aws

Panduan Pengguna

Amazon EBS

Amazon EBS: Panduan Pengguna

Table of Contents

Apa itu Amazon Elastic Block Store?

Amazon Elastic Block Store (Amazon EBS) menyediakan sumber daya penyimpanan blok berkinerja tinggi yang dapat diskalakan yang dapat digunakan dengan instans Amazon Elastic Compute Cloud (Amazon EC2). Dengan Amazon Elastic Block Store, Anda dapat membuat dan mengelola sumber daya penyimpanan blok berikut:

- Volume Amazon EBS — Ini adalah volume penyimpanan yang Anda lampirkan ke instans Amazon EC2. Setelah Anda melampirkan volume ke sebuah instance, Anda dapat menggunakannya dengan cara yang sama seperti Anda akan menggunakan hard drive lokal yang terpasang ke komputer, misalnya untuk menyimpan file atau menginstal aplikasi.
- Snapshot Amazon EBS — Ini adalah point-in-time cadangan volume Amazon EBS yang bertahan secara independen dari volume itu sendiri. Anda dapat membuat snapshot untuk mencadangkan data pada volume Amazon EBS Anda. Anda kemudian dapat memulihkan volume baru dari snapshot tersebut kapan saja.

Topik

Fitur-fitur Amazon EBS

Amazon EBS menyediakan fitur dan manfaat berikut:

- Beberapa jenis volume - Amazon EBS menyediakan beberapa jenis volume yang memungkinkan Anda mengoptimalkan kinerja penyimpanan dan biaya untuk berbagai aplikasi. Jenis volume dibagi menjadi dua kategori utama: penyimpanan yang didukung SSD untuk beban kerja transaksional, dan penyimpanan yang didukung HDD untuk beban kerja intensif throughput.
- Skalabilitas — Anda dapat membuat volume Amazon EBS dengan spesifikasi kapasitas dan kinerja yang memenuhi kebutuhan Anda. Saat kebutuhan Anda berubah, Anda dapat menggunakan operasi Volume Elastis untuk meningkatkan kapasitas atau menyetel kinerja secara dinamis, tanpa waktu henti.

- Backup dan recovery — Gunakan snapshot Amazon EBS untuk mencadangkan data yang tersimpan di volume Anda. Anda kemudian dapat menggunakan snapshot tersebut untuk memulihkan volume secara instan atau memigrasikan data di seluruh AWS akun, AWS Wilayah, atau Availability Zone.
- Perlindungan data — Gunakan enkripsi Amazon EBS untuk mengenkripsi volume Amazon EBS dan snapshot Amazon EBS Anda. Operasi enkripsi terjadi pada server yang meng-host instans Amazon EC2, memastikan keamanan keduanya data-at-rest dan data-in-transit antara instance dan volume terlampir dan snapshot berikutnya.
- Ketersediaan dan daya tahan data — volume io2 Block Express memberikan daya tahan 99,999% dengan tingkat kegagalan tahunan 0,001%. Jenis volume lainnya memberikan daya tahan 99,8% hingga 99,9% dengan tingkat kegagalan tahunan 0,1% hingga 0,2%. Selain itu, data volume secara otomatis direplikasi di beberapa server di Availability Zone untuk mencegah hilangnya data dari kegagalan komponen tunggal.
- Pengarsipan data — EBS Snapshots Archive menyediakan tingkat penyimpanan berbiaya rendah untuk mengarsipkan point-in-time salinan Snapshot EBS lengkap yang harus Anda simpan selama 90 hari atau lebih untuk alasan peraturan dan kepatuhan, atau untuk rilis proyek masa depan.

Layanan terkait

Amazon EBS bekerja dengan layanan berikut:

- Amazon Elastic Compute Cloud — Layanan yang memungkinkan Anda meluncurkan dan mengelola mesin virtual (instans Amazon EC2) di Cloud. AWS Anda dapat melampirkan volume EBS ke instance tersebut dan menggunakannya dengan cara yang sama seperti Anda menggunakan hard drive lokal, misalnya untuk menyimpan file atau menginstal aplikasi. Untuk informasi selengkapnya, lihat Apa itu Amazon EC2?
- AWS Key Management Service— Layanan terkelola yang memungkinkan Anda membuat dan mengelola kunci kriptografi. Anda dapat menggunakan kunci AWS KMS kriptografi untuk mengenkripsi data yang disimpan di volume Amazon EBS Anda dan di snapshot Amazon EBS Anda. Untuk informasi selengkapnya, lihat Cara Amazon EBS menggunakan AWS KMS.
- Amazon Data Lifecycle Manager — Layanan terkelola yang mengotomatiskan pembuatan, penyimpanan, dan penghapusan snapshot EBS dan AMI yang didukung EBS. Anda dapat menggunakan Amazon Data Lifecycle Manager untuk mengotomatiskan pencadangan volume Amazon EBS dan instans Amazon EC2. Untuk informasi selengkapnya, lihat Amazon Data Lifecycle Manager.

