat Anda menyetel status ke Aktif dan kontrol perutean untuk Wilayah B tempat Anda menyetel status ke Aktif.

Kemudian, ketika Anda menjalankan rencana dan menentukan bahwa Anda ingin mengaktifkan Wilayah A, alur kerja yang menyertakan blok eksekusi ini memperbarui kontrol perutean yang ditentukan ke Aktif, yang mengarahkan lalu lintas ke Wilayah A.

Cara kerjanya

Dengan mengonfigurasi blok eksekusi kontrol perutean ARC, Anda dapat mengalihkan lalu lintas aplikasi ke tujuan Wilayah AWS, atau, untuk active/active pendekatan, menghentikan lalu lintas agar tidak dialihkan ke Wilayah yang Anda nonaktifkan. Jika paket Anda menyertakan beberapa alur kerja, pastikan Anda memberikan input yang sama untuk catatan DNS untuk semua blok eksekusi kontrol perutean yang Anda gunakan.

Blok ini tidak mendukung mode eksekusi yang tidak beraturan.

Apa yang dievaluasi sebagai bagian dari evaluasi rencana

Saat sakelar Wilayah mengevaluasi paket Anda, sakelar Wilayah melakukan beberapa pemeriksaan pada konfigurasi dan izin blok eksekusi kontrol perutean Anda. Sakelar wilayah memverifikasi bahwa kontrol perutean yang ditentukan dikonfigurasi dan dapat diakses dengan benar.

Sakelar wilayah juga memvalidasi bahwa peran IAM paket memiliki izin yang diperlukan untuk mengakses dan memperbarui status kontrol perutean. Untuk informasi selengkapnya tentang izin

yang diperlukan untuk blok eksekusi peralihan Wilayah, lihatContoh kebijakan berbasis identitas untuk peralihan Wilayah di ARC.

Izin IAM yang benar sangat penting untuk berfungsinya blok eksekusi kontrol perutean. Jika salah satu validasi ini gagal, sakelar Wilayah mengembalikan peringatan bahwa ada masalah, dan menyediakan pesan kesalahan tertentu untuk membantu Anda menyelesaikan masalah izin atau konfigurasi. Ini memastikan bahwa paket Anda memiliki akses yang diperlukan untuk mengelola dan berinteraksi dengan kontrol perutean ARC selama langkah ini berjalan selama eksekusi rencana.

Blok eksekusi Database Global Amazon Aurora

Blok eksekusi Amazon Aurora Global Database memungkinkan Anda melakukan alur kerja pemulihan failover atau switchover untuk database global.

- Failover—Gunakan pendekatan ini untuk melakukan pemulihan dari pemadaman yang tidak direncanakan. Dengan pendekatan ini, Anda melakukan failover lintas wilayah ke salah satu cluster DB sekunder di database global Aurora Anda. Tujuan titik pemulihan (RPO) untuk pendekatan ini biasanya merupakan nilai bukan nol yang diukur dalam hitungan detik. Jumlah kehilangan data tergantung pada kelambatan replikasi database global Aurora pada Wilayah AWS saat kegagalan. Untuk informasi selengkapnya, lihat Memulihkan database global Amazon Aurora dari pemadaman yang tidak direncanakan di Panduan Pengguna Amazon Aurora.
- Switchover — Operasi ini sebelumnya disebut failover terencana terkelola. Gunakan pendekatan ini untuk skenario terkontrol, seperti pemeliharaan operasional dan prosedur operasional terencana lainnya di mana semua cluster Aurora dan layanan lain yang berinteraksi dengan mereka berada dalam keadaan sehat. Karena fitur ini menyinkronkan klaster DB sekunder dengan primer sebelum membuat perubahan lain, RPO adalah 0 (tidak ada kehilangan data). Untuk informasi selengkapnya, lihat Melakukan switchover untuk database global Amazon Aurora di Panduan Pengguna Amazon Aurora.

Konfigurasi

Untuk mengonfigurasi blok eksekusi Aurora Global Database, masukkan nilai berikut:

1. Nama langkah: Masukkan nama.
2. Deskripsi langkah (opsional): Masukkan deskripsi langkah.
3. Nama cluster Aurora Global Database: Masukkan pengenal untuk database global.
4. ARN Cluster untuk Wilayah: Masukkan ARN cluster untuk digunakan di setiap Wilayah dalam rencana.

5. Tentukan opsi untuk database Aurora: Pilih Switchover atau Failover (kehilangan data), tergantung pada bagaimana Anda inginkan
6. Nama klaster Basis Data Global Aurora:
7. Timeout: Masukkan nilai batas waktu.

Kemudian, pilih Save step.

Cara kerjanya

Dengan mengonfigurasi blok eksekusi Aurora Global Databases, Anda dapat melakukan failover atau mengalihkan database global sebagai bagian dari pemulihan aplikasi Anda. Jika Anda menggunakan active/active pendekatan, sakelar Wilayah menggunakan Wilayah lain yang dikonfigurasi sebagai sumbernya. Artinya, jika Wilayah dinonaktifkan, sakelar Wilayah menggunakan Wilayah aktif lainnya sebagai sumber untuk mencocokkan persentase untuk skala.

Blok ini mendukung mode eksekusi yang anggun dan tidak menyenangkan. Pengaturan Unraceful melakukan failover Aurora Global Database, yang dapat menyebabkan kehilangan data.

Untuk informasi selengkapnya tentang pemulihan bencana Aurora Global Database, termasuk failover dan switchover, lihat Menggunakan switchover atau failover di database global Amazon Aurora di Panduan Pengguna Amazon Aurora.

Apa yang dievaluasi sebagai bagian dari evaluasi rencana

Saat sakelar Wilayah mengevaluasi paket Anda, sakelar Wilayah melakukan beberapa pemeriksaan pada konfigurasi dan izin blok eksekusi Aurora Anda. Sakelar wilayah memverifikasi bahwa yang berikut ini benar:

- Kluster global Aurora yang ditentukan dalam konfigurasi ada.
- Ada cluster Aurora DB di Wilayah sumber dan tujuan.
- Cluster DB sumber dan tujuan berada dalam keadaan yang memungkinkan peralihan Database Global.
- Ada instance DB di cluster sumber dan tujuan
- Versi mesin cluster global untuk aksi switchover kompatibel. Ini termasuk memverifikasi bahwa cluster berada pada versi Mayor, Minor, dan patch yang sama, dengan beberapa pengecualian yang tercantum dalam dokumentasi Aurora.

Peralihan wilayah juga memvalidasi bahwa peran IAM paket memiliki izin yang diperlukan untuk failover dan peralihan Aurora. Untuk informasi selengkapnya tentang izin yang diperlukan untuk blok eksekusi peralihan Wilayah, lihatContoh kebijakan berbasis identitas untuk peralihan Wilayah di ARC.

Izin IAM yang benar sangat penting untuk berfungsinya blok eksekusi Aurora dengan benar. Jika salah satu validasi ini gagal, sakelar Wilayah mengembalikan peringatan bahwa ada masalah, dan menyediakan pesan kesalahan tertentu untuk membantu Anda menyelesaikan masalah izin atau konfigurasi. Ini memastikan bahwa rencana Anda memiliki akses yang diperlukan untuk mengelola dan berinteraksi dengan Aurora selama langkah ini berjalan selama eksekusi rencana.

Blok eksekusi persetujuan manual

Blok eksekusi persetujuan manual memungkinkan Anda memasukkan langkah persetujuan yang Anda kaitkan dengan peran IAM. Pengguna dengan akses ke peran dapat menyetujui atau menolak pelaksanaan langkah, untuk menjeda langkah sampai persetujuan diberikan, atau, berpotensi, mencegah rencana dari kemajuan.

Untuk memastikan bahwa persetujuan manual diperlukan selama pelaksanaan rencana, Anda memasukkan langkah persetujuan manual di lokasi tertentu dalam alur kerja, dan kemudian mengonfigurasi peran IAM untuk menentukan siapa yang dapat menyetujui langkah tersebut.

Konfigurasi

Untuk mengonfigurasi blok eksekusi persetujuan manual, masukkan nilai berikut:

1. Nama langkah: Masukkan nama.
2. Deskripsi langkah (opsional): Masukkan deskripsi langkah.
3. Peran persetujuan IAM: Masukkan ARN untuk peran IAM yang memiliki izin untuk menyetujui eksekusi secara manual yang berlanjut untuk rencana peralihan Wilayah. Peran IAM harus berada dalam akun yang merupakan pemilik rencana.
4. Timeout: Masukkan nilai batas waktu.

Kemudian, pilih Save step.

Cara kerjanya

Dengan mengonfigurasi blok eksekusi persetujuan manual, Anda dapat meminta persetujuan sebagai bagian dari pemulihan aplikasi Anda. Untuk blok eksekusi manual, sakelar Region melakukan hal berikut:

- Ketika Region switch menjalankan blok eksekusi manual, itu menghentikan eksekusi dan menetapkan status eksekusi rencana ke persetujuan tertunda.
- Siapa pun yang memiliki akses ke peran yang ditentukan dalam blok eksekusi dapat menyetujui atau menolak eksekusi langkah tersebut.
- Jika mereka menyetujui eksekusi langkah, Region switch melanjutkan dengan eksekusi rencana. Jika mereka menolak, sakelar Wilayah membatalkan eksekusi rencana.

Blok ini tidak mendukung mode eksekusi yang tidak beraturan.

Apa yang dievaluasi sebagai bagian dari evaluasi rencana

Sakelar wilayah tidak menyelesaikan evaluasi apa pun untuk blok eksekusi persetujuan manual.

Tindakan kustom Blok eksekusi Lambda

Blok eksekusi Lambda tindakan kustom memungkinkan Anda menambahkan langkah yang disesuaikan ke rencana dengan menggunakan fungsi Lambda.

Konfigurasi

Untuk mengonfigurasi blok eksekusi Lambda, masukkan nilai berikut:

1. Nama langkah: Masukkan nama.
2. Deskripsi langkah (opsional): Masukkan deskripsi langkah.
3. Fungsi Lambda ARN yang akan dipanggil saat mengaktifkan atau menonaktifkan Wilayah: Tentukan ARN dari fungsi Lambda yang akan dijalankan untuk langkah ini.
4. Wilayah untuk menjalankan fungsi Lambda: Di menu tarik-turun, pilih Wilayah tempat Anda ingin menjalankan fungsi Lambda.
5. Timeout: Masukkan nilai batas waktu.
6. Interval coba lagi: Masukkan interval coba lagi, untuk menjalankan kembali fungsi Lambda jika tidak berhasil dalam interval ini.

Kemudian, pilih Save step.

Cara kerjanya

- Saat membuat blok eksekusi Lambda tindakan kustom, Anda harus menentukan dua fungsi Lambda untuk langkah yang akan dieksekusi — satu di setiap Wilayah paket.

- Anda dapat mengonfigurasi Wilayah mana yang Anda inginkan untuk menjalankan Lambda, misalnya, di Wilayah pengaktifan atau di Wilayah penonaktifan. Namun, jika Anda mengeksekusi di Wilayah penonaktifan, Anda mengambil ketergantungan pada Wilayah itu. Kami tidak menyarankan Anda mengambil ketergantungan pada Wilayah penonaktifan.

Blok ini mendukung mode eksekusi yang anggun dan tidak menyenangkan. Dalam mode eksekusi yang tidak menyenangkan, sakelar Wilayah melewatkan langkah blok eksekusi Lambda.

Apa yang dievaluasi sebagai bagian dari evaluasi rencana

Saat sakelar Wilayah mengevaluasi paket Anda, sakelar Wilayah melakukan beberapa pemeriksaan pada konfigurasi dan izin blok eksekusi Lambda Anda. Sakelar wilayah memverifikasi bahwa yang berikut ini benar:

- Fungsi Lambda yang ditentukan dalam konfigurasi ada.
- Pengaturan konkurensi fungsi Lambda tidak dibatasi, termasuk memverifikasi yang berikut:
  - Konkurensi tidak disetel ke 0.
  - Setidaknya satu eksekusi bersamaan tersedia, atau konkurensi tanpa syarat itu ada.

Sakelar wilayah melakukan dry run fungsi Lambda untuk memvalidasi parameter dan izin yang ditentukan, tanpa menjalankan logika fungsi yang sebenarnya. Biaya Lambda standar dikeluarkan saat Anda melakukan dry run.

Peralihan wilayah juga memvalidasi bahwa peran IAM paket memiliki izin yang diperlukan untuk eksekusi Lambda. Untuk informasi selengkapnya tentang izin yang diperlukan untuk blok eksekusi peralihan Wilayah, lihatContoh kebijakan berbasis identitas untuk peralihan Wilayah di ARC.

Izin IAM yang benar sangat penting untuk berfungsinya blok eksekusi Lambda dengan benar. Jika salah satu validasi ini gagal, sakelar Wilayah mengembalikan peringatan bahwa ada masalah, dan menyediakan pesan kesalahan tertentu untuk membantu Anda menyelesaikan masalah izin atau konfigurasi. Ini memastikan bahwa paket Anda memiliki akses yang diperlukan untuk mengelola dan berinteraksi dengan Lambda selama langkah ini berjalan selama eksekusi rencana.

Blok eksekusi pemeriksaan kesehatan Amazon Route 53

Blok eksekusi pemeriksaan kesehatan Amazon Route 53 memungkinkan Anda menentukan Wilayah tempat lalu lintas aplikasi Anda akan diarahkan selama failover. Blok eksekusi membuat pemeriksaan kesehatan Amazon Route 53, yang kemudian Anda lampirkan ke catatan DNS Route 53 di akun

Anda. Saat Anda menjalankan rencana peralihan Wilayah, status pemeriksaan kesehatan Route 53 diperbarui, dan lalu lintas dialihkan berdasarkan konfigurasi DNS Anda.

Konfigurasi

Untuk mengonfigurasi blok eksekusi pemeriksaan kesehatan Route 53, masukkan nilai berikut:

1. Nama langkah: Masukkan nama.
2. Deskripsi langkah (opsional): Masukkan deskripsi langkah.
3. ID zona yang dihosting: Id zona yang dihosting untuk domain dan catatan DNS Anda di Route 53.
4. Nama rekaman: Nama rekaman (nama domain) untuk catatan yang Anda gunakan, dengan pemeriksaan kesehatan terkait, untuk mengarahkan lalu lintas untuk aplikasi Anda.
5. Record set identifier (opsional): Jika perlu, berikan informasi tambahan untuk membantu Region beralih mengidentifikasi kumpulan rekaman yang Anda kaitkan dengan masing-masing Wilayah untuk mengarahkan lalu lintas ke Wilayah tersebut. Mengidentifikasi informasi untuk dimasukkan, jika diperlukan, adalah sebagai berikut:
   - Record set identifier: Anda dapat memasukkan Set identifier atau lalu lintas Nilai/Rute untuk set rekaman.
   - Wilayah: Masukkan Wilayah yang terkait dengan kumpulan rekaman yang memiliki informasi pengenal kumpulan catatan yang Anda masukkan.
6. Pilih Simpan langkah.
7. Konfigurasikan pemeriksaan kesehatan di Route 53.

   Sakelar wilayah menyediakan ID pemeriksaan kesehatan, untuk setiap Wilayah, untuk setiap nama rekaman dalam zona yang dihosting yang ditentukan dalam blok eksekusi. Pastikan Anda mengonfigurasi pemeriksaan kesehatan untuk kumpulan catatan yang sesuai di akun Anda di Route 53 sehingga sakelar Wilayah dapat mengarahkan lalu lintas aplikasi Anda dengan benar selama eksekusi paket. Di tab Pemeriksaan Kesehatan pada halaman detail rencana, Anda dapat melihat pemeriksaan kesehatan untuk semua blok eksekusi dan Wilayah.

Cara kerjanya

Anda menambahkan blok eksekusi pemeriksaan kesehatan ke alur kerja sakelar Wilayah sehingga Anda dapat mengarahkan lalu lintas ke Wilayah sekunder, untuk active/passive konfigurasi, atau menjauh dari Wilayah yang dinonaktifkan, untuk konfigurasi. active/active Jika Anda menambahkan beberapa alur kerja ke paket Anda, berikan nilai konfigurasi yang sama untuk semua blok eksekusi pemeriksaan kesehatan yang menggunakan catatan DNS yang sama.

Berdasarkan informasi yang Anda berikan saat mengonfigurasi blok eksekusi, sakelar Wilayah mencoba menentukan kumpulan catatan yang benar untuk setiap Wilayah dalam paket Anda. Biasanya, ID zona yang dihosting dan nama catatan adalah informasi yang cukup untuk menentukan kumpulan rekaman dan Wilayah terkait. Jika tidak, ketika sakelar Wilayah menjalankan evaluasi paket otomatisnya setelah Anda membuat paket, peringatan akan dikembalikan untuk memberi tahu Anda bahwa informasi lebih lanjut diperlukan.

Sakelar wilayah menjual pemeriksaan kesehatan untuk setiap blok eksekusi pemeriksaan kesehatan Route 53. Untuk rencana yang menggunakan pendekatan active/passive pemulihan, pemeriksaan kesehatan untuk Wilayah primer dimulai sebagai sehat, dan pemeriksaan kesehatan untuk Wilayah siaga awalnya diatur ke tidak sehat. Untuk rencana yang menggunakan pendekatan active/active pemulihan, pemeriksaan kesehatan untuk semua Wilayah dimulai dalam keadaan sehat.

Untuk mengaktifkan peralihan Wilayah agar berhasil menjalankan blok eksekusi ini untuk paket Anda, Anda harus menambahkan pemeriksaan kesehatan ke catatan DNS Anda.

Untuk sebuah active/active rencana, langkah eksekusi bekerja dengan cara berikut:

- Ketika alur kerja nonaktif berjalan untuk suatu Wilayah, pemeriksaan kesehatan diatur ke tidak sehat, dan lalu lintas tidak lagi diarahkan ke Wilayah.
- Ketika alur kerja aktivasi berjalan untuk suatu Wilayah, pemeriksaan kesehatan diatur ke sehat, dan lalu lintas diarahkan ke Wilayah.

Untuk sebuah active/passive rencana, langkah eksekusi bekerja dengan cara berikut:

- Ketika alur kerja aktivasi berjalan untuk suatu Wilayah, pemeriksaan kesehatan untuk Wilayah tersebut diatur ke sehat, dan lalu lintas diarahkan ke Wilayah. Pada saat yang sama, pemeriksaan kesehatan untuk Wilayah lain dalam rencana diatur menjadi tidak sehat, dan lalu lintas berhenti diarahkan ke Wilayah itu.