- EBS direct API — Layanan yang memungkinkan Anda membuat snapshot EBS, menulis data langsung ke snapshot Anda, membaca data dari snapshot Anda, dan mengidentifikasi perbedaan atau perubahan antara dua snapshot. Untuk informasi selengkapnya, lihat Menggunakan API langsung EBS untuk mengakses konten snapshot EBS.
- Recycle Bin — Layanan pemulihan data yang memungkinkan Anda memulihkan snapshot EBS yang terhapus secara tidak sengaja dan AMI yang didukung EBS. Untuk informasi selengkapnya, lihat Recycle Bin.

Mengakses Amazon EBS

Anda dapat membuat dan mengelola sumber daya Amazon EBS menggunakan antarmuka berikut:

Konsol Amazon EC2

Antarmuka web untuk membuat dan mengelola volume dan snapshot. Jika Anda telah mendaftar untuk sebuah AWS akun, Anda dapat mengakses konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.

AWS Command Line Interface

Alat baris perintah yang memungkinkan Anda mengelola sumber daya Amazon EBS menggunakan perintah di shell baris perintah Anda. Hal ini didukung di Windows, Mac, dan Linux. Untuk informasi selengkapnya, lihat Panduan AWS Command Line Interface Pengguna dan Referensi AWS CLI Perintah.

AWS Tools for PowerShell

Satu set PowerShell modul yang memungkinkan Anda untuk menjalankan skrip pada sumber daya Amazon EBS Anda dari baris PowerShell perintah. Untuk informasi selengkapnya, lihat Panduan AWS Tools for Windows PowerShell Pengguna dan Referensi AWS Tools for PowerShell Cmdlet.

AWS CloudFormation

AWS Layanan terkelola sepenuhnya yang memungkinkan Anda membuat templat JSON atau YAMB yang dapat digunakan kembali yang menjelaskan AWS sumber daya Anda, lalu menyediakan dan mengonfigurasi sumber daya tersebut untuk Anda. Untuk informasi selengkapnya, silakan lihat Panduan Pengguna AWS CloudFormation.

API Kueri Amazon EC2

API Kueri Amazon EC2 menyediakan permintaan HTTP atau HTTPS yang menggunakan kata kerja HTTP `GET` atau `POST` dan parameter kueri bernama. `Action` Untuk informasi selengkapnya, lihat Referensi API Amazon EC2.

AWS SDK

API khusus bahasa yang memungkinkan Anda membangun aplikasi yang terintegrasi dengan AWS layanan. AWS SDK tersedia untuk banyak bahasa pemrograman populer. Untuk informasi selengkapnya, lihat Alat untuk Dibangun AWS.

Harga

Dengan Amazon EBS, Anda hanya membayar atas apa yang Anda berikan. Untuk informasi selengkapnya, lihat harga Amazon EBS.

Siapkan untuk Amazon EBS

Selesaikan tugas di bagian ini untuk menyiapkan diri untuk bekerja dengan sumber daya Amazon EBS.

Tugas

- Mendaftar untuk Akun AWS
- Buat pengguna dengan akses administratif
- (Opsional) Membuat dan menggunakan kunci terkelola pelanggan untuk enkripsi Amazon EBS
- (Opsional) Aktifkan blokir akses publik untuk snapshot Amazon EBS

Mendaftar untuk Akun AWS

Jika Anda tidak memiliki Akun AWS, selesaikan langkah-langkah berikut untuk membuatnya.

Untuk mendaftar untuk Akun AWS

1. Buka https://portal.aws.amazon.com/billing/signup.
2. Ikuti petunjuk online.

   Bagian dari prosedur pendaftaran melibatkan tindakan menerima panggilan telepon dan memasukkan kode verifikasi di keypad telepon.

   Saat Anda mendaftar untuk sebuah Akun AWS, sebuah Pengguna root akun AWSdibuat. Pengguna root memiliki akses ke semua Layanan AWS dan sumber daya di akun. Sebagai praktik keamanan terbaik, tetapkan akses administratif ke pengguna, dan gunakan hanya pengguna root untuk melakukan tugas yang memerlukan akses pengguna root.

AWS mengirimi Anda email konfirmasi setelah proses pendaftaran selesai. Anda dapat melihat aktivitas akun Anda saat ini dan mengelola akun Anda dengan mengunjungi https://aws.amazon.com/ dan memilih Akun Saya.

Buat pengguna dengan akses administratif

Setelah Anda mendaftar Akun AWS, amankan Pengguna root akun AWS, aktifkan AWS IAM Identity Center, dan buat pengguna administratif sehingga Anda tidak menggunakan pengguna root untuk tugas sehari-hari.

Amankan Anda Pengguna root akun AWS

1. Masuk ke AWS Management Consolesebagai pemilik akun dengan memilih pengguna Root dan memasukkan alamat Akun AWS email Anda. Di laman berikutnya, masukkan kata sandi.

   Untuk bantuan masuk dengan menggunakan pengguna root, lihat Masuk sebagai pengguna root di AWS Sign-In Panduan Pengguna.
2. Mengaktifkan autentikasi multi-faktor (MFA) untuk pengguna root Anda.

   Untuk petunjuk, lihat Mengaktifkan perangkat