Apa yang dievaluasi sebagai bagian dari evaluasi rencana

Saat sakelar Wilayah mengevaluasi paket Anda, sakelar Wilayah melakukan beberapa pemeriksaan pada konfigurasi dan izin blok eksekusi Lambda Anda. Sakelar wilayah memverifikasi bahwa pemeriksaan kesehatan dilampirkan ke catatan DNS yang ditentukan dalam konfigurasi blok eksekusi. Artinya, sakelar Wilayah memverifikasi bahwa catatan DNS untuk spesifik Wilayah AWS dikonfigurasi untuk menggunakan pemeriksaan kesehatan untuk Wilayah tersebut.

Buat rencana anak

Untuk mendukung skenario pemulihan yang lebih kompleks, Anda dapat membuat rencana anak dengan menambahkannya dengan blok eksekusi rencana peralihan Wilayah. Hirarki terbatas pada dua tingkatan, tetapi satu rencana induk dapat mencakup beberapa rencana anak.

Untuk kompatibilitas, paket anak harus mendukung semua Wilayah yang didukung oleh paket induk. Selain itu, pendekatan pemulihan, active/active atau aktif/pasif, harus sama untuk rencana orang tua dan anak.

Ingatlah cara-cara berikut di mana rencana anak merespons perubahan yang Anda buat pada rencana orang tua dan skenario rencana orang tua.

- Blok eksekusi induk ditandai sebagai selesai ketika semua rencana anak dan blok eksekusi lainnya di dalamnya selesai.
- Jika ada langkah yang gagal dalam paket anak apa pun, blok eksekusi rencana peralihan Wilayah gagal dalam paket induk.
- Tindakan kontrol yang dimulai dalam rencana induk selama langkah peralihan Wilayah, seperti jeda, sakelar yang anggun atau tidak pantas, atau pembatalan, secara otomatis dicoba pada paket anak, terlepas dari langkah rencana anak saat ini.
- Operasi skips memiliki perilaku khusus: rencana induk dilewati, tetapi rencana anak akan tetap dijalankan.
- Jika rencana anak sudah dijalankan di blok sakelar Wilayah, untuk menentukan apakah paket tersebut terus berjalan, sakelar Region menilai kompatibilitas paket anak dengan paket induk. Jika konfigurasi paket anak cocok dengan persyaratan paket induk, sakelar Region memperlakukan paket anak seolah-olah itu dimulai oleh rencana induk.
- Langkah rencana induk akan gagal jika paket anak berjalan dengan parameter konfigurasi yang tidak kompatibel, seperti berikut ini:
  - Rencana anak beroperasi di Wilayah yang berbeda
  - Rencana anak menjalankan operasi penonaktifan saat sakelar Wilayah mengharapkannya untuk menjalankan operasi pengaktifan
- Jika rencana anak berhasil diselesaikan selama rencana orang tua dijeda, rencana induk akan berhasil ketika rencana induk dilanjutkan.

Buat pemicu untuk paket sakelar Wilayah

Jika Anda ingin mengotomatiskan pemulihan untuk aplikasi Anda di sakelar Wilayah, Anda dapat membuat satu atau beberapa pemicu untuk paket peralihan Wilayah Anda. Pemicu secara otomatis mulai menjalankan rencana sakelar Wilayah, berdasarkan kondisi CloudWatch alarm yang Anda pilih.

Untuk membuat pemicu untuk rencana peralihan Wilayah

1. Setelah Anda membuat rencana, pada halaman Rincian rencana, pilih tab Pemicu.
2. Pilih Kelola pemicu.
3. Pilih alur kerja yang ingin Anda otomatiskan eksekusi, lalu pilih Tambah pemicu.
4. Berikan deskripsi untuk pemicunya.
5. Pilih CloudWatch alarm, lalu pilih hingga 10 CloudWatch alarm untuk membuat kondisi pemicu.

   Ketika Anda memilih lebih dari satu kondisi, semua kondisi harus dipenuhi sebelum eksekusi otomatis rencana akan dimulai.

Jalankan rencana peralihan Wilayah untuk memulihkan aplikasi

Untuk memulihkan aplikasi Wilayah AWS saat mengalami gangguan, Anda menjalankan rencana peralihan Wilayah di Amazon Application Recovery Controller (ARC).

- Jika aplikasi Anda diterapkan dengan active/active pendekatan, alur kerja dalam paket Anda menonaktifkan Wilayah yang mengalami gangguan sehingga Wilayah aktif Anda yang lain diskalakan dengan tepat dan mulai menerima semua lalu lintas aplikasi Anda.
- Jika aplikasi Anda di-deploy dengan active/passive pendekatan, alur kerja dalam paket Anda akan menonaktifkan Region yang mengalami gangguan dan mengaktifkan Region siaga Anda, dengan meningkatkan resource Anda di sana, jika diperlukan, dan mengarahkan lalu lintas aplikasi Anda ke Region siaga.

Untuk melakukan pemulihan aplikasi secara manual, jalankan rencana peralihan Wilayah Anda dengan melakukan hal berikut.

Opsi lainnya adalah memicu eksekusi secara otomatis dengan CloudWatch alarm Amazon tertentu yang Anda tentukan untuk memulai eksekusi rencana. Anda dapat menentukan pemicu untuk eksekusi rencana saat membuat atau memperbarui rencana. Untuk informasi selengkapnya, lihat Buat pemicu untuk paket sakelar Wilayah.

Untuk menjalankan rencana peralihan Wilayah

1. Di AWS Management Console, navigasikan ke Wilayah AWS yang ingin Anda aktifkan untuk aplikasi Anda.
2. Pada konsol Amazon Application Recovery Controller (ARC), pilih Region switch, lalu pilih paket yang ingin Anda jalankan.
3. Pilih Execute plan.
4. Jika rencana Anda menyertakan langkah-langkah persetujuan manual, setujui setiap langkah saat diminta.

Saat rencana dijalankan, Anda dapat melacak kemajuannya di halaman detail eksekusi, yang terbuka saat Anda memilih untuk menjalankan rencana.

Anda juga dapat melihat informasi tentang pemulihan aplikasi yang sedang berlangsung di dasbor sakelar Wilayah. Pada konsol sakelar Wilayah, di navigasi kiri, di bawah sakelar Wilayah, pilih salah satu dari berikut ini:

- Dasbor global
- Eksekusi atas nama Wilayah

Ketahuilah bahwa, jika ada gangguan di suatu Wilayah, dasbor global mungkin tidak menampilkan semua data paket Anda. Karena itu, kami menyarankan Anda hanya mengandalkan dasbor eksekusi Regional selama acara operasional. Dasbor eksekusi Regional lebih tangguh karena menggunakan bidang data sakelar Wilayah lokal.

Ketika eksekusi rencana selesai, Anda dapat melihat informasi tentang eksekusi rencana, dan rencana lain yang telah dijalankan oleh sakelar Wilayah, pada halaman Rincian rencana di tab Riwayat eksekusi rencana.

Dasbor sakelar wilayah

Sakelar wilayah menyertakan dasbor global yang dapat Anda gunakan untuk mengamati status paket peralihan Wilayah di seluruh organisasi dan Wilayah Anda. Sakelar wilayah juga memiliki dasbor eksekusi Regional yang hanya menampilkan eksekusi paket di Wilayah tempat Anda saat ini masuk ke. AWS Management Console

Ketahuilah bahwa, jika ada gangguan di suatu Wilayah, dasbor global mungkin tidak menampilkan semua data paket Anda. Karena itu, kami menyarankan Anda hanya mengandalkan dasbor eksekusi

Regional selama acara operasional. Dasbor eksekusi Regional lebih tangguh karena menggunakan bidang data sakelar Wilayah lokal.

Untuk membuka dasbor global sakelar Wilayah

1. Buka konsol ARC dihttps://console.aws.amazon.com/route53recovery/home#/dashboard.
2. Di bawah Sakelar Wilayah, pilih Dasbor global.

Untuk membuka Dashboard Regional switch Regional

1. Buka konsol ARC dihttps://console.aws.amazon.com/route53recovery/home#/dashboard.
2. Di bawah Sakelar Wilayah, pilih Dasbor Regional.

Dukungan lintas akun di sakelar Wilayah

Di sakelar Wilayah, Anda dapat menambahkan sumber daya dari akun lain ke paket Anda. Anda juga dapat membagikan paket peralihan Wilayah dengan akun lain. Untuk informasi selengkapnya, silakan lihat bagian-bagian berikut ini.

Sumber daya lintas akun

Peralihan wilayah memungkinkan sumber daya dihosting di akun yang terpisah dari akun yang berisi paket peralihan Wilayah. Ketika Region switch mengeksekusi rencana, itu mengasumsikan ExecutionRole. Jika paket menggunakan sumber daya dari akun yang berbeda dari akun yang menghosting paket, maka sakelar Region menggunakan ExecutionRole untuk mengasumsikan akses sumber daya tersebut. crossAccountRole

Setiap sumber daya dalam paket switch Region memiliki dua bidang opsional: crossAccountRole dan externalid.

- crossAccountRole: Peran ini memungkinkan akses ke sumber daya di akun yang berbeda dari akun yang menghosting paket peralihan Wilayah. Peran tersebut hanya memerlukan izin untuk bertindak atas sumber daya dalam akunnya — peran tersebut tidak memerlukan izin untuk bertindak atas sumber daya di akun yang menghosting paket peralihan Wilayah.
- ExternalId: Ini adalah ID eksternal STS dari kebijakan kepercayaan akun yang berisi sumber daya yang memerlukan tindakan. Ini adalah string alfanumerik yang merupakan rahasia bersama antara kedua akun.

Paket peralihan Wilayah Berbagi

Sakelar wilayah terintegrasi dengan AWS Resource Access Manager (AWS RAM) untuk memungkinkan Anda berbagi paket. Akun AWS Saat membagikan paket, akun yang Anda tentukan dapat melihat detail paket, menjalankan rencana, dan melihat eksekusi paket, yang memberikan kontrol dan fleksibilitas lebih untuk kemampuan pemulihan di berbagai tim.

Untuk memulai berbagi lintas akun di sakelar Wilayah, Anda membuat pembagian sumber daya di AWS RAM. Pembagian sumber daya menentukan peserta yang berwenang untuk membagikan paket yang dimiliki akun Anda. Peserta dapat melihat dan menjalankan rencana bersama melalui konsol, CLI, atau. AWS SDKs

Penting: Anda Akun AWS harus memiliki rencana yang ingin Anda bagikan. Anda tidak dapat membagikan rencana yang telah dibagikan kepada Anda. Untuk berbagi rencana dengan organisasi Anda, atau dengan unit organisasi di AWS Organizations, Anda harus mengaktifkan berbagi dengan Organizations.

Untuk informasi lebih lanjut tentang AWS RAM, lihat Support berbagi paket di seluruh akun untuk peralihan Wilayah ARC.

Support berbagi paket di seluruh akun untuk peralihan Wilayah ARC

Amazon Application Recovery Controller (ARC) terintegrasi dengan AWS Resource Access Manager untuk mengaktifkan berbagi sumber daya. AWS RAM adalah layanan yang memungkinkan Anda untuk berbagi sumber daya dengan orang lain Akun AWS atau melalui AWS Organizations. Untuk sakelar Wilayah ARC, Anda dapat membagikan paket sakelar Wilayah. (Untuk menggunakan sumber daya dari akun lain dalam paket Anda, Anda menggunakan peran CrossAccount. Untuk mempelajari lebih lanjut, lihatSumber daya lintas akun.)

Dengan AWS RAM, Anda berbagi sumber daya yang Anda miliki dengan membuat pembagian sumber daya. Pembagian sumber daya menentukan sumber daya untuk dibagikan, dan peserta untuk membagikannya. Peserta dapat mencakup:

- Khusus Akun AWS di dalam atau di luar organisasi pemilik di AWS Organizations
- Unit organisasi di dalam organisasinya di AWS Organizations
- Seluruh organisasinya di AWS Organizations

Untuk informasi selengkapnya AWS RAM, lihat Panduan AWS RAM Pengguna.

Dengan menggunakan AWS Resource Access Manager untuk berbagi paket di seluruh akun di ARC, Anda dapat menggunakan satu paket dengan beberapa paket berbeda Akun AWS. Saat Anda memilih untuk membagikan paket, paket lain Akun AWS yang Anda tentukan dapat menjalankan rencana untuk melakukan pemulihan aplikasi.

AWS RAM adalah layanan yang membantu AWS pelanggan berbagi sumber daya dengan aman. Akun AWS Dengan AWS RAM, Anda dapat berbagi sumber daya dalam organisasi atau unit organisasi (OUs) di AWS Organizations, dengan menggunakan peran dan pengguna IAM. AWS RAM adalah cara terpusat dan terkontrol untuk berbagi rencana.

Ketika Anda berbagi rencana, Anda dapat mengurangi jumlah total rencana yang dibutuhkan organisasi Anda. Dengan paket bersama, Anda dapat mengalokasikan total biaya menjalankan rencana di berbagai tim, untuk memaksimalkan manfaat ARC dengan biaya lebih rendah. Berbagi rencana di seluruh akun juga dapat memudahkan proses orientasi beberapa aplikasi ke ARC, terutama jika Anda memiliki sejumlah besar aplikasi yang didistribusikan di beberapa akun dan tim operasi.

Untuk memulai berbagi lintas akun di ARC, Anda membuat pembagian sumber daya i n AWS RAM. Pembagian sumber daya menentukan peserta yang berwenang untuk membagikan paket yang dimiliki akun Anda.

Topik ini menjelaskan cara berbagi sumber daya yang Anda miliki, dan cara menggunakan sumber daya yang dibagikan dengan Anda.

Daftar Isi

Prasyarat untuk berbagi rencana

- Untuk berbagi rencana, Anda harus memilikinya di Anda Akun AWS. Ini berarti bahwa sumber daya harus dialokasikan atau disediakan di akun Anda. Anda tidak dapat membagikan rencana yang telah dibagikan kepada Anda.
- Untuk berbagi rencana dengan organisasi Anda atau unit organisasi di AWS Organizations, Anda harus mengaktifkan berbagi dengan AWS Organizations. Untuk informasi lebih lanjut, lihat Aktifkan pembagian dengan AWS Organizations dalam Panduan Pengguna AWS RAM .

Berbagi rencana

Saat Anda membagikan paket, peserta yang Anda tentukan untuk dibagikan paket dapat melihat dan, jika Anda memberikan izin tambahan, jalankan paket tersebut.

Untuk berbagi rencana, Anda harus menambahkannya ke berbagi sumber daya. Pembagian sumber daya adalah AWS RAM sumber daya yang memungkinkan Anda berbagi sumber daya Anda Akun AWS. Pembagian sumber daya menentukan sumber daya untuk dibagikan, dan peserta yang mereka bagikan. Untuk membagikan paket, Anda dapat membuat pembagian sumber daya baru atau menambahkan sumber daya ke pembagian sumber daya yang ada. Untuk membuat pembagian sumber daya baru, Anda dapat menggunakan AWS RAM konsol, atau menggunakan operasi AWS RAM API dengan AWS Command Line Interface atau AWS SDKs.

Jika Anda adalah bagian dari organisasi AWS Organizations dan berbagi dalam organisasi Anda diaktifkan, peserta dalam organisasi Anda secara otomatis diberikan akses ke paket bersama. Jika tidak, peserta menerima undangan untuk bergabung dengan pembagian sumber daya dan diberikan akses ke paket bersama setelah menerima undangan.

Anda dapat membagikan paket yang Anda miliki dengan menggunakan AWS RAM konsol, atau dengan menggunakan operasi AWS RAM API dengan AWS CLI atau SDKs.

Untuk berbagi paket yang Anda miliki dengan menggunakan AWS RAM konsol

Lihat Membuat berbagi sumber daya di Panduan AWS RAM Pengguna.

Untuk berbagi rencana yang Anda miliki dengan menggunakan AWS CLI

Gunakan perintah create-resource-share.

Memberikan izin untuk berbagi paket

Berbagi paket di seluruh akun memerlukan izin tambahan berikut untuk prinsipal IAM yang berbagi paket dengan menggunakan: AWS RAM

```
# read and execute plan permissions
"arc-region-switch:GetPlan",
"arc-region-switch:GetPlanInRegion",
"arc-region-switch:GetPlanExecution",
"arc-region-switch:ListPlanExecutionEvents",
"arc-region-switch:ListPlanExecutions",
"arc-region-switch:ListRoute53HealthChecks",
"arc-region-switch:GetPlanEvaluationStatus",
"arc-region-switch:StartPlanExecution",
"arc-region-switch:CancelPlanExecution",
"arc-region-switch:UpdatePlanExecution",
"arc-region-switch:UpdatePlanExecutionStep"
```

Pemilik yang membagikan paket harus memiliki izin berikut. Jika Anda mencoba membagikan paket AWS RAM tanpa izin ini, kesalahan akan dikembalikan.

```
"arc-region-switch:PutResourcePolicy" # Permission only apis
"arc-region-switch:DeleteResourcePolicy" # Permission only apis
"arc-region-switch:GetResourcePolicy" # Permission only apis
```

Untuk informasi selengkapnya tentang cara AWS Resource Access Manager menggunakan IAM, lihat Cara AWS Resource Access Manager menggunakan IAM di AWS RAM Panduan Pengguna.

Membatalkan berbagi rencana bersama

Saat Anda membatalkan pembagian rencana, berikut ini berlaku untuk peserta dan pemilik:

- Peserta tidak dapat lagi melihat atau menjalankan rencana yang tidak dibagikan.

Untuk membatalkan berbagi paket bersama yang Anda miliki, hapus dari pembagian sumber daya. Anda dapat melakukan ini dengan menggunakan AWS RAM konsol atau dengan menggunakan operasi AWS RAM API dengan AWS CLI atau SDKs.

Untuk membatalkan berbagi paket bersama yang Anda miliki menggunakan konsol AWS RAM

Lihat Memperbarui Pembagian Sumber Daya di Panduan Pengguna.AWS RAM

Untuk membatalkan berbagi paket bersama yang Anda miliki menggunakan AWS CLI

Gunakan perintah disassociate-resource-share.

Mengidentifikasi rencana bersama

Pemilik dan peserta dapat mengidentifikasi rencana bersama dengan melihat informasi di AWS RAM. Mereka juga bisa mendapatkan informasi tentang sumber daya bersama dengan menggunakan konsol ARC dan AWS CLI.

Secara umum, untuk mempelajari lebih lanjut tentang sumber daya yang telah Anda bagikan atau yang telah dibagikan dengan Anda, lihat informasi di Panduan AWS Resource Access Manager Pengguna:

- Sebagai pemilik, Anda dapat melihat semua sumber daya yang Anda bagikan dengan orang lain dengan menggunakan AWS RAM. Untuk informasi selengkapnya, lihat Melihat sumber daya bersama Anda di AWS RAM.
- Sebagai peserta, Anda dapat melihat semua sumber daya yang dibagikan dengan Anda menggunakan AWS RAM. Untuk informasi selengkapnya, lihat Melihat sumber daya bersama Anda di AWS RAM.

Sebagai pemilik, Anda dapat menentukan apakah Anda membagikan paket dengan melihat informasi di AWS Management Console atau dengan menggunakan operasi AWS Command Line Interface With ARC API.

Untuk mengidentifikasi apakah paket yang Anda miliki dibagikan dengan menggunakan konsol

Di AWS Management Console halaman detail untuk rencana, lihat status berbagi paket.

Sebagai peserta, ketika sebuah paket dibagikan dengan Anda, Anda biasanya harus menerima pembagian sehingga Anda dapat mengakses paket tersebut.

Tanggung jawab dan izin untuk paket bersama

Izin untuk pemilik

Peserta dapat melihat atau menjalankan rencana (jika mereka memiliki izin yang benar).

Izin untuk peserta

Saat Anda membagikan paket yang Anda miliki dengan orang lain Akun AWS, peserta dapat melihat atau menjalankan rencana (jika mereka memiliki izin yang benar).

Saat Anda membagikan paket dengan menggunakan AWS RAM, peserta memiliki, secara default, izin hanya-baca. Untuk meninjau daftar izin hanya-baca untuk sakelar Wilayah, lihat. Izin baca-saja Peserta memerlukan izin tambahan untuk menjalankan rencana peralihan Wilayah. Peserta yang perlu menjalankan rencana memerlukan izin tambahan. Ketahuilah bahwa Anda tidak dapat memberikan izin kepada AWS RAM peserta untuk operasi berikut:

- `ApprovePlanExecutionStep`
- `UpdatePlan`

Biaya penagihan

Pemilik rencana di ARC ditagih untuk biaya yang terkait dengan rencana tersebut. Tidak ada biaya tambahan, untuk pemilik rencana atau untuk peserta, untuk membuat sumber daya yang dihosting dalam rencana.

Untuk informasi dan contoh harga terperinci, lihat Harga Amazon Application Recovery Controller (ARC) dan gulir ke bawah ke Amazon Application Recovery Controller (ARC).

Kuota

Semua sumber daya yang dibuat dalam rencana bersama dihitung terhadap kuota untuk pemilik paket.

Untuk daftar kuota paket peralihan Wilayah, lihatKuota untuk sakelar Wilayah.

Pengalihan Identity and Access Management untuk Wilayah di ARC

AWS Identity and Access Management (IAM) adalah Layanan AWS yang membantu administrator mengontrol akses ke AWS sumber daya dengan aman. Administrator IAM mengontrol siapa yang dapat diautentikasi (masuk) dan diberi wewenang (memiliki izin) untuk menggunakan sumber daya ARC. IAM adalah Layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

Daftar Isi

Bagaimana sakelar Wilayah di ARC bekerja dengan IAM

Sebelum Anda menggunakan IAM untuk mengelola akses ke ARC, pelajari fitur IAM apa yang tersedia untuk digunakan dengan ARC.

Sebelum Anda menggunakan IAM untuk mengelola akses ke sakelar Wilayah di Amazon Application Recovery Controller (ARC), pelajari fitur IAM apa yang tersedia untuk digunakan dengan sakelar Wilayah.

Fitur IAM yang dapat Anda gunakan dengan sakelar Wilayah di Amazon Application Recovery Controller (ARC)

Fitur IAM	Dukungan sakelar wilayah
Kebijakan berbasis identitas	Ya
Kebijakan berbasis sumber daya	Ya
Tindakan kebijakan	Ya
Sumber daya kebijakan	Ya
Kunci kondisi kebijakan	Ya
ACLs	Ya
ABAC (tanda dalam kebijakan)	Ya
Kredensial sementara	Ya
Izin principal	Ya
Peran layanan	Tidak
Peran terkait layanan	Tidak

Untuk mendapatkan tampilan tingkat tinggi dan keseluruhan tentang cara kerja AWS layanan dengan sebagian besar fitur IAM, lihat AWS layanan yang bekerja dengan IAM di Panduan Pengguna IAM.

Kebijakan berbasis identitas untuk peralihan Wilayah

Mendukung kebijakan berbasis identitas: Ya

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, seperti pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana, dan berdasarkan kondisi seperti apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Tentukan izin IAM kustom dengan kebijakan terkelola pelanggan dalam Panduan Pengguna IAM.

Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan secara spesifik apakah tindakan dan sumber daya diizinkan atau ditolak, serta kondisi yang menjadi dasar dikabulkan atau ditolaknya tindakan tersebut. Anda tidak dapat menentukan secara spesifik prinsipal dalam sebuah kebijakan berbasis identitas karena prinsipal berlaku bagi pengguna atau peran yang melekat kepadanya. Untuk mempelajari semua elemen yang dapat Anda gunakan dalam kebijakan JSON, lihat Referensi elemen kebijakan JSON IAM dalam Panduan Pengguna IAM.

Untuk melihat contoh kebijakan berbasis identitas ARC, lihat. Contoh kebijakan berbasis identitas di Amazon Application Recovery Controller (ARC)

Kebijakan berbasis sumber daya dalam peralihan Wilayah

Mendukung kebijakan berbasis sumber daya: Ya

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu.

Tindakan kebijakan untuk peralihan Wilayah

Mendukung tindakan kebijakan: Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen `Action` dari kebijakan JSON menjelaskan tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam sebuah kebijakan. Tindakan kebijakan biasanya memiliki nama yang sama dengan operasi AWS API terkait. Ada beberapa pengecualian, misalnya tindakan hanya izin yang tidak memiliki operasi API yang cocok. Ada juga beberapa operasi yang memerlukan beberapa tindakan dalam suatu kebijakan. Tindakan tambahan ini disebut tindakan dependen.

Sertakan tindakan dalam kebijakan untuk memberikan izin untuk melakukan operasi terkait.

Tindakan kebijakan dalam sakelar ARC untuk Wilayah menggunakan awalan berikut sebelum tindakan:

```
arc-region-switch
```

Untuk menetapkan secara spesifik beberapa tindakan dalam satu pernyataan, pisahkan tindakan tersebut dengan koma. Misalnya, berikut ini:

```
"Action": [
      "arc-region-switch:action1",
      "arc-region-switch:action2"
         ]
```

Anda dapat menentukan beberapa tindakan menggunakan wildcard (*). Sebagai contoh, untuk menentukan semua tindakan yang dimulai dengan kata `Describe`, sertakan tindakan berikut:

```
"Action": "arc-region-switch:Describe*"
```

Untuk melihat contoh kebijakan berbasis identitas ARC untuk peralihan Wilayah, lihat. Contoh kebijakan berbasis identitas untuk peralihan Wilayah di ARC

Sumber daya kebijakan untuk peralihan Wilayah

Mendukung sumber daya kebijakan: Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen kebijakan JSON `Resource` menentukan objek yang menjadi target penerapan tindakan. Pernyataan harus menyertakan elemen `Resource` atau `NotResource`. Praktik terbaiknya, tentukan sumber daya menggunakan Amazon Resource Name (ARN). Anda dapat melakukan ini untuk tindakan yang mendukung jenis sumber daya tertentu, yang dikenal sebagai izin tingkat sumber daya.

Untuk tindakan yang tidak mendukung izin di tingkat sumber daya, misalnya operasi pencantuman, gunakan wildcard (*) untuk menunjukkan bahwa pernyataan tersebut berlaku untuk semua sumber daya.

```
"Resource": "*"
```

Untuk melihat contoh kebijakan berbasis identitas ARC untuk peralihan Wilayah, lihat. Contoh kebijakan berbasis identitas untuk peralihan Wilayah di ARC

Kunci kondisi kebijakan untuk sakelar Wilayah

Mendukung kunci kondisi kebijakan khusus layanan: Yes

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen `Condition` (atau blok `Condition`) akan memungkinkan Anda menentukan kondisi yang menjadi dasar suatu pernyataan berlaku. Elemen `Condition` bersifat opsional. Anda dapat membuat ekspresi bersyarat yang menggunakan operator kondisi, misalnya sama dengan atau kurang dari, untuk mencocokkan kondisi dalam kebijakan dengan nilai-nilai yang diminta.

Jika Anda menentukan beberapa elemen `Condition` dalam sebuah pernyataan, atau beberapa kunci dalam elemen `Condition` tunggal, maka AWS akan mengevaluasinya menggunakan operasi `AND` logis. Jika Anda menentukan beberapa nilai untuk satu kunci kondisi, AWS mengevaluasi kondisi menggunakan `OR` operasi logis. Semua kondisi harus dipenuhi sebelum izin pernyataan diberikan.

Anda juga dapat menggunakan variabel placeholder saat menentukan kondisi. Sebagai contoh, Anda dapat memberikan izin kepada pengguna IAM untuk mengakses sumber daya hanya jika izin tersebut mempunyai tanda yang sesuai dengan nama pengguna IAM mereka. Untuk informasi selengkapnya, lihat Elemen kebijakan IAM: variabel dan tanda dalam Panduan Pengguna IAM.

AWS mendukung kunci kondisi global dan kunci kondisi khusus layanan. Untuk melihat semua kunci kondisi AWS global, lihat kunci konteks kondisi AWS global di Panduan Pengguna IAM.

Untuk melihat contoh kebijakan berbasis identitas ARC untuk peralihan Wilayah, lihat. Contoh kebijakan berbasis identitas untuk peralihan Wilayah di ARC

Daftar kontrol akses (ACLs) di sakelar Wilayah

Mendukung ACLs: Ya

Access control lists (ACLs) mengontrol prinsipal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACLs mirip dengan kebijakan berbasis sumber daya, meskipun mereka tidak menggunakan format dokumen kebijakan JSON.

Kontrol akses berbasis atribut (ABAC) dengan sakelar Wilayah

Mendukung ABAC (tanda dalam kebijakan): Ya

Kontrol akses berbasis atribut (ABAC) adalah strategi otorisasi yang menentukan izin berdasarkan atribut. Dalam AWS, atribut ini disebut tag. Anda dapat melampirkan tag ke entitas IAM (pengguna atau peran) dan ke banyak AWS sumber daya. Penandaan ke entitas dan sumber daya adalah langkah pertama dari ABAC. Kemudian rancanglah kebijakan ABAC untuk mengizinkan operasi ketika tanda milik prinsipal cocok dengan tanda yang ada di sumber daya yang ingin diakses.

ABAC sangat berguna di lingkungan yang berkembang dengan cepat dan berguna di situasi saat manajemen kebijakan menjadi rumit.

Untuk mengendalikan akses berdasarkan tanda, berikan informasi tentang tanda di elemen kondisi dari kebijakan menggunakan kunci kondisi `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, atau `aws:TagKeys`.

Jika sebuah layanan mendukung ketiga kunci kondisi untuk setiap jenis sumber daya, nilainya adalah Ya untuk layanan tersebut. Jika suatu layanan mendukung ketiga kunci kondisi untuk hanya beberapa jenis sumber daya, nilainya adalah Parsial.

Untuk informasi selengkapnya tentang ABAC, lihat Tentukan izin dengan otorisasi ABAC dalam Panduan Pengguna IAM. Untuk melihat tutorial yang menguraikan langkah-langkah pengaturan ABAC, lihat Menggunakan kontrol akses berbasis atribut (ABAC) dalam Panduan Pengguna IAM.

TODO Recovery Region Switch (Region switch) mendukung ABAC.

Menggunakan kredensi sementara dengan sakelar Wilayah

Mendukung kredensial sementara: Ya

Beberapa Layanan AWS tidak berfungsi saat Anda masuk menggunakan kredensil sementara. Untuk informasi tambahan, termasuk yang Layanan AWS bekerja dengan kredensi sementara, lihat Layanan AWS yang bekerja dengan IAM di Panduan Pengguna IAM.

Anda menggunakan kredensi sementara jika Anda masuk AWS Management Console menggunakan metode apa pun kecuali nama pengguna dan kata sandi. Misalnya, ketika Anda mengakses AWS menggunakan tautan masuk tunggal (SSO) perusahaan Anda, proses tersebut secara otomatis membuat kredensil sementara. Anda juga akan secara otomatis membuat kredensial sementara ketika Anda masuk ke konsol sebagai seorang pengguna lalu beralih peran. Untuk informasi selengkapnya tentang peralihan peran, lihat Beralih dari pengguna ke peran IAM (konsol) dalam Panduan Pengguna IAM.

Anda dapat membuat kredenal sementara secara manual menggunakan API AWS CLI atau AWS . Anda kemudian dapat menggunakan kredensi sementara tersebut untuk mengakses. AWS AWS merekomendasikan agar Anda secara dinamis menghasilkan kredensi sementara alih-alih menggunakan kunci akses jangka panjang. Untuk informasi selengkapnya, lihat Kredensial keamanan sementara di IAM.

Izin utama lintas layanan untuk sakelar Wilayah

Mendukung sesi akses maju (FAS): Ya

Saat Anda menggunakan entitas IAM (pengguna atau peran) untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Kebijakan memberikan izin kepada principal. Saat Anda menggunakan beberapa layanan, Anda mungkin melakukan tindakan yang kemudian memicu tindakan lain di layanan yang berbeda. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut.

Peran layanan untuk sakelar Wilayah

Mendukung peran layanan: Tidak

Peran layanan adalah peran IAM yang diambil oleh sebuah layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, mengubah, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat Buat sebuah peran untuk mendelegasikan izin ke Layanan AWS dalam Panduan pengguna IAM.

Peran terkait layanan untuk sakelar Wilayah

Mendukung peran terkait layanan: Tidak

Peran terkait layanan adalah jenis peran layanan yang ditautkan ke. Layanan AWS Layanan tersebut dapat menjalankan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Anda Akun AWS dan dimiliki oleh layanan. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.

Untuk detail tentang pembuatan atau manajemen peran terkait layanan, lihat Layanan AWS yang berfungsi dengan IAM. Cari layanan dalam tabel yang memiliki `Yes` di kolom Peran terkait layanan. Pilih tautan Ya untuk melihat dokumentasi peran terkait layanan untuk layanan tersebut.

Contoh kebijakan berbasis identitas untuk peralihan Wilayah di ARC

Secara default, pengguna dan peran tidak memiliki izin untuk membuat atau memodifikasi sumber daya ARC. Mereka juga tidak dapat melakukan tugas dengan menggunakan AWS Management

Console, AWS Command Line Interface (AWS CLI), atau AWS API. Untuk memberikan izin kepada pengguna untuk melakukan tindakan di sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM. Administrator kemudian dapat menambahkan kebijakan IAM ke peran, dan pengguna dapat mengambil peran.

Untuk mempelajari cara membuat kebijakan berbasis identitas IAM dengan menggunakan contoh dokumen kebijakan JSON ini, lihat Membuat kebijakan IAM (konsol) di Panduan Pengguna IAM.

Untuk detail tentang tindakan dan jenis sumber daya yang ditentukan oleh ARC, termasuk format ARNs untuk setiap jenis sumber daya, lihat Kunci tindakan, sumber daya, dan kondisi untuk Amazon Application Recovery Controller (ARC) di Referensi Otorisasi Layanan.

Topik

- Praktik terbaik kebijakan
- Merencanakan kebijakan kepercayaan peran eksekusi
- Izin akses penuh
- Izin baca-saja
- Izin blok eksekusi
- Akses sumber daya lintas akun
- Kebijakan peran eksekusi rencana lengkap

Praktik terbaik kebijakan

Kebijakan berbasis identitas menentukan apakah seseorang dapat membuat, mengakses, atau menghapus sumber daya ARC di akun Anda. Tindakan ini membuat Akun AWS Anda dikenai biaya. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

- Mulailah dengan kebijakan AWS terkelola dan beralih ke izin hak istimewa paling sedikit — Untuk mulai memberikan izin kepada pengguna dan beban kerja Anda, gunakan kebijakan AWS terkelola yang memberikan izin untuk banyak kasus penggunaan umum. Mereka tersedia di Anda Akun AWS. Kami menyarankan Anda mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola AWS pelanggan yang khusus untuk kasus penggunaan Anda. Untuk informasi selengkapnya, lihat Kebijakan yang dikelola AWS atau Kebijakan yang dikelola AWS untuk fungsi tugas dalam Panduan Pengguna IAM.
- Menerapkan izin dengan hak akses paling rendah – Ketika Anda menetapkan izin dengan kebijakan IAM, hanya berikan izin yang diperlukan untuk melakukan tugas. Anda melakukannya dengan mendefinisikan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi

tertentu, yang juga dikenal sebagai izin dengan hak akses paling rendah. Untuk informasi selengkapnya tentang cara menggunakan IAM untuk mengajukan izin, lihat Kebijakan dan izin dalam IAM dalam Panduan Pengguna IAM.

- Gunakan kondisi dalam kebijakan IAM untuk membatasi akses lebih lanjut – Anda dapat menambahkan suatu kondisi ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. Sebagai contoh, Anda dapat menulis kondisi kebijakan untuk menentukan bahwa semua permintaan harus dikirim menggunakan SSL. Anda juga dapat menggunakan ketentuan untuk memberikan akses ke tindakan layanan jika digunakan melalui yang spesifik Layanan AWS, seperti AWS CloudFormation. Untuk informasi selengkapnya, lihat Elemen kebijakan JSON IAM: Kondisi dalam Panduan Pengguna IAM.
- Gunakan IAM Access Analyzer untuk memvalidasi kebijakan IAM Anda untuk memastikan izin yang aman dan fungsional – IAM Access Analyzer memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan tersebut mematuhi bahasa kebijakan IAM (JSON) dan praktik terbaik IAM. IAM Access Analyzer menyediakan lebih dari 100 pemeriksaan kebijakan dan rekomendasi yang dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang aman dan fungsional. Untuk informasi selengkapnya, lihat Validasi kebijakan dengan IAM Access Analyzer dalam Panduan Pengguna IAM.
- Memerlukan otentikasi multi-faktor (MFA) - Jika Anda memiliki skenario yang mengharuskan pengguna IAM atau pengguna root di Anda, Akun AWS aktifkan MFA untuk keamanan tambahan. Untuk meminta MFA ketika operasi API dipanggil, tambahkan kondisi MFA pada kebijakan Anda. Untuk informasi selengkapnya, lihat Amankan akses API dengan MFA dalam Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang praktik terbaik dalam IAM, lihat Praktik terbaik keamanan di IAM dalam Panduan Pengguna IAM.

Merencanakan kebijakan kepercayaan peran eksekusi

Ini adalah kebijakan kepercayaan yang diperlukan untuk peran pelaksanaan rencana:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "arc-region-switch.amazonaws.com"
      },
```

```
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Izin akses penuh

Kebijakan IAM berikut memberikan akses penuh untuk semua sakelar Wilayah: APIs

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "arc-region-switch.amazonaws.com"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "arc-region-switch:CreatePlan",
        "arc-region-switch:UpdatePlan",
        "arc-region-switch:GetPlan",
        "arc-region-switch:ListPlans",
        "arc-region-switch:DeletePlan",
        "arc-region-switch:GetPlanInRegion",
        "arc-region-switch:ListPlansInRegion",
        "arc-region-switch:ApprovePlanExecutionStep",
        "arc-region-switch:GetPlanEvaluationStatus",
        "arc-region-switch:GetPlanExecution",
         "arc-region-switch:CancelPlanExecution",
        "arc-region-switch:ListRoute53HealthChecks",
        "arc-region-switch:ListPlanExecutions",
        "arc-region-switch:ListPlanExecutionEvents",
        "arc-region-switch:ListTagsForResource",
        "arc-region-switch:TagResource",
        "arc-region-switch:UntagResource",
        "arc-region-switch:UpdatePlanExecution",
        "arc-region-switch:UpdatePlanExecutionStep"
```

```
      ],
      "Resource": "*"
    }
  ]
}
```

Izin baca-saja

Kebijakan IAM berikut memberikan izin akses hanya-baca untuk peralihan Wilayah:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "arc-region-switch:GetPlan",
        "arc-region-switch:ListPlans",
        "arc-region-switch:GetPlanInRegion",
        "arc-region-switch:ListPlansInRegion",
        "arc-region-switch:GetPlanEvaluationStatus",
        "arc-region-switch:GetPlanExecution",
        "arc-region-switch:ListRoute53HealthChecks",
        "arc-region-switch:ListPlanExecutions",
        "arc-region-switch:ListPlanExecutionEvents",
        "arc-region-switch:ListTagsForResource"
      ],
      "Resource": "*"
    }
  ]
}
```

Izin blok eksekusi

Bagian berikut menyediakan kebijakan IAM untuk blok eksekusi tertentu yang Anda tambahkan ke paket peralihan Wilayah.

EC2 Blok eksekusi EC2 Auto Scaling Amazon

Kebijakan untuk peran eksekusi paket untuk mengelola grup EC2 Amazon EC2 Auto Scaling:

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

```
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "autoscaling:DescribeAutoScalingGroups"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "autoscaling:UpdateAutoScalingGroup"
      ],
      "Resource": [
        "arn:aws:autoscaling:us-
east-1:123456789012:autoScalingGroup:123d456e-123e-1111-abcd-
EXAMPLE22222:autoScalingGroupName/app-asg-primary",
        "arn:aws:autoscaling:us-
west-2:123456789012:autoScalingGroup:1234a321-123e-1234-aabb-
EXAMPLE33333:autoScalingGroupName/app-asg-secondary"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:GetMetricStatistics"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "cloudwatch:namespace": "AWS/AutoScaling"
        }
      }
    }
  ]
}
```

Blok eksekusi penskalaan sumber daya Amazon EKS

Kebijakan untuk peran eksekusi rencana untuk mengelola klaster Amazon EKS:

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```
    {
      "Effect": "Allow",
      "Action": [
        "eks:DescribeCluster"
      ],
      "Resource": [
        "arn:aws:eks:us-east-1:123456789012:cluster/app-eks-primary",
        "arn:aws:eks:us-west-2:123456789012:cluster/app-eks-secondary"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "eks:ListAssociatedAccessPolicies"
      ],
      "Resource": [
        "arn:aws:eks:us-east-1:123456789012:access-entry/app-eks-primary/*",
        "arn:aws:eks:us-west-2:123456789012:access-entry/app-eks-secondary/*"
      ]
    }
  ]
}
```

Catatan: Selain kebijakan IAM ini, peran eksekusi rencana perlu ditambahkan ke entri akses klaster Amazon EKS dengan kebijakan `AmazonArcRegionSwitchScalingPolicy` akses. Untuk informasi selengkapnya, lihat Konfigurasikan izin entri akses EKS.

Blok eksekusi penskalaan layanan Amazon ECS

Kebijakan untuk peran eksekusi rencana untuk mengelola layanan Amazon ECS:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecs:DescribeServices",
        "ecs:UpdateService"
      ],
      "Resource": [
        "arn:aws:ecs:us-east-1:123456789012:service/app-cluster-primary/app-service",
        "arn:aws:ecs:us-west-2:123456789012:service/app-cluster-secondary/app-service"
```

```
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ecs:DescribeClusters"
      ],
      "Resource": [
        "arn:aws:ecs:us-east-1:123456789012:cluster/app-cluster-primary",
        "arn:aws:ecs:us-west-2:123456789012:cluster/app-cluster-secondary"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ecs:ListServices"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "application-autoscaling:DescribeScalableTargets",
        "application-autoscaling:RegisterScalableTarget"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:GetMetricStatistics"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "cloudwatch:namespace": "ECS/ContainerInsights"
        }
      }
    }
  ]
}
```

Routing ARC mengontrol blok eksekusi

Catatan: Blok eksekusi kontrol perutean Amazon ARC mengharuskan Kebijakan Kontrol Layanan (SCPs) apa pun yang diterapkan pada peran eksekusi paket memungkinkan akses ke Wilayah berikut untuk layanan ini:

- `route53-recovery-control-config: us-west-2`
- `route53-recovery-cluster: us-west-2, us-east-1, eu-west-1, ap-southeast-2, ap-northeast-1`

Kebijakan untuk peran eksekusi rencana untuk mengelola kontrol perutean ARC:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "route53-recovery-control-config:DescribeControlPanel",
        "route53-recovery-control-config:DescribeCluster"
      ],
      "Resource": [
        "arn:aws:route53-recovery-control::123456789012:controlpanel/
abcd1234abcd1234abcd1234abcd1234",
        "arn:aws:route53-recovery-control::123456789012:cluster/4b325d3b-0e28-4dcf-
ba4a-EXAMPLE11111"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "route53-recovery-cluster:GetRoutingControlState",
        "route53-recovery-cluster:UpdateRoutingControlStates"
      ],
      "Resource": [
        "arn:aws:route53-recovery-
control::123456789012:controlpanel/1234567890abcdef1234567890abcdef/routingcontrol/
abcdef1234567890",
        "arn:aws:route53-recovery-
control::123456789012:controlpanel/1234567890abcdef1234567890abcdef/
routingcontrol/1234567890abcdef"
      ]
```

```
    }
  ]
}
```

Anda dapat mengambil ID panel kontrol kontrol routing dan ID Cluster menggunakan CLI. Untuk informasi selengkapnya, lihat Siapkan komponen kontrol perutean.

Blok eksekusi Basis Data Aurora Global

Kebijakan untuk peran pelaksanaan rencana untuk mengelola database global Aurora:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "rds:DescribeGlobalClusters",
        "rds:DescribeDBClusters"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "rds:FailoverGlobalCluster",
        "rds:SwitchoverGlobalCluster"
      ],
      "Resource": [
        "arn:aws:rds:us-east-1:123456789012:global-cluster:app-global-db",
        "arn:aws:rds:us-east-1:123456789012:cluster:app-db-primary",
        "arn:aws:rds:us-west-2:123456789012:cluster:app-db-secondary"
      ]
    }
  ]
}
```

Blok eksekusi persetujuan manual

Kebijakan untuk peran yang dapat menyetujui langkah-langkah manual:

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "arc-region-switch:ApprovePlanExecutionStep"
      ],
      "Resource": "arn:aws:arc-region-switch::123456789012:plan/sample-plan:0fba5e"
    }
  ]
}
```

Tindakan kustom Blok eksekusi Lambda

Kebijakan untuk peran eksekusi rencana untuk menjalankan fungsi Lambda:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "lambda:GetFunction",
        "lambda:InvokeFunction"
      ],
      "Resource": [
        "arn:aws:lambda:us-east-1:123456789012:function:app-recovery-primary",
        "arn:aws:lambda:us-west-2:123456789012:function:app-recovery-secondary"
      ]
    }
  ]
}
```

Blok eksekusi pemeriksaan kesehatan Route 53

Kebijakan untuk peran pelaksanaan rencana untuk menggunakan pemeriksaan kesehatan Rute 53:

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

```
      "Action": [
        "route53:ListResourceRecordSets"
      ],
      "Resource": [
        "arn:aws:route53:::hostedzone/Z1234567890ABCDEFGHIJ"
      ]
    }
  ]
}
```

Blok eksekusi rencana sakelar wilayah

Kebijakan untuk peran pelaksanaan rencana untuk melaksanakan rencana anak:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "arc-region-switch:StartPlanExecution",
        "arc-region-switch:GetPlanExecution",
        "arc-region-switch:CancelPlanExecution",
        "arc-region-switch:UpdatePlanExecution",
        "arc-region-switch:ListPlanExecutions"
      ],
      "Resource": [
        "arn:aws:arc-region-switch::123456789012:plan/child-plan-1:50c1a1",
        "arn:aws:arc-region-switch::123456789012:plan/child-plan-2:d1e5e1"
      ]
    }
  ]
}
```

CloudWatch alarm untuk kesehatan aplikasi

Kebijakan untuk peran pelaksanaan rencana untuk mengakses CloudWatch alarm untuk kesehatan aplikasi, yang digunakan untuk membantu menentukan waktu pemulihan yang sebenarnya:

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:DescribeAlarmHistory",
        "cloudwatch:DescribeAlarms"
      ],
      "Resource": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:app-health-primary",
        "arn:aws:cloudwatch:us-west-2:123456789012:alarm:app-health-secondary"
      ]
    }
  ]
}
```

Akses sumber daya lintas akun

Jika sumber daya ada di akun yang berbeda, Anda memerlukan peran lintas akun. Berikut contoh kebijakan kepercayaan untuk peran lintas akun:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:role/RegionSwitchExecutionRole"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "sts:ExternalId": "UniqueExternalId123"
        }
      }
    }
  ]
}
```

Dan izin untuk peran eksekusi rencana untuk mengambil peran lintas akun ini:

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```
    {
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Resource": "arn:aws:iam::987654321098:role/RegionSwitchCrossAccountRole",
      "Condition": {
        "StringEquals": {
          "sts:ExternalId": "UniqueExternalId123"
        }
      }
    }
  ]
}
```

Kebijakan peran eksekusi rencana lengkap

Kebijakan komprehensif yang mencakup izin untuk semua blok eksekusi akan cukup besar. Dalam praktiknya, Anda hanya harus menyertakan izin untuk blok eksekusi yang Anda gunakan dalam paket spesifik Anda. Berikut ini contoh kebijakan:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iam:SimulatePrincipalPolicy",
      "Resource": "arn:aws:iam::123456789012:role/RegionSwitchExecutionRole"
    },
    {
      "Effect": "Allow",
      "Action": [
        "arc-region-switch:GetPlan",
        "arc-region-switch:GetPlanExecution",
        "arc-region-switch:ListPlanExecutions"
      ],
      "Resource": "*"
    },
    // Include additional statements for specific execution blocks here
  ]
}
```

Ingatlah untuk menyertakan hanya izin yang diperlukan untuk blok eksekusi tertentu yang Anda gunakan dalam paket Anda, untuk mengikuti prinsip hak istimewa paling sedikit.

Pencatatan dan pemantauan untuk sakelar Wilayah di ARC

Anda dapat menggunakan Amazon CloudWatch, AWS CloudTrail, dan Amazon EventBridge untuk memantau sakelar Wilayah di Amazon Application Recovery Controller (ARC), untuk mendapatkan peringatan, menganalisis pola, dan membantu memecahkan masalah.

Topik

Logging Region beralih panggilan API menggunakan AWS CloudTrail

Sakelar Wilayah Amazon Application Recovery Controller (ARC) terintegrasi dengan AWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan di ARC. CloudTrail menangkap semua panggilan API untuk ARC sebagai peristiwa. Panggilan yang diambil termasuk panggilan dari konsol ARC dan panggilan kode ke operasi ARC API.

Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail acara secara berkelanjutan ke bucket Amazon S3, termasuk acara untuk ARC. Jika Anda tidak mengonfigurasi jejak, Anda masih dapat melihat peristiwa terbaru di CloudTrail konsol dalam Riwayat acara.

Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat untuk ARC, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan.

Untuk mempelajari selengkapnya CloudTrail, lihat Panduan AWS CloudTrail Pengguna.

Informasi ARC di CloudTrail

CloudTrail diaktifkan pada Akun AWS saat Anda membuat akun. Ketika aktivitas terjadi di ARC, aktivitas tersebut dicatat dalam suatu CloudTrail peristiwa bersama dengan peristiwa AWS layanan lainnya dalam riwayat Acara. Anda dapat melihat, mencari, dan mengunduh acara terbaru di situs Anda Akun AWS. Untuk informasi selengkapnya, lihat Bekerja dengan riwayat CloudTrail Acara.

Untuk catatan acara yang sedang berlangsung di Anda Akun AWS, termasuk acara untuk ARC, buat jejak. Jejak memungkinkan CloudTrail untuk mengirimkan file log ke bucket Amazon S3. Secara default, saat Anda membuat jejak di konsol, jejak tersebut berlaku untuk semua Wilayah AWS. Jejak mencatat peristiwa dari semua Wilayah di AWS partisi dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat mengonfigurasi AWS layanan lain untuk

menganalisis lebih lanjut dan menindaklanjuti data peristiwa yang dikumpulkan dalam CloudTrail log. Untuk informasi selengkapnya, lihat berikut:

- Gambaran umum untuk membuat jejak
- CloudTrail layanan dan integrasi yang didukung
- Mengonfigurasi notifikasi Amazon SNS untuk CloudTrail
- Menerima file CloudTrail log dari beberapa Wilayah dan Menerima file CloudTrail log dari beberapa akun

Semua tindakan ARC dicatat oleh CloudTrail dan didokumentasikan dalam TBD API REFERENCE LINK. Misalnya, panggilan keTBD, TBD dan TBD tindakan menghasilkan entri dalam file CloudTrail log.

Setiap entri peristiwa atau log berisi informasi tentang entitas yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan hal berikut ini:

- Apakah permintaan itu dibuat dengan kredenal pengguna root atau AWS Identity and Access Management (IAM).
- Apakah permintaan tersebut dibuat dengan kredensial keamanan sementara untuk satu peran atau pengguna gabungan.
- Apakah permintaan itu dibuat oleh AWS layanan lain.

Untuk informasi selengkapnya, lihat Elemen userIdentity CloudTrail .

Melihat peristiwa peralihan Wilayah dalam riwayat acara

CloudTrail memungkinkan Anda melihat peristiwa terbaru dalam riwayat Acara. Sebagian besar peristiwa untuk permintaan API peralihan Wilayah berada di Wilayah tempat Anda bekerja dengan paket peralihan Wilayah, misalnya, tempat Anda membuat rencana atau menjalankan rencana. Namun, beberapa tindakan peralihan Wilayah yang Anda jalankan di konsol ARC dibuat menggunakan operasi API rencana kontrol, bukan operasi bidang data. Untuk operasi pesawat kontrol, Anda melihat peristiwa di AS Timur (Virginia N.). Untuk mempelajari tentang panggilan API mana yang merupakan operasi bidang kontrol, lihat Operasi API sakelar wilayah.

Memahami entri file log ARC

Trail adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai file log ke bucket Amazon S3 yang Anda tentukan. CloudTrail file log berisi satu atau lebih entri log. Peristiwa mewakili

permintaan tunggal dari sumber mana pun dan mencakup informasi tentang tindakan yang diminta, tanggal dan waktu tindakan, parameter permintaan, dan sebagainya. CloudTrail file log bukanlah jejak tumpukan yang diurutkan dari panggilan API publik, jadi file tersebut tidak muncul dalam urutan tertentu.

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan `StartPlanExecution` tindakan untuk beralih Wilayah.


```
{
    "eventVersion": "1.11",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "A1B2C3D4E5F6G7EXAMPLE",
        "arn": "arn:aws:iam::111122223333:role/admin",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROA33L3W36EXAMPLE",
                "arn": "arn:aws:iam::111122223333:role/admin",
                "accountId": "111122223333",
                "userName": "EXAMPLENAME"
            },
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2025-07-06T17:38:05Z"
            }
        }
    },
    "eventTime": "2025-07-06T18:08:03Z",
    "eventSource": "arc-region-switch.amazonaws.com",
    "eventName": "StartPlanExecution",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.50",
    "userAgent": "Boto3/1.17.101 Python/3.8.10 Linux/4.14.231-180.360.amzn2.x86_64
 exec-env/AWS_Lambda_python3.8 Botocore/1.20.102",
    "requestParameters": {
        "planArn": "arn:aws:arc-region-switch::555555555555:plan/
CloudTrailIntegTestPlan:bbbbb",
        "targetRegion": "us-east-1",
        "action": "activate"    }
    "responseElements": {
```

```
        "executionId": "us-east-1/ddddddddEXAMPLE",
        "plan": "arn:aws:arc-region-switch::555555555555:plan/
CloudTrailIntegTestPlan:bbbbb",
        "planVersion": "1",
        "activateRegion": "us-east-1"    },
    "requestID": "fd42dcf7-6446-41e9-b408-d096example",
    "eventID": "4b5c42df-1174-46c8-be99-d67aexample",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management",
      "tlsDetails": {
        "tlsVersion": "TLSv1.3",
        "cipherSuite": "TLS_AES_128_GCM_SHA256",
        "clientProvidedHostHeader": "us-east-1.arc.amazon.aws"
}
```


Menggunakan sakelar Wilayah di ARC dengan Amazon EventBridge

Menggunakan Amazon EventBridge, Anda dapat menyiapkan aturan berbasis peristiwa yang memantau sumber daya peralihan Wilayah Anda di Amazon Application Recovery Controller (ARC), lalu memulai tindakan target yang menggunakan layanan lain. AWS Misalnya, Anda dapat menetapkan aturan untuk mengirimkan notifikasi email dengan memberi sinyal topik Amazon SNS setiap kali rencana peralihan Wilayah menyelesaikan eksekusi.

Anda dapat membuat aturan di Amazon EventBridge untuk menindaklanjuti peristiwa peralihan Wilayah ARC berikut:

- Eksekusi rencana peralihan wilayah. Acara menentukan bahwa rencana peralihan Wilayah telah dijalankan (dieksekusi).
- Evaluasi rencana peralihan wilayah. Acara ini menetapkan bahwa evaluasi rencana peralihan Wilayah telah selesai.

Untuk menangkap peristiwa ARC tertentu yang Anda minati, tentukan pola khusus peristiwa yang EventBridge dapat digunakan untuk mendeteksi peristiwa. Pola acara memiliki struktur yang sama dengan peristiwa yang cocok. Pola mengutip bidang yang ingin Anda cocokkan dan memberikan nilai yang Anda cari.

Peristiwa dipancarkan atas dasar upaya terbaik. Mereka dikirim dari ARC ke hampir real-time EventBridge dalam keadaan operasional normal. Namun, situasi dapat muncul yang mungkin menunda atau mencegah pengiriman suatu peristiwa.

Untuk informasi tentang cara kerja EventBridge aturan dengan pola peristiwa, lihat Peristiwa dan Pola Peristiwa di EventBridge.

Pantau sumber daya sakelar Wilayah dengan EventBridge

Dengan EventBridge, Anda dapat membuat aturan yang menentukan tindakan yang akan diambil saat ARC memancarkan peristiwa untuk sumber daya peralihan Wilayah.

Untuk mengetik atau menyalin dan menempelkan pola acara ke EventBridge konsol, di konsol, pilih opsi Masukkan opsi saya sendiri. Untuk membantu Anda menentukan pola peristiwa yang mungkin berguna bagi Anda, topik ini mencakup contoh pola peralihan Wilayah.

Untuk membuat aturan untuk peristiwa sumber daya

1. Buka EventBridge konsol Amazon di https://console.aws.amazon.com/events/.
2. Wilayah AWS Untuk membuat aturan, pilih Wilayah tempat Anda membuat paket yang ingin Anda pantau acara.
3. Pilih Buat aturan.
4. Masukkan Nama untuk aturan tersebut, dan, secara opsional, deskripsi.
5. Untuk bus Acara, biarkan nilai default, default.
6. Pilih Berikutnya.
7. Untuk langkah pola acara Build, untuk sumber Event, tinggalkan nilai default, AWS peristiwa.
8. Di bawah Contoh acara, pilih Masukkan milik saya.
9. Untuk contoh peristiwa, ketik atau salin dan tempel pola acara. Sebagai contoh, lihat bagian selanjutnya.

Contoh pola sakelar Wilayah

Pola acara memiliki struktur yang sama dengan peristiwa yang cocok. Pola mengutip bidang yang ingin Anda cocokkan dan memberikan nilai yang Anda cari.

Anda dapat menyalin dan menempelkan pola peristiwa dari bagian ini EventBridge ke dalam untuk membuat aturan yang dapat Anda gunakan untuk memantau tindakan dan sumber daya ARC.

Pola peristiwa berikut memberikan contoh yang mungkin Anda gunakan EventBridge untuk kemampuan sakelar Wilayah di ARC.

- Pilih semua acara dari Region switch untuk PlanExecution.

```
{
    "source": [ "aws.arc-region-switch" ],
    "detail-type": [ "ARC Region switch Plan Execution" ]
}
```

- Pilih semua acara dari Region switch untuk PlanEvaluation.

```
     {
     "source": [ "aws.arc-region-switch" ],
     "detail-type": [ "ARC Region Switch Plan Evaluation" ]
     }
```

Berikut ini adalah contoh peristiwa ARC untuk eksekusi rencana peralihan Wilayah:

```
{
   "version": "0",
   "id": "1111111-bbbb-aaaa-cccc-dddddEXAMPLE", # Random uuid
   "detail-type": "ARC Region Switch Plan Execution",
   "source": "aws.arc-region-switch",
   "account": "111122223333",
   "time": "2023-11-16T23:38:14Z",
   "region": "us-east-1",
   "resources": ["arn:aws:arc-region-switch::111122223333:plan/aaaaaExample"], #
 planArn
   "detail": {
    "version": "0.0.1",
    "eventType": "ExecutionStarted",
    "executionId": "bbbbbbbEXAMPLE",
    "executionAction": "activating/deactivating {region}",
    "idempotencyKey": "1111111-2222-3333-4444-5555555555", # As there is a possibility
 of dual logging
   }
}
```

Berikut ini adalah contoh peristiwa ARC untuk eksekusi level langkah rencana peralihan Wilayah:

```
{
   "version": "0",
   "id": "1111111-bbbb-aaaa-cccc-dddddEXAMPLE", # Random uuid
   "detail-type": "ARC Region Switch Plan Execution",
   "source": "aws.arc-region-switch",
   "account": "111122223333",
   "time": "2023-11-16T23:38:14Z",
   "region": "us-east-1",
   "resources": ["arn:aws:arc-region-switch::111122223333:plan/aaaaaExample"], #
 planArn
   "detail": {
    "version": "0.0.1",
    "eventType": "StepStarted",
    "executionId": "bbbbbbEXAMPLE",
    "executionAction": "activating/deactivating {region}",
    "idempotencyKey": "1111111-2222-3333-4444-5555555555", # As there is a possibility
 of dual logging
    "stepDetails" : {
     "stepName": "Routing control step",
     "resource": ["arn:aws:route53-recovery-control::111122223333:controlpanel/
abcdefghiEXAMPLE/routingcontrol/jklmnopqrsEXAMPLE"]
    }
   }
}
```

Berikut ini adalah contoh peristiwa ARC untuk peringatan evaluasi rencana peralihan Wilayah.

Untuk evaluasi rencana peralihan Wilayah, peristiwa akan dipancarkan saat peringatan dikembalikan. Jika peringatan tidak dihapus, sebuah peristiwa dipancarkan untuk peringatan hanya sekali setiap 24 jam. Ketika acara dihapus, tidak ada peristiwa lebih lanjut yang dipancarkan untuk peringatan itu.

```
{
   "version": "0",
   "id": "05d4d2d5-9c76-bfea-72d2-d4614802adb4", # Random uuid
   "detail-type": "ARC Region Switch Plan Execution",
   "source": "aws.arc-region-switch",
   "account": "111122223333",
   "time": "2023-11-16T23:38:14Z",
   "region": "us-east-1",
   "resources": ["arn:aws:arc-region-switch::111122223333:plan/a2b89be4821bfd1d"],
   "detail": {
     "version": "0.0.1",
     "idempotencyKey": "1111111-2222-3333-4444-5555555555",
```

```
     "metadata": {
        "evaluationTime" : "timestamp",
        "warning" : "There is a plan evaluation warning for arn:aws:arc-region-
switch::111122223333:plan/a2b89be4821bfd1d. Navigate to the Region switch console to
 resolve."
     }
   }
}
```

Tentukan grup CloudWatch log yang akan digunakan sebagai target

Saat membuat EventBridge aturan, Anda harus menentukan target tempat peristiwa yang cocok dengan aturan dikirim. Untuk daftar target yang tersedia EventBridge, lihat Target yang tersedia di EventBridge konsol. Salah satu target yang dapat Anda tambahkan ke EventBridge aturan adalah grup CloudWatch log Amazon. Bagian ini menjelaskan persyaratan untuk menambahkan grup CloudWatch log sebagai target, dan menyediakan prosedur untuk menambahkan grup log saat Anda membuat aturan.

Untuk menambahkan grup CloudWatch log sebagai target, Anda dapat melakukan salah satu hal berikut:

- Buat grup log baru
- Pilih grup log yang ada

Jika Anda menentukan grup log baru menggunakan konsol saat membuat aturan, EventBridge secara otomatis membuat grup log untuk Anda. Pastikan grup log yang Anda gunakan sebagai target EventBridge aturan dimulai dengan`/aws/events`. Jika Anda ingin memilih grup log yang ada, ketahuilah bahwa hanya grup log yang dimulai dengan yang `/aws/events` muncul sebagai opsi di menu tarik-turun. Untuk informasi selengkapnya, lihat Membuat grup log baru di Panduan CloudWatch Pengguna Amazon.

Jika Anda membuat atau menggunakan grup CloudWatch log untuk digunakan sebagai target menggunakan CloudWatch operasi di luar konsol, pastikan Anda menetapkan izin dengan benar. Jika Anda menggunakan konsol untuk menambahkan grup log ke EventBridge aturan, maka kebijakan berbasis sumber daya untuk grup log diperbarui secara otomatis. Namun, jika Anda menggunakan AWS Command Line Interface atau AWS SDK untuk menentukan grup log, Anda harus memperbarui kebijakan berbasis sumber daya untuk grup log. Contoh kebijakan berikut menggambarkan izin yang harus Anda tentukan dalam kebijakan berbasis sumber daya untuk grup log:

```
{
  "Statement": [
    {
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "events.amazonaws.com",
          "delivery.logs.amazonaws.com"
        ]
      },
      "Resource": "arn:aws:logs:region:account:log-group:/aws/events/*:*",
      "Sid": "TrustEventsToStoreLogEvent"
    }
  ],
  "Version": "2012-10-17"
}
```

Anda tidak dapat mengonfigurasi kebijakan berbasis sumber daya untuk grup log menggunakan konsol. Untuk menambahkan izin yang diperlukan ke kebijakan berbasis sumber daya, gunakan operasi API. CloudWatch PutResourcePolicy Kemudian, Anda dapat menggunakan perintah describe-resource-policiesCLI untuk memeriksa apakah kebijakan Anda diterapkan dengan benar.

Untuk membuat aturan untuk acara sumber daya dan menentukan target grup CloudWatch log

1. Buka EventBridge konsol Amazon di https://console.aws.amazon.com/events/.
2. Pilih aturan Wilayah AWS yang ingin Anda buat.
3. Pilih Buat aturan lalu masukkan informasi apa pun tentang aturan itu, seperti pola acara atau detail jadwal.

   Untuk informasi selengkapnya tentang membuat EventBridge aturan untuk kesiapan, lihat Memantau sumber daya pemeriksaan kesiapan dengan. EventBridge
4. Pada halaman Pilih target, pilih CloudWatchsebagai target Anda.
5. Pilih grup CloudWatch log dari menu tarik-turun.

Kuota untuk sakelar Wilayah

Sakelar wilayah di Amazon Application Recovery Controller (ARC) tunduk pada kuota berikut.

Entitas	Kuota
Jumlah paket per akun	10
Jumlah blok eksekusi per rencana	100
Jumlah blok eksekusi rencana peralihan Wilayah per rencana	25
Jumlah CloudWatch alarm per kondisi pemicu	10

Contoh kode untuk Application Recovery Controller menggunakan AWS SDKs

Contoh kode berikut menunjukkan cara menggunakan Application Recovery Controller dengan AWS software development kit (SDK).

Tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Contoh kode

- Contoh dasar untuk Application Recovery Controller menggunakan AWS SDKs
  - Tindakan untuk Application Recovery Controller menggunakan AWS SDKs
    - Gunakan GetRoutingControlState dengan AWS SDK
    - Gunakan UpdateRoutingControlState dengan AWS SDK

Contoh dasar untuk Application Recovery Controller menggunakan AWS SDKs

Contoh kode berikut menunjukkan cara menggunakan dasar-dasar Amazon Route 53 Application Recovery Controller dengan AWS SDKs.

Contoh

- Tindakan untuk Application Recovery Controller menggunakan AWS SDKs
  - Gunakan GetRoutingControlState dengan AWS SDK
  - Gunakan UpdateRoutingControlState dengan AWS SDK

Tindakan untuk Application Recovery Controller menggunakan AWS SDKs

Contoh kode berikut menunjukkan bagaimana melakukan tindakan Application Recovery Controller individual dengan AWS SDKs. Setiap contoh menyertakan tautan ke GitHub, di mana Anda dapat menemukan instruksi untuk mengatur dan menjalankan kode.

Contoh berikut hanya mencakup tindakan yang paling umum digunakan. Untuk daftar lengkapnya, lihat Referensi API Pengontrol Pemulihan Aplikasi Amazon Route 53.

Contoh

- Gunakan GetRoutingControlState dengan AWS SDK
- Gunakan UpdateRoutingControlState dengan AWS SDK

Gunakan **`GetRoutingControlState`** dengan AWS SDK

Contoh kode berikut menunjukkan cara menggunakan`GetRoutingControlState`.

Java

SDK untuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
    public static GetRoutingControlStateResponse
 getRoutingControlState(List<ClusterEndpoint> clusterEndpoints,
            String routingControlArn) {
        // As a best practice, we recommend choosing a random cluster endpoint to
 get or
        // set routing control states.
        // For more information, see
        // https://docs.aws.amazon.com/r53recovery/latest/dg/route53-arc-best-
practices.html#route53-arc-best-practices.regional
        Collections.shuffle(clusterEndpoints);
        for (ClusterEndpoint clusterEndpoint : clusterEndpoints) {
            try {
```

```
                System.out.println(clusterEndpoint);
                Route53RecoveryClusterClient client =
 Route53RecoveryClusterClient.builder()
                        .endpointOverride(URI.create(clusterEndpoint.endpoint()))
                        .region(Region.of(clusterEndpoint.region())).build();
                return client.getRoutingControlState(
                        GetRoutingControlStateRequest.builder()
                                .routingControlArn(routingControlArn).build());
            } catch (Exception exception) {
                System.out.println(exception);
            }
        }
        return null;
    }
```

- Untuk detail API, lihat GetRoutingControlStatedi Referensi AWS SDK for Java 2.x API.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
import boto3


def create_recovery_client(cluster_endpoint):
    """
    Creates a Boto3 Route 53 Application Recovery Controller client for the
 specified
    cluster endpoint URL and AWS Region.

    :param cluster_endpoint: The cluster endpoint URL and Region.
    :return: The Boto3 client.
    """
    return boto3.client(
        "route53-recovery-cluster",
```

```
        endpoint_url=cluster_endpoint["Endpoint"],
        region_name=cluster_endpoint["Region"],
    )



def get_routing_control_state(routing_control_arn, cluster_endpoints):
    """
    Gets the state of a routing control. Cluster endpoints are tried in
    sequence until the first successful response is received.

    :param routing_control_arn: The ARN of the routing control to look up.
    :param cluster_endpoints: The list of cluster endpoints to query.
    :return: The routing control state response.
    """

    # As a best practice, we recommend choosing a random cluster endpoint to get
 or set routing control states.
    # For more information, see https://docs.aws.amazon.com/r53recovery/latest/
dg/route53-arc-best-practices.html#route53-arc-best-practices.regional
    random.shuffle(cluster_endpoints)
    for cluster_endpoint in cluster_endpoints:
        try:
            recovery_client = create_recovery_client(cluster_endpoint)
            response = recovery_client.get_routing_control_state(
                RoutingControlArn=routing_control_arn
            )
            return response
        except Exception as error:
            print(error)
            raise error
```

- Untuk detail API, lihat GetRoutingControlStatedi AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **`UpdateRoutingControlState`** dengan AWS SDK

Contoh kode berikut menunjukkan cara menggunakan`UpdateRoutingControlState`.

Java

SDK untuk Java 2.x

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
    public static UpdateRoutingControlStateResponse
 updateRoutingControlState(List<ClusterEndpoint> clusterEndpoints,
            String routingControlArn,
            String routingControlState) {
        // As a best practice, we recommend choosing a random cluster endpoint to
 get or
        // set routing control states.
        // For more information, see
        // https://docs.aws.amazon.com/r53recovery/latest/dg/route53-arc-best-
practices.html#route53-arc-best-practices.regional
        Collections.shuffle(clusterEndpoints);
        for (ClusterEndpoint clusterEndpoint : clusterEndpoints) {
            try {
                System.out.println(clusterEndpoint);
                Route53RecoveryClusterClient client =
 Route53RecoveryClusterClient.builder()
                        .endpointOverride(URI.create(clusterEndpoint.endpoint()))
                        .region(Region.of(clusterEndpoint.region()))
                        .build();
                return client.updateRoutingControlState(
                        UpdateRoutingControlStateRequest.builder()

 .routingControlArn(routingControlArn).routingControlState(routingControlState).build());
            } catch (Exception exception) {
                System.out.println(exception);
            }
        }
        return null;
```

```
    }
```

- Untuk detail API, lihat UpdateRoutingControlStatedi Referensi AWS SDK for Java 2.x API.

Python

SDK untuk Python (Boto3)

> **Note**
>
> Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara pengaturan dan menjalankannya di Repositori Contoh Kode AWS.

```
import boto3


def create_recovery_client(cluster_endpoint):
    """
    Creates a Boto3 Route 53 Application Recovery Controller client for the
 specified
    cluster endpoint URL and AWS Region.

    :param cluster_endpoint: The cluster endpoint URL and Region.
    :return: The Boto3 client.
    """
    return boto3.client(
        "route53-recovery-cluster",
        endpoint_url=cluster_endpoint["Endpoint"],
        region_name=cluster_endpoint["Region"],
    )



def update_routing_control_state(
    routing_control_arn, cluster_endpoints, routing_control_state
):
    """
    Updates the state of a routing control. Cluster endpoints are tried in
    sequence until the first successful response is received.
```

```
    :param routing_control_arn: The ARN of the routing control to update the
 state for.
    :param cluster_endpoints: The list of cluster endpoints to try.
    :param routing_control_state: The new routing control state.
    :return: The routing control update response.
    """

    # As a best practice, we recommend choosing a random cluster endpoint to get
 or set routing control states.
    # For more information, see https://docs.aws.amazon.com/r53recovery/latest/
dg/route53-arc-best-practices.html#route53-arc-best-practices.regional
    random.shuffle(cluster_endpoints)
    for cluster_endpoint in cluster_endpoints:
        try:
            recovery_client = create_recovery_client(cluster_endpoint)
            response = recovery_client.update_routing_control_state(
                RoutingControlArn=routing_control_arn,
                RoutingControlState=routing_control_state,
            )
            return response
        except Exception as error:
            print(error)
```

- Untuk detail API, lihat UpdateRoutingControlStatedi AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihatMenggunakan layanan ini dengan AWS SDK. Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Keamanan di Pengontrol Pemulihan Aplikasi Amazon

Keamanan cloud di AWS adalah prioritas tertinggi. Sebagai AWS pelanggan, Anda mendapat manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan organisasi yang paling sensitif terhadap keamanan.

Keamanan adalah tanggung jawab bersama antara Anda AWS dan Anda. Model tanggung jawab bersama menjelaskan hal ini sebagai keamanan dari cloud dan keamanan dalam cloud:

- Keamanan cloud — AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan AWS layanan di AWS Cloud. AWS juga memberi Anda layanan yang dapat Anda gunakan dengan aman. Auditor pihak ketiga secara teratur menguji dan memverifikasi efektivitas keamanan kami sebagai bagian dari Program AWS Kepatuhan Program AWS Kepatuhan . Untuk mempelajari tentang program kepatuhan yang berlaku untuk Amazon Application Recovery Controller, lihat AWS Layanan dalam Lingkup oleh AWS Layanan Program Kepatuhan .
- Keamanan di cloud — Tanggung jawab Anda ditentukan oleh AWS layanan yang Anda gunakan. Anda juga bertanggung jawab atas faktor lain, yang mencakup sensitivitas data Anda, persyaratan perusahaan Anda, serta undang-undang dan peraturan yang berlaku.

Dokumentasi ini membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan ARC. Topik berikut menunjukkan cara mengonfigurasi ARC untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga mempelajari cara menggunakan AWS layanan lain yang membantu Anda memantau dan mengamankan sumber daya ARC Anda.

Topik

Perlindungan data di Amazon Application Recovery Controller

Model tanggung jawab AWS bersama model tanggung berlaku untuk perlindungan data di Amazon Application Recovery Controller. Seperti yang dijelaskan dalam model AWS ini, bertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS Cloud. Anda bertanggung jawab untuk mempertahankan kendali atas konten yang di-host pada infrastruktur ini. Anda juga bertanggung jawab atas tugas-tugas konfigurasi dan manajemen keamanan untuk Layanan AWS yang Anda gunakan. Lihat informasi yang lebih lengkap tentang privasi data dalam Pertanyaan Umum Privasi Data. Lihat informasi tentang perlindungan data di Eropa di pos blog Model Tanggung Jawab Bersama dan GDPR AWS di Blog Keamanan AWS .

Untuk tujuan perlindungan data, kami menyarankan Anda melindungi Akun AWS kredensil dan mengatur pengguna individu dengan AWS IAM Identity Center atau AWS Identity and Access Management (IAM). Dengan cara itu, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tanggung jawab tugasnya. Kami juga menyarankan supaya Anda mengamankan data dengan cara-cara berikut:

- Gunakan autentikasi multi-faktor (MFA) pada setiap akun.
- Gunakan SSL/TLS untuk berkomunikasi dengan AWS sumber daya. Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Siapkan API dan pencatatan aktivitas pengguna dengan AWS CloudTrail. Untuk informasi tentang penggunaan CloudTrail jejak untuk menangkap AWS aktivitas, lihat Bekerja dengan CloudTrail jejak di AWS CloudTrail Panduan Pengguna.
- Gunakan solusi AWS enkripsi, bersama dengan semua kontrol keamanan default di dalamnya Layanan AWS.
- Gunakan layanan keamanan terkelola tingkat lanjut seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.
- Jika Anda memerlukan modul kriptografi tervalidasi FIPS 140-3 saat mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Lihat informasi selengkapnya tentang titik akhir FIPS yang tersedia di Standar Pemrosesan Informasi Federal (FIPS) 140-3.

Kami sangat merekomendasikan agar Anda tidak pernah memasukkan informasi identifikasi yang sensitif, seperti nomor rekening pelanggan Anda, ke dalam tanda atau bidang isian bebas seperti bidang Nama. Ini termasuk saat Anda bekerja dengan ARC atau lainnya Layanan AWS menggunakan konsol, API AWS CLI, atau AWS SDKs. Data apa pun yang Anda masukkan ke dalam tanda atau bidang isian bebas yang digunakan untuk nama dapat digunakan untuk log penagihan

atau log diagnostik. Saat Anda memberikan URL ke server eksternal, kami sangat menganjurkan supaya Anda tidak menyertakan informasi kredensial di dalam URL untuk memvalidasi permintaan Anda ke server itu.

Enkripsi diam

Informasi konfigurasi pelanggan disimpan dalam tabel global Amazon DynamoDB milik layanan, dan dienkripsi saat istirahat.

Kumpulan data yang berisi status sel dalam cluster ARC ditulis ke volume Amazon EBS untuk cadangan. ARC menggunakan enkripsi Amazon EBS default saat data dalam keadaan diam.

Enkripsi bergerak

Permintaan dan tanggapan pelanggan—untuk konfigurasi ARC, kueri status kesiapan, pembaruan status sel, dan sebagainya—dienkripsi selama transportasi di seluruh layanan dengan menggunakan TLS.

Identity and Access Management untuk Amazon Application Recovery Controller (ARC)

AWS Identity and Access Management (IAM) adalah Layanan AWS yang membantu administrator mengontrol akses ke AWS sumber daya dengan aman. Administrator IAM mengontrol siapa yang dapat diautentikasi (masuk) dan diberi wewenang (memiliki izin) untuk menggunakan sumber daya ARC. IAM adalah Layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

Audiens

Cara Anda menggunakan AWS Identity and Access Management (IAM) berbeda, tergantung pada pekerjaan yang Anda lakukan di ARC.

Pengguna layanan — Jika Anda menggunakan layanan ARC untuk melakukan pekerjaan Anda, administrator Anda memberi Anda kredensi dan izin yang Anda butuhkan. Saat Anda menggunakan lebih banyak fitur ARC untuk melakukan pekerjaan Anda, Anda mungkin memerlukan izin tambahan. Memahami cara akses dikelola dapat membantu Anda meminta izin yang tepat dari administrator Anda. Jika Anda tidak dapat mengakses fitur di ARC, lihatMemecahkan masalah identitas dan akses Amazon Application Recovery Controller (ARC).

Administrator layanan — Jika Anda bertanggung jawab atas sumber daya ARC di perusahaan Anda, Anda mungkin memiliki akses penuh ke ARC. Tugas Anda adalah menentukan fitur dan sumber daya ARC mana yang harus diakses pengguna layanan Anda. Kemudian, Anda harus mengirimkan permintaan kepada administrator IAM untuk mengubah izin pengguna layanan Anda. Tinjau informasi di halaman ini untuk memahami konsep dasar IAM. Untuk mempelajari lebih lanjut tentang bagaimana perusahaan Anda dapat menggunakan IAM dengan ARC, lihatBagaimana kemampuan Amazon Application Recovery Controller (ARC) bekerja dengan IAM.

Administrator IAM - Jika Anda seorang administrator IAM, Anda mungkin ingin mempelajari detail tentang cara menulis kebijakan untuk mengelola akses ke ARC. Untuk melihat contoh kebijakan berbasis identitas ARC yang dapat Anda gunakan di IAM, lihat. Contoh kebijakan berbasis identitas di Amazon Application Recovery Controller (ARC)

Mengautentikasi dengan identitas

Otentikasi adalah cara Anda masuk AWS menggunakan kredensi identitas Anda. Anda harus diautentikasi (masuk ke AWS) sebagai Pengguna root akun AWS, sebagai pengguna IAM, atau dengan mengasumsikan peran IAM.

Anda dapat masuk AWS sebagai identitas federasi dengan menggunakan kredensil yang disediakan melalui sumber identitas. AWS IAM Identity Center Pengguna (IAM Identity Center), autentikasi masuk tunggal perusahaan Anda, dan kredensi Google atau Facebook Anda adalah contoh identitas federasi. Saat Anda masuk sebagai identitas terfederasi, administrator Anda sebelumnya menyiapkan federasi identitas menggunakan peran IAM. Ketika Anda mengakses AWS dengan menggunakan federasi, Anda secara tidak langsung mengambil peran.

Bergantung pada jenis pengguna Anda, Anda dapat masuk ke AWS Management Console atau portal AWS akses. Untuk informasi selengkapnya tentang masuk AWS, lihat Cara masuk ke Panduan AWS Sign-In Pengguna Anda Akun AWS.

Jika Anda mengakses AWS secara terprogram, AWS sediakan kit pengembangan perangkat lunak (SDK) dan antarmuka baris perintah (CLI) untuk menandatangani permintaan Anda secara kriptografis dengan menggunakan kredensyal Anda. Jika Anda tidak menggunakan AWS alat, Anda harus menandatangani permintaan sendiri. Guna mengetahui informasi selengkapnya tentang penggunaan metode yang disarankan untuk menandatangani permintaan sendiri, lihat AWS Signature Version 4 untuk permintaan API dalam Panduan Pengguna IAM.

Apa pun metode autentikasi yang digunakan, Anda mungkin diminta untuk menyediakan informasi keamanan tambahan. Misalnya, AWS merekomendasikan agar Anda menggunakan otentikasi multi-

faktor (MFA) untuk meningkatkan keamanan akun Anda. Untuk mempelajari selengkapnya, lihat Autentikasi multi-faktor dalam Panduan Pengguna AWS IAM Identity Center dan Autentikasi multi-faktor AWS di IAM dalam Panduan Pengguna IAM.

Akun AWS pengguna root

Saat Anda membuat Akun AWS, Anda mulai dengan satu identitas masuk yang memiliki akses lengkap ke semua Layanan AWS dan sumber daya di akun. Identitas ini disebut pengguna Akun AWS root dan diakses dengan masuk dengan alamat email dan kata sandi yang Anda gunakan untuk membuat akun. Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari. Lindungi kredensial pengguna root Anda dan gunakan kredensial tersebut untuk melakukan tugas yang hanya dapat dilakukan pengguna root. Untuk daftar lengkap tugas yang mengharuskan Anda masuk sebagai pengguna root, lihat Tugas yang memerlukan kredensial pengguna root dalam Panduan Pengguna IAM.

Identitas gabungan

Sebagai praktik terbaik, mewajibkan pengguna manusia, termasuk pengguna yang memerlukan akses administrator, untuk menggunakan federasi dengan penyedia identitas untuk mengakses Layanan AWS dengan menggunakan kredensyal sementara.

Identitas federasi adalah pengguna dari direktori pengguna perusahaan Anda, penyedia identitas web, direktori Pusat Identitas AWS Directory Service, atau pengguna mana pun yang mengakses Layanan AWS dengan menggunakan kredensyal yang disediakan melalui sumber identitas. Ketika identitas federasi mengakses Akun AWS, mereka mengambil peran, dan peran memberikan kredensi sementara.

Untuk manajemen akses terpusat, kami sarankan Anda menggunakan AWS IAM Identity Center. Anda dapat membuat pengguna dan grup di Pusat Identitas IAM, atau Anda dapat menghubungkan dan menyinkronkan ke sekumpulan pengguna dan grup di sumber identitas Anda sendiri untuk digunakan di semua aplikasi Akun AWS dan aplikasi Anda. Untuk informasi tentang Pusat Identitas IAM, lihat Apakah itu Pusat Identitas IAM? dalam Panduan Pengguna AWS IAM Identity Center .

Pengguna dan grup IAM

Pengguna IAM adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus untuk satu orang atau aplikasi. Jika memungkinkan, kami merekomendasikan untuk mengandalkan kredensial sementara, bukan membuat pengguna IAM yang memiliki kredensial jangka panjang seperti kata sandi dan kunci akses. Namun, jika Anda memiliki kasus penggunaan tertentu yang memerlukan kredensial jangka panjang dengan pengguna IAM, kami merekomendasikan Anda merotasi kunci

akses. Untuk informasi selengkapnya, lihat Merotasi kunci akses secara teratur untuk kasus penggunaan yang memerlukan kredensial jangka panjang dalam Panduan Pengguna IAM.

Grup IAM adalah identitas yang menentukan sekumpulan pengguna IAM. Anda tidak dapat masuk sebagai grup. Anda dapat menggunakan grup untuk menentukan izin bagi beberapa pengguna sekaligus. Grup mempermudah manajemen izin untuk sejumlah besar pengguna sekaligus. Misalnya, Anda dapat meminta kelompok untuk menyebutkan IAMAdmins dan memberikan izin kepada grup tersebut untuk mengelola sumber daya IAM.

Pengguna berbeda dari peran. Pengguna secara unik terkait dengan satu orang atau aplikasi, tetapi peran dimaksudkan untuk dapat digunakan oleh siapa pun yang membutuhkannya. Pengguna memiliki kredensial jangka panjang permanen, tetapi peran memberikan kredensial sementara. Untuk mempelajari selengkapnya, lihat Kasus penggunaan untuk pengguna IAM dalam Panduan Pengguna IAM.

Peran IAM

Peran IAM adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus. Peran ini mirip dengan pengguna IAM, tetapi tidak terkait dengan orang tertentu. Untuk mengambil peran IAM sementara AWS Management Console, Anda dapat beralih dari pengguna ke peran IAM (konsol). Anda dapat mengambil peran dengan memanggil operasi AWS CLI atau AWS API atau dengan menggunakan URL kustom. Untuk informasi selengkapnya tentang cara menggunakan peran, lihat Metode untuk mengambil peran dalam Panduan Pengguna IAM.

Peran IAM dengan kredensial sementara berguna dalam situasi berikut:

- Akses pengguna terfederasi – Untuk menetapkan izin ke identitas terfederasi, Anda membuat peran dan menentukan izin untuk peran tersebut. Ketika identitas terfederasi mengautentikasi, identitas tersebut terhubung dengan peran dan diberi izin yang ditentukan oleh peran. Untuk informasi tentang peran untuk federasi, lihat Buat peran untuk penyedia identitas pihak ketiga dalam Panduan Pengguna IAM. Jika menggunakan Pusat Identitas IAM, Anda harus mengonfigurasi set izin. Untuk mengontrol apa yang dapat diakses identitas Anda setelah identitas tersebut diautentikasi, Pusat Identitas IAM akan mengorelasikan set izin ke peran dalam IAM. Untuk informasi tentang set izin, lihat Set izin dalam Panduan Pengguna AWS IAM Identity Center .
- Izin pengguna IAM sementara – Pengguna atau peran IAM dapat mengambil peran IAM guna mendapatkan berbagai izin secara sementara untuk tugas tertentu.
- Akses lintas akun – Anda dapat menggunakan peran IAM untuk mengizinkan seseorang (prinsipal tepercaya) di akun lain untuk mengakses sumber daya di akun Anda. Peran adalah cara utama

untuk memberikan akses lintas akun. Namun, dengan beberapa Layanan AWS, Anda dapat melampirkan kebijakan secara langsung ke sumber daya (alih-alih menggunakan peran sebagai proxy). Untuk mempelajari perbedaan antara peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat Akses sumber daya lintas akun di IAM dalam Panduan Pengguna IAM.

- Akses lintas layanan — Beberapa Layanan AWS menggunakan fitur lain Layanan AWS. Misalnya, saat Anda melakukan panggilan dalam suatu layanan, biasanya layanan tersebut menjalankan aplikasi di Amazon EC2 atau menyimpan objek di Amazon S3. Sebuah layanan mungkin melakukannya menggunakan izin prinsipal yang memanggil, menggunakan peran layanan, atau peran terkait layanan.
  - Sesi akses teruskan (FAS) — Saat Anda menggunakan pengguna atau peran IAM untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan sebuah tindakan yang kemudian menginisiasi tindakan lain di layanan yang berbeda. FAS menggunakan izin dari pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. Permintaan FAS hanya dibuat ketika layanan menerima permintaan yang memerlukan interaksi dengan orang lain Layanan AWS atau sumber daya untuk menyelesaikannya. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan ketika mengajukan permintaan FAS, lihat Sesi akses maju.
  - Peran layanan – Peran layanan adalah peran IAM yang dijalankan oleh layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, mengubah, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat Buat sebuah peran untuk mendelegasikan izin ke Layanan AWS dalam Panduan pengguna IAM.
  - Peran terkait layanan — Peran terkait layanan adalah jenis peran layanan yang ditautkan ke peran layanan. Layanan AWS Layanan tersebut dapat menjalankan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Anda Akun AWS dan dimiliki oleh layanan. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.
- Aplikasi yang berjalan di Amazon EC2 — Anda dapat menggunakan peran IAM untuk mengelola kredensyal sementara untuk aplikasi yang berjalan pada EC2 instance dan membuat AWS CLI atau AWS permintaan API. Ini lebih baik untuk menyimpan kunci akses dalam EC2 instance. Untuk menetapkan AWS peran ke EC2 instance dan membuatnya tersedia untuk semua aplikasinya, Anda membuat profil instance yang dilampirkan ke instance. Profil instance berisi peran dan memungkinkan program yang berjalan pada EC2 instance untuk mendapatkan kredensi sementara. Untuk informasi selengkapnya, lihat Menggunakan peran IAM untuk memberikan izin ke aplikasi yang berjalan di EC2 instans Amazon di Panduan Pengguna IAM.

Mengelola akses menggunakan kebijakan

Anda mengontrol akses AWS dengan membuat kebijakan dan melampirkannya ke AWS identitas atau sumber daya. Kebijakan adalah objek AWS yang, ketika dikaitkan dengan identitas atau sumber daya, menentukan izinnya. AWS mengevaluasi kebijakan ini ketika prinsipal (pengguna, pengguna root, atau sesi peran) membuat permintaan. Izin dalam kebijakan menentukan apakah permintaan diizinkan atau ditolak. Sebagian besar kebijakan disimpan AWS sebagai dokumen JSON. Untuk informasi selengkapnya tentang struktur dan isi dokumen kebijakan JSON, lihat Gambaran umum kebijakan JSON dalam Panduan Pengguna IAM.

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Secara default, pengguna dan peran tidak memiliki izin. Untuk memberikan izin kepada pengguna untuk melakukan tindakan di sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM. Administrator kemudian dapat menambahkan kebijakan IAM ke peran, dan pengguna dapat mengambil peran.

Kebijakan IAM mendefinisikan izin untuk suatu tindakan terlepas dari metode yang Anda gunakan untuk melakukan operasinya. Misalnya, anggaplah Anda memiliki kebijakan yang mengizinkan tindakan `iam:GetRole`. Pengguna dengan kebijakan tersebut bisa mendapatkan informasi peran dari AWS Management Console, API AWS CLI, atau AWS API.

Kebijakan berbasis identitas

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, seperti pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana, dan berdasarkan kondisi seperti apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Tentukan izin IAM kustom dengan kebijakan terkelola pelanggan dalam Panduan Pengguna IAM.

Kebijakan berbasis identitas dapat dikategorikan lebih lanjut sebagai kebijakan inline atau kebijakan yang dikelola. Kebijakan inline disematkan langsung ke satu pengguna, grup, atau peran. Kebijakan terkelola adalah kebijakan mandiri yang dapat Anda lampirkan ke beberapa pengguna, grup, dan peran dalam. Akun AWS Kebijakan AWS terkelola mencakup kebijakan terkelola dan kebijakan yang dikelola pelanggan. Untuk mempelajari cara memilih antara kebijakan yang dikelola atau kebijakan inline, lihat Pilih antara kebijakan yang dikelola dan kebijakan inline dalam Panduan Pengguna IAM.

Kebijakan berbasis sumber daya

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya tempat kebijakan dilampirkan, kebijakan menentukan tindakan apa yang dapat dilakukan oleh prinsipal tertentu pada sumber daya tersebut dan dalam kondisi apa. Anda harus menentukan prinsipal dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau. Layanan AWS

Kebijakan berbasis sumber daya merupakan kebijakan inline yang terletak di layanan tersebut. Anda tidak dapat menggunakan kebijakan AWS terkelola dari IAM dalam kebijakan berbasis sumber daya.

Daftar kontrol akses (ACLs)

Access control lists (ACLs) mengontrol prinsipal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACLs mirip dengan kebijakan berbasis sumber daya, meskipun mereka tidak menggunakan format dokumen kebijakan JSON.

Amazon S3, AWS WAF, dan Amazon VPC adalah contoh layanan yang mendukung. ACLs Untuk mempelajari selengkapnya ACLs, lihat Ringkasan daftar kontrol akses (ACL) di Panduan Pengembang Layanan Penyimpanan Sederhana Amazon.

Jenis-jenis kebijakan lain

AWS mendukung jenis kebijakan tambahan yang kurang umum. Jenis-jenis kebijakan ini dapat mengatur izin maksimum yang diberikan kepada Anda oleh jenis kebijakan yang lebih umum.

- Batasan izin – Batasan izin adalah fitur lanjutan tempat Anda mengatur izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas ke entitas IAM (pengguna IAM atau peran IAM). Anda dapat menetapkan batasan izin untuk suatu entitas. Izin yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas milik entitas dan batasan izinnya. Kebijakan berbasis sumber daya yang menentukan pengguna atau peran dalam bidang `Principal` tidak dibatasi oleh batasan izin. Penolakan eksplisit dalam salah satu kebijakan ini akan menggantikan pemberian izin. Untuk informasi selengkapnya tentang batasan izin, lihat Batasan izin untuk entitas IAM dalam Panduan Pengguna IAM.
- Kebijakan kontrol layanan (SCPs) — SCPs adalah kebijakan JSON yang menentukan izin maksimum untuk organisasi atau unit organisasi (OU) di. AWS Organizations AWS Organizations

adalah layanan untuk mengelompokkan dan mengelola secara terpusat beberapa Akun AWS yang dimiliki bisnis Anda. Jika Anda mengaktifkan semua fitur dalam suatu organisasi, maka Anda dapat menerapkan kebijakan kontrol layanan (SCPs) ke salah satu atau semua akun Anda. SCP membatasi izin untuk entitas di akun anggota, termasuk masing-masing. Pengguna root akun AWS Untuk informasi selengkapnya tentang Organizations dan SCPs, lihat Kebijakan kontrol layanan di Panduan AWS Organizations Pengguna.

- Kebijakan kontrol sumber daya (RCPs) — RCPs adalah kebijakan JSON yang dapat Anda gunakan untuk menetapkan izin maksimum yang tersedia untuk sumber daya di akun Anda tanpa memperbarui kebijakan IAM yang dilampirkan ke setiap sumber daya yang Anda miliki. RCP membatasi izin untuk sumber daya di akun anggota dan dapat memengaruhi izin efektif untuk identitas, termasuk Pengguna root akun AWS, terlepas dari apakah itu milik organisasi Anda. Untuk informasi selengkapnya tentang Organizations dan RCPs, termasuk daftar dukungan Layanan AWS tersebut RCPs, lihat Kebijakan kontrol sumber daya (RCPs) di Panduan AWS Organizations Pengguna.
- Kebijakan sesi – Kebijakan sesi adalah kebijakan lanjutan yang Anda berikan sebagai parameter ketika Anda membuat sesi sementara secara programatis untuk peran atau pengguna terfederasi. Izin sesi yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas pengguna atau peran dan kebijakan sesi. Izin juga bisa datang dari kebijakan berbasis sumber daya. Penolakan eksplisit dalam salah satu kebijakan ini akan menggantikan pemberian izin. Untuk informasi selengkapnya, lihat Kebijakan sesi dalam Panduan Pengguna IAM.

Berbagai jenis kebijakan

Ketika beberapa jenis kebijakan berlaku pada suatu permintaan, izin yang dihasilkan lebih rumit untuk dipahami. Untuk mempelajari cara AWS menentukan apakah akan mengizinkan permintaan saat beberapa jenis kebijakan terlibat, lihat Logika evaluasi kebijakan di Panduan Pengguna IAM.

Bagaimana kemampuan Amazon Application Recovery Controller (ARC) bekerja dengan IAM

Untuk informasi tentang cara kerja masing-masing kemampuan Amazon Application Recovery Controller (ARC) dengan IAM, lihat topik berikut:

- IAM untuk pergeseran zona
- IAM untuk pergeseran otomatis zona
- IAM untuk kontrol perutean

- IAM untuk pemeriksaan kesiapan
- IAM untuk sakelar Wilayah

Contoh kebijakan berbasis identitas di Amazon Application Recovery Controller (ARC)

Untuk melihat contoh kebijakan berbasis identitas untuk setiap kemampuan di Amazon Application Recovery Controller (ARC), lihat topik berikut di AWS Identity and Access Management bab-babnya untuk setiap kemampuan:

- Contoh kebijakan berbasis identitas untuk pergeseran otomatis zona di ARC
- Contoh kebijakan berbasis identitas untuk pergeseran zona di ARC
- Contoh kebijakan berbasis identitas untuk kontrol perutean di ARC
- Contoh kebijakan berbasis identitas untuk pemeriksaan kesiapan di ARC

AWS kebijakan terkelola untuk Amazon Application Recovery Controller (ARC)

Untuk informasi tentang kebijakan AWS terkelola untuk kemampuan ARC dengan kebijakan terkelola, termasuk kebijakan terkelola untuk peran terkait layanan, lihat topik berikut:

- Kebijakan terkelola untuk pergeseran otomatis zona
- Kebijakan terkelola untuk kontrol perutean
- Kebijakan terkelola untuk pemeriksaan kesiapan

Pembaruan kebijakan AWS terkelola untuk Amazon Application Recovery Controller (ARC)

Lihat detail tentang pembaruan kebijakan AWS terkelola untuk kapabilitas di ARC sejak layanan ini mulai melacak perubahan ini. Untuk peringatan otomatis tentang perubahan pada halaman ini, berlangganan umpan RSS di halaman riwayat Dokumen ARC.

Perubahan	Deskripsi	Tanggal
AWSZonalAutoshiftPracticeRunSLRPolicy kebijakan terkelola - Kebijakan yang diperbarui	Menambahkan pernyataan kebijakan `Autoshift PracticeCheckPermissions` dengan izin`autoscaling:DescribeAutoScalingGroups` ,, `ec2:DescribeInstances elasticloadbalancing:DescribeTargetHealth` , dan `elasticloadbalancing:DescribeTargetHealth` untuk mendukung pemeriksaan kapasitas seimbang. Untuk mempelajari selengkapnya, lihat Bagaimana zonal autoshift dan praktek berjalan bekerja.	Juni 30, 2025
AWSServiceRoleForPercPracticePolicy— Kebijakan baru	ARC menambahkan peran terkait layanan baru untuk autoshift dan praktik berjalan. ARC menggunakan izin yang diaktifkan oleh peran terkait layanan untuk memantau CloudWatch alarm Amazon yang disediakan pelanggan dan AWS Health Dashboard peristiwa pelanggan untuk menjalankan praktik, dan untuk memulai praktik berjalan.	30 November 2023

Perubahan	Deskripsi	Tanggal
	Untuk mempelajari lebih lanjut tentang peran baru terkait layanan, lihat. Izin peran terkait layanan untuk AWSService RoleForZonalAutoshiftPracticeRun	
AmazonRoute53 RecoveryControlConfigReadOnlyAccess - Kebijakan yang diperbarui	Menambahkan izin untuk`GetResourcePolicy` , untuk mendukung pengembalian detail tentang kebijakan AWS Resource Access Manager sumber daya untuk sumber daya bersama.	18 Oktober 2023

Route53 RecoveryReadinessServiceRolePolicy - Kebijakan yang diperbarui	ARC menambahkan izin baru untuk menanyakan informasi tentang EC2 instans Amazon.
	
	ARC menggunakan izin berikut untuk mendukung polling EC2 instans Amazon, untuk menjalankan pemeriksaan kesiapan dan menentukan status kesiapan untuk instans.
	
	`ec2:DescribeVpnGateways`
	
	`ec2:DescribeCustomerGateways`
		17 Februari 2023

Perubahan	Deskripsi	Tanggal
Route53 RecoveryReadinessServiceRolePolicy - Kebijakan yang diperbarui	ARC menambahkan izin baru untuk menanyakan informasi tentang fungsi Lambda. ARC menggunakan izin berikut untuk menanyakan informasi tentang fungsi Lambda untuk menjalankan pemeriksaan kesiapan dan menentukan status kesiapan untuk fungsi tersebut. `lambda:ListProvisionedConcurrencyConfigs`	31 Agustus 2022
AmazonRoute53 RecoveryControlConfigFullAccess - Kebijakan yang diperbarui	Menghapus izin Amazon Route 53 dari kebijakan dan menambahkan catatan yang mencantumkan izin opsional.	26 Mei 2022
AmazonRoute53 RecoveryControlConfigFullAccess - Kebijakan yang diperbarui	Menambahkan izin Amazon Route 53 yang diperlukan yang tidak diperlukan ke kebijakan.	15 April 2022
AmazonRoute53 RecoveryClusterReadOnlyAccess - Kebijakan yang diperbarui	ARC menambahkan izin baru,`route53-recovery-cluster:ListRoutingControls` , untuk memungkinkan daftar kontrol routing ARNs dengan ketersediaan tinggi.	15 Maret 2022

Perubahan	Deskripsi	Tanggal
AmazonRoute53 RecoveryControlConfigReadOnlyAccess - Kebijakan yang diperbarui	ARC menambahkan izin baru,`route53-recovery-control-config:ListTagsForResources` , untuk mengizinkan tag daftar untuk sumber daya.	Desember 20, 2021
Route53 RecoveryReadinessServiceRolePolicy - Kebijakan yang diperbarui	ARC menambahkan izin baru untuk menanyakan informasi tentang Amazon API Gateway. ARC menggunakan izin,`apigateway:GET` , untuk menanyakan informasi tentang API Gateway untuk menjalankan pemeriksaan kesiapan dan menentukan status kesiapan.	28 Oktober 2021
AmazonRoute53 RecoveryReadinessReadOnlyAccess - Ditambahkan izin baru	ARC menambahkan dua izin baru ke AmazonRoute53 RecoveryReadinessReadOnlyAccess: ARC menggunakan `route53-recovery-readiness:GetArchitectureRecommendations` dan `route53-recovery-readiness:GetCellReadinessSummary` mengizinkan akses hanya-baca ke tindakan ini untuk bekerja dengan kesiapan pemulihan.	Oktober 15, 2021

Perubahan	Deskripsi	Tanggal

Route53 RecoveryReadinessServiceRolePolicy - Kebijakan yang diperbarui	ARC menambahkan izin baru untuk menanyakan informasi tentang fungsi Lambda.
	
	ARC menggunakan izin berikut untuk menanyakan informasi tentang fungsi Lambda untuk menjalankan pemeriksaan kesiapan dan menentukan status kesiapan untuk fungsi tersebut.
	
	`lambda:GetFunctionConcurrency`
	
	`lambda:GetFunctionConfiguration`
	
	`lambda:GetProvisionedConcurrencyConfig`
	
	`lambda:ListAliases`
	
	`lambda:ListVersionsByFunction`
	
	`lambda:ListEventSourceMappings`
	
	`lambda:ListFunctions`
		Oktober 8, 2021

Perubahan	Deskripsi	Tanggal
Route53 RecoveryReadinessServiceRolePolicy - Ditambahkan kebijakan terkelola baru	ARC menambahkan kebijakan terkelola baru berikut: AmazonRoute53 RecoveryReadinessFullAccess AmazonRoute53 RecoveryReadinessReadOnlyAccess AmazonRoute53 RecoveryClusterFullAccess AmazonRoute53 RecoveryClusterReadOnlyAccess AmazonRoute53 RecoveryControlConfigFullAccess AmazonRoute53 RecoveryControlConfigReadOnlyAccess	18 Agustus 2021
ARC mulai melacak perubahan	ARC mulai melacak perubahan untuk kebijakan yang AWS dikelola.	27 Juli 2021

Memecahkan masalah identitas dan akses Amazon Application Recovery Controller (ARC)

Gunakan informasi berikut untuk membantu Anda mendiagnosis dan memperbaiki masalah umum yang mungkin Anda temui saat bekerja dengan Amazon Application Recovery Controller (ARC) dan IAM.

Topik

- Saya tidak berwenang untuk melakukan tindakan di ARC
- Saya tidak berwenang untuk melakukan iam: PassRole
- Saya ingin mengizinkan orang di luar saya Akun AWS untuk mengakses sumber daya ARC saya

Saya tidak berwenang untuk melakukan tindakan di ARC

Jika AWS Management Console memberitahu Anda bahwa Anda tidak berwenang untuk melakukan tindakan, maka Anda harus menghubungi administrator Anda untuk bantuan. Administrator Anda adalah orang yang memberi Anda kredensi Anda.

Contoh kesalahan berikut terjadi ketika pengguna IAM `mateojackson` mencoba menggunakan konsol untuk melihat detail tentang suatu sumber daya fiktif *`my-example-widget`*, tetapi tidak memiliki izin fiktif `route53-recovery-readiness:`*`GetWidget`*.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 route53-recovery-readiness:GetWidget on resource: my-example-widget
```

Dalam hal ini, Mateo meminta administratornya untuk memperbarui kebijakannya agar dia dapat mengakses *`my-example-widget`* menggunakan `route53-recovery-readiness:`*`GetWidget`* tindakan.

Saya tidak berwenang untuk melakukan iam: PassRole

Jika Anda menerima kesalahan bahwa Anda tidak berwenang untuk melakukan `iam:PassRole` tindakan, kebijakan Anda harus diperbarui agar Anda dapat meneruskan peran ke ARC.

Beberapa Layanan AWS memungkinkan Anda untuk meneruskan peran yang ada ke layanan tersebut alih-alih membuat peran layanan baru atau peran terkait layanan. Untuk melakukannya, Anda harus memiliki izin untuk meneruskan peran ke layanan.

Contoh kesalahan berikut terjadi ketika pengguna IAM bernama `marymajor` mencoba menggunakan konsol untuk melakukan tindakan di ARC. Namun, tindakan tersebut memerlukan layanan untuk mendapatkan izin yang diberikan oleh peran layanan. Mary tidak memiliki izin untuk meneruskan peran tersebut pada layanan.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Dalam kasus ini, kebijakan Mary harus diperbarui agar dia mendapatkan izin untuk melakukan tindakan `iam:PassRole` tersebut.

Jika Anda memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Saya ingin mengizinkan orang di luar saya Akun AWS untuk mengakses sumber daya ARC saya

Anda dapat membuat peran yang dapat digunakan pengguna di akun lain atau orang-orang di luar organisasi Anda untuk mengakses sumber daya Anda. Anda dapat menentukan siapa saja yang dipercaya untuk mengambil peran tersebut. Untuk layanan yang mendukung kebijakan berbasis sumber daya atau daftar kontrol akses (ACLs), Anda dapat menggunakan kebijakan tersebut untuk memberi orang akses ke sumber daya Anda.

Untuk mempelajari selengkapnya, periksa referensi berikut:

- Untuk mengetahui apakah ARC mendukung fitur-fitur ini, lihatBagaimana kemampuan Amazon Application Recovery Controller (ARC) bekerja dengan IAM.
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda di seluruh sumber daya Akun AWS yang Anda miliki, lihat Menyediakan akses ke pengguna IAM di pengguna lain Akun AWS yang Anda miliki di Panduan Pengguna IAM.
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda kepada pihak ketiga Akun AWS, lihat Menyediakan akses yang Akun AWS dimiliki oleh pihak ketiga dalam Panduan Pengguna IAM.
- Untuk mempelajari cara memberikan akses melalui federasi identitas, lihat Menyediakan akses ke pengguna terautentikasi eksternal (federasi identitas) dalam Panduan Pengguna IAM.
- Untuk mempelajari perbedaan antara menggunakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat Akses sumber daya lintas akun di IAM di Panduan Pengguna IAM.

Pencatatan dan pemantauan di ARC

Pemantauan adalah bagian penting dari menjaga ketersediaan dan kinerja ARC dan AWS solusi Anda. Anda harus mengumpulkan data pemantauan dari semua bagian AWS solusi Anda sehingga Anda dapat lebih mudah men-debug kegagalan multi-titik jika terjadi. AWS menyediakan beberapa alat untuk memantau sumber daya dan aktivitas ARC Anda, dan menanggapi potensi insiden, misalnya, dan AWS CloudTrail Amazon. CloudWatch

Untuk informasi tentang pemantauan untuk setiap kemampuan di ARC, lihat topik berikut:

- Pencatatan dan pemantauan untuk pergeseran zona
- Pencatatan dan pemantauan untuk pergeseran otomatis zona

- Pencatatan dan pemantauan untuk kontrol perutean
- Pencatatan dan pemantauan untuk sakelar Wilayah
- Pencatatan dan pemantauan untuk pemeriksaan kesiapan

Validasi kepatuhan untuk Amazon Application Recovery Controller

Auditor pihak ketiga menilai keamanan dan kepatuhan Amazon Application Recovery Controller sebagai bagian dari beberapa program AWS kepatuhan. Hal ini mencakup SOC, PCI, HIPAA, dan lainnya.

Untuk mempelajari apakah an Layanan AWS berada dalam lingkup program kepatuhan tertentu, lihat Layanan AWS di Lingkup oleh Program Kepatuhan Layanan AWS dan pilih program kepatuhan yang Anda minati. Untuk informasi umum, lihat Program AWS Kepatuhan Program AWS .

Anda dapat mengunduh laporan audit pihak ketiga menggunakan AWS Artifact. Untuk informasi selengkapnya, lihat Mengunduh Laporan di AWS Artifact .

Tanggung jawab kepatuhan Anda saat menggunakan Layanan AWS ditentukan oleh sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, dan hukum dan peraturan yang berlaku. AWS menyediakan sumber daya berikut untuk membantu kepatuhan:

- Kepatuhan dan Tata Kelola Keamanan – Panduan implementasi solusi ini membahas pertimbangan arsitektur serta memberikan langkah-langkah untuk menerapkan fitur keamanan dan kepatuhan.
- Referensi Layanan yang Memenuhi Syarat HIPAA — Daftar layanan yang memenuhi syarat HIPAA. Tidak semua memenuhi Layanan AWS syarat HIPAA.
- AWS Sumber Daya AWS — Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- AWS Panduan Kepatuhan Pelanggan - Memahami model tanggung jawab bersama melalui lensa kepatuhan. Panduan ini merangkum praktik terbaik untuk mengamankan Layanan AWS dan memetakan panduan untuk kontrol keamanan di berbagai kerangka kerja (termasuk Institut Standar dan Teknologi Nasional (NIST), Dewan Standar Keamanan Industri Kartu Pembayaran (PCI), dan Organisasi Internasional untuk Standardisasi (ISO)).
- Mengevaluasi Sumber Daya dengan Aturan dalam Panduan AWS Config Pengembang — AWS Config Layanan menilai seberapa baik konfigurasi sumber daya Anda mematuhi praktik internal, pedoman industri, dan peraturan.

- AWS Security Hub— Ini Layanan AWS memberikan pandangan komprehensif tentang keadaan keamanan Anda di dalamnya AWS. Security Hub menggunakan kontrol keamanan untuk sumber daya AWS Anda serta untuk memeriksa kepatuhan Anda terhadap standar industri keamanan dan praktik terbaik. Untuk daftar layanan dan kontrol yang didukung, lihat Referensi kontrol Security Hub.
- Amazon GuardDuty — Ini Layanan AWS mendeteksi potensi ancaman terhadap beban kerja Akun AWS, kontainer, dan data Anda dengan memantau lingkungan Anda untuk aktivitas mencurigakan dan berbahaya. GuardDuty dapat membantu Anda mengatasi berbagai persyaratan kepatuhan, seperti PCI DSS, dengan memenuhi persyaratan deteksi intrusi yang diamanatkan oleh kerangka kerja kepatuhan tertentu.
- AWS Audit ManagerIni Layanan AWS membantu Anda terus mengaudit AWS penggunaan Anda untuk menyederhanakan cara Anda mengelola risiko dan kepatuhan terhadap peraturan dan standar industri.

Ketahanan dalam Pengontrol Pemulihan Aplikasi Amazon

Infrastruktur AWS global dibangun di sekitar Wilayah AWS dan Availability Zones. Wilayah AWS menyediakan beberapa Availability Zone yang terpisah secara fisik dan terisolasi, yang terhubung dengan latensi rendah, throughput tinggi, dan jaringan yang sangat redundan. Dengan Zona Ketersediaan, Anda dapat merancang serta mengoperasikan aplikasi dan basis data yang secara otomatis melakukan fail over di antara zona tanpa gangguan. Zona Ketersediaan memiliki ketersediaan dan toleransi kesalahan yang lebih baik, dan dapat diskalakan dibandingkan infrastruktur pusat data tunggal atau multi tradisional.

Untuk informasi selengkapnya tentang Wilayah AWS dan Availability Zone, lihat Infrastruktur AWS Global.

Selain infrastruktur AWS global, ARC menawarkan beberapa fitur untuk membantu mendukung ketahanan data dan kebutuhan cadangan Anda.

Keamanan infrastruktur di Amazon Application Recovery Controller

Sebagai layanan terkelola, dilindungi oleh keamanan jaringan AWS global. Untuk informasi tentang layanan AWS keamanan dan cara AWS melindungi infrastruktur, lihat Keamanan AWS Cloud. Untuk mendesain AWS lingkungan Anda menggunakan praktik terbaik untuk keamanan infrastruktur, lihat Perlindungan Infrastruktur dalam Kerangka Kerja yang AWS Diarsiteksikan dengan Baik Pilar Keamanan.

Anda menggunakan panggilan API yang AWS dipublikasikan untuk mengakses ARC melalui jaringan. Klien harus mendukung hal-hal berikut:

- Keamanan Lapisan Pengangkutan (TLS). Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Sandi cocok dengan sistem kerahasiaan maju sempurna (perfect forward secrecy, PFS) seperti DHE (Ephemeral Diffie-Hellman) atau ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Sebagian besar sistem modern seperti Java 7 dan versi lebih baru mendukung mode-mode ini.

Selain itu, permintaan harus ditandatangani menggunakan ID kunci akses dan kunci akses rahasia yang terkait dengan prinsipal IAM. Atau Anda bisa menggunakan AWS Security Token Service (AWS STS) untuk membuat kredensial keamanan sementara guna menandatangani permintaan.

Riwayat dokumen untuk Panduan Pengembang Amazon Application Recovery Controller (ARC)

Entri berikut menjelaskan perubahan penting yang dibuat pada dokumentasi Amazon Application Recovery Controller (ARC).

- Versi: terbaru
- Pembaruan dokumentasi terbaru: Agustus 1, 2025

Perubahan	Deskripsi	Tanggal
Layanan sakelar Wilayah Baru	Peralihan wilayah memungkinkan pelanggan untuk mengatur langkah-langkah spesifik, mendukung lintas akun, yang diperlukan untuk mengoperasikan aplikasi Multi-wilayah mereka dari yang lain. Wilayah AWS Untuk informasi selengkapnya, lihat Sakelar wilayah di ARC.	Agustus 1, 2025
Perangkat tambahan untuk latihan lari	Anda sekarang dapat memulai latihan sesuai permintaan di ARC. Selain itu, praktik berjalan sekarang mencakup pemeriksaan untuk kapasitas yang cukup di wilayah lain AZs di Wilayah. Untuk informasi selengkapnya, lihat Cara kerjanya.	Juni 30, 2025
Memperbarui kebijakan terkelola	Memperbarui kebijakan `AWSZonalAutoshiftP`	Juni 30, 2025

Perubahan	Deskripsi	Tanggal
	`racticeRunSLRPolicy` terkelola dengan menambahkan pernyataan kebijakan `AutoshiftPracticeCheckPermissions` dengan izin`autoscaling:DescribeAutoScalingGroups` ,, `ec2:DescribeInstances elasticloadbalancing:DescribeTargetHealth` , dan `elasticloadbalancing:DescribeTargetHealth` untuk mendukung pemeriksaan kapasitas seimbang. Untuk informasi selengkapnya, lihat kebijakan AWSZonalAutoshiftPracticeRunSLRPolicy terkelola.	
Pembaruan untuk jenis pengecualian untuk pergeseran otomatis zona	Anda sekarang dapat berinteraksi dengan pergeseran otomatis zona berdasarkan per sumber daya. Untuk informasi selengkapnya, lihat Cara kerjanya.	April 21, 2025

Perubahan	Deskripsi	Tanggal
Uji autoshift zona ARC dengan AWS FIS	Anda dapat menggunak an AWS FIS untuk menguji bagaimana ARC zonal autoshift secara otomatis memulihkan aplikasi Anda selama gangguan daya AZ Untuk informasi lebih lanjut, lihat Menguji pergeseran otomatis zona dengan. AWS FIS	Maret 26, 2025
ARC sekarang mendukung IPv6 titik akhir untuk kontrol perutean dan pergeseran zona.	ARC sekarang mendukung IPv6 titik akhir untuk kontrol perutean dan pergeseran zona. Untuk informasi selengkap nya, lihat Mengatur komponen kontrol perutean.	November 21, 2024
Kemampuan pergeseran zona untuk grup Amazon EC2 Auto Scaling	ARC sekarang mendukung pergeseran zona untuk grup Amazon EC2 Auto Scaling. Untuk informasi selengkapnya, lihat Dukungan untuk grup EC2 Auto Scaling Amazon.	November 18, 2024

Perubahan	Deskripsi	Tanggal
Kemampuan pergeseran zona untuk Amazon EKS	Anda dapat memulai pergeseran zona untuk kluster Amazon EKS, atau Anda dapat mengizinkan melakukan nya AWS untuk Anda dengan mengaktifkan pergeseran otomatis zona. Pergeseran ini memperbarui alur lalu lintas east-to-west jaringan di klaster Anda untuk hanya mempertim bangkan titik akhir jaringan untuk Pod yang berjalan di node pekerja dalam keadaan sehat AZs. Untuk informasi selengkap nya, lihat Support for Amazon Elastic Kubernetes Service.	Oktober 22, 2024
Kemampuan pergeseran zona untuk Network Load Balancers	ARC sekarang mendukung pergeseran zona untuk Network Load Balancers dengan konfigurasi cross-zon e enabled atau cross-zone disabled. Untuk informasi selengkapnya, lihat Support for Network Load Balancers.	Oktober 11, 2024

Perubahan	Deskripsi	Tanggal
Pemberitahuan pengamat pergeseran otomatis	Dengan notifikasi pengamat pergeseran otomatis, Anda dapat mengonfigurasi pergeseran otomatis zona untuk memberi tahu Anda, melalui Amazon EventBrid ge, setiap kali AWS memulai perpindahan otomatis untuk mengalihkan lalu lintas dari Zona Ketersediaan yang berpotensi mengalami gangguan. Anda tidak perlu mengonfigurasi sumber daya tertentu dengan pergeseran otomatis zona untuk mengaktif kan pemberitahuan terpisah ini. Untuk informasi selengkapnya, lihat Menggunakan pergesera n otomatis zona dengan Amazon. EventBridge	Juli 12, 2024

Perubahan	Deskripsi	Tanggal
Doc reorganisasi dengan masing-masing kemampuan	Mengatur ulang konten panduan pengembang untuk dibungkam menjadi panduan sub-dev. Artinya, sekarang ada bagian terpisah yang berisi informasi komprehensif untuk setiap kemampuan di ARC: pergeseran zona dan pergeseran otomatis zona untuk pemulihan Multi-AZ, dan kontrol perutean dan pemeriksaan kesiapan untuk pemulihan Multi-wilayah. Untuk informasi selengkapnya, lihat Apa itu Amazon Application Recovery Controller (ARC).	April 30, 2024
Menambahkan kemampuan pergeseran otomatis zona	Menambahkan kemampuan baru di ARC di mana Anda mengizinkan AWS untuk mengalihkan lalu lintas sumber daya untuk aplikasi dari Availability Zone, atas nama Anda, untuk membantu mengurangi waktu pemulihan selama acara. Untuk informasi selengkapnya, lihat Zonal autoshift di Amazon Application Recovery Controller (ARC).	30 November 2023

Perubahan	Deskripsi	Tanggal
Menambahkan peran terkait layanan baru	Menambahkan peran terkait layanan baru, AWSServic eRoleForZonalAutoshiftPract iceRun, untuk menjalankan praktik pergeseran otomatis zona. Untuk informasi selengkapnya, lihat Izin peran terkait layanan untuk. AWSService RoleForZo nalAutoshiftPracticeRun	30 November 2023
Menambahkan dukungan lintas akun untuk cluster	Menambahkan dukungan lintas akun untuk cluster di ARC dengan AWS Resource Access Manager, sehingga Anda dapat dengan mudah dan aman menggunakan satu cluster untuk meng-host panel kontrol dan kontrol routing yang dimiliki oleh beberapa akun berbeda. AWS Untuk informasi selengkapnya, lihat Support cross-account untuk cluster di ARC.	18 Oktober 2023

Perubahan	Deskripsi	Tanggal
Memperbarui kebijakan terkelola	Memperbarui kebijakan `AmazonRoute53RecoveryControlConfigReadOnly` terkelola untuk menambahkan izin`GetResourcePolicy`, guna mendukung pengembalian detail tentang kebijakan AWS Resource Access Manager sumber daya untuk sumber daya bersama. Untuk informasi selengkapnya, lihat Kebijakan terkelola AWS.	September 19, 2023
Peran terkait layanan yang diperbarui	Menambahkan izin baru, `ec2:DescribeVpnGateways` dan`ec2:DescribeCustomerGateways`, ke peran terkait layanan untuk ARC, untuk mendukung polling instans Amazon. EC2 Untuk informasi selengkapnya, lihat Menggunakan peran terkait layanan untuk ARC.	17 Februari 2023

Perubahan	Deskripsi	Tanggal
Rilis GA untuk pergeseran zona	Mendukung rilis GA pergeseran zona untuk ARC, yang mencakup kontrol akses berbasis atribut (ABAC) untuk sumber daya terkelola yang terdaftar di ARC untuk pergeseran zona. Untuk informasi selengkapnya, lihat Kontrol akses berbasis atribut (ABAC) dengan ARC.	10 Januari 2023
Menambahkan pergeseran zona Multi-AZ baru	Menambahkan konten yang menjelaskan layanan baru di ARC, pergeseran zona, untuk aplikasi Multi-AZ. Anda dapat memulai pergeseran zona untuk memindahkan lalu lintas sementara untuk sumber daya penyeimbang beban dari Availability Zone. Untuk informasi lebih lanjut, lihat Pergeseran zona di ARC.	28 November 2022
Peran terkait layanan yang diperbarui	Menambahkan izin baru,`lambda:ListProvisionedConcurrencyConfigs` , ke peran terkait layanan untuk ARC untuk menanyakan informasi tentang fungsi Lambda. Untuk informasi selengkapnya, lihat Menggunakan peran terkait layanan untuk ARC.	31 Agustus 2022

Perubahan	Deskripsi	Tanggal
Kebijakan terkelola yang diperbarui	Memperbarui kebijakan `AmazonRoute53RecoveryControlConfigFullAccess` terkelola untuk menghapus izin Amazon Route 53 dan mencantumkannya sebagai opsional. Untuk informasi selengkapnya, lihat kebijakan AWS terkelola untuk Amazon Application Recovery Controller (ARC).	26 Mei 2022
Kebijakan terkelola yang diperbarui	Memperbarui kebijakan `AmazonRoute53RecoveryControlConfigFullAccess` terkelola untuk menyertakan izin Amazon Route 53 yang diperlukan. Untuk informasi selengkapnya, lihat kebijakan AWS terkelola untuk Amazon Application Recovery Controller (ARC).	15 April 2022
Menambahkan contoh CLI untuk API kontrol perutean daftar baru	Menambahkan contoh perintah CLI dan rekomendasi praktik terbaik untuk operasi API kontrol perutean daftar baru yang disertakan dalam API bidang data ARC yang sangat andal. Untuk informasi selengkapnya, lihat Daftar dan perbarui kontrol dan status perutean.	31 Maret 2022

Perubahan	Deskripsi	Tanggal
Menambahkan dukungan untuk mengesampingkan aturan keselamatan	Menambahkan dukungan untuk mengesampingkan aturan keselamatan, yang memungkinkan Anda melewati perlindungan kontrol perutean yang diberlakukan dengan aturan keselamatan yang telah Anda konfigurasi. Pengganti an aturan keselamatan dapat diperlukan, misalnya, dalam skenario “pecah kaca” selama kegagalan untuk pemulihan bencana. Untuk informasi selengkap nya, lihat Mengganti aturan keselamatan untuk mengubah rute lalu lintas.	2 Maret 2022
Menambahkan dukungan penandaan tambahan	Menambahkan dukungan untuk menandai sumber daya tambahan di ARC, termasuk cluster, panel kontrol, kontrol perutean, dan aturan keselamatan. Untuk informasi selengkap nya, lihat Menandai di Amazon Application Recovery Controlle r (ARC).	Desember 20, 2021

Perubahan	Deskripsi	Tanggal
Kebijakan terkelola yang diperbarui	Memperbarui kebijakan `AmazonRoute53RecoveryControlConfigReadOnly` terkelola untuk menambahkan izin untuk mencantumkan tag untuk sumber daya. Untuk informasi selengkapnya, lihat kebijakan AWS terkelola untuk Amazon Application Recovery Controller (ARC)	Desember 20, 2021
Menambahkan dukungan untuk peringatan real-time dengan EventBridge	Menambahkan dukungan untuk EventBridge, yang berarti bahwa sekarang Anda dapat menambahkan aturan untuk mendapatkan peringatan dan menindaklanjuti perubahan status kesiapan ARC, misalnya, ketika status berubah dari SIAP menjadi TIDAK SIAP. Untuk informasi selengkapnya, lihat Menggunakan ARC dengan Amazon EventBridge.	Desember 20, 2021

Perubahan	Deskripsi	Tanggal
Menambahkan contoh kode status kontrol routing	Menambahkan contoh kode untuk mengilustrasikan titik akhir cluster yang mencoba secara berurutan saat Anda menggunakan operasi API untuk mendapatkan atau memperbarui status kontrol perutean. Untuk informasi selengkapnya, lihat contoh API untuk Amazon Application Recovery Controller (ARC).	November 16, 2021
Menambahkan izin baru ke kebijakan hanya-baca	Menambahkan dua izin baru ke kebijakan`AmazonRoute53RecoveryReadinessReadOnlyAccess` : `route53-recovery-readiness:GetArchitectureRecommendations` dan`route53-recovery-readiness:GetCellReadinessSummary` . Untuk informasi selengkapnya, lihat kebijakan AWS terkelola untuk Amazon Application Recovery Controller (ARC).	November 9, 2021

Perubahan	Deskripsi	Tanggal
Menambahkan dukungan untuk jenis sumber daya Amazon API Gateway	Menambahkan jenis sumber daya baru, Amazon API Gateway, dan memperbar ui izin peran terkait layanan ARC sehingga ARC dapat mengaudit API Gateway dengan pemeriksaan kesiapan. Untuk informasi selengkapnya, lihat Aturan kesiapan dan jenis sumber daya yang didukung serta Menggunakan peran terkait layanan untuk ARC.	28 Oktober 2021
Menambahkan dukungan untuk jenis sumber daya fungsi Lambda	Menambahkan jenis sumber daya baru, fungsi Lambda, dan memperbarui izin peran terkait layanan ARC sehingga ARC dapat mengaudit fungsi Lambda dengan pemeriksaan kesiapan. Untuk informasi selengkapnya, lihat Aturan kesiapan dan jenis sumber daya yang didukung serta Menggunakan peran terkait layanan untuk ARC.	Oktober 8, 2021

Perubahan	Deskripsi	Tanggal
Ditambahkan link ke CloudFormation dan Terraform template	Menambahkan tautan ke templat Terraform yang dapat diunduh AWS CloudFormation dan Hashicorp untuk membantu Anda memulai dengan cepat menggunakan ARC. Untuk informasi lebih lanjut, lihat Kesiapan pemulihan dengan aplikasi baru.	13 September 2021
Menambahkan kebijakan terkelola baru	Menambahkan kebijakan AWS terkelola berikut untuk ARC: `AmazonRoute53RecoveryReadinessFullAccess` `AmazonRoute53RecoveryReadinessReadOnlyAccess` ,`AmazonRoute53RecoveryClusterFullAccess` ,`AmazonRoute53RecoveryClusterReadOnlyAccess` ,,`AmazonRoute53RecoveryControlConfigFullAccess` , dan`AmazonRoute53RecoveryControlConfigReadOnlyAccess` . Untuk informasi selengkapnya, lihat kebijakan AWS terkelola untuk Amazon Application Recovery Controller (ARC).	18 Agustus 2021

Perubahan	Deskripsi	Tanggal
Mulai melacak kebijakan AWS terkelola untuk Amazon Application Recovery Controller (ARC)	Pembaruan untuk kebijakan terkelola akan dilacak dari tanggal rilis awal ke depan. Untuk informasi selengkapnya, lihat kebijakan AWS terkelola untuk Amazon Application Recovery Controller (ARC).	27 Juli 2021
Rilis awal Amazon Application Recovery Controller (ARC)	ARC meningkatkan ketersediaan aplikasi dengan mengoordinasikan failover secara terpusat di dalam AWS Wilayah atau di beberapa Wilayah. ARC menyediakan pemeriksaan kesiapan untuk memastikan bahwa aplikasi Anda diskalakan untuk menangani lalu lintas failover dan dikonfigurasi untuk merutekan kegagalan. Ini juga menyediakan kontrol perutean yang sangat andal sehingga Anda dapat memulihkan aplikasi dengan mengalihkan lalu lintas, misalnya, di seluruh Availability Zone atau Regions. Untuk informasi lebih lanjut, lihat Apa itu ARC? .	27 Juli 2021

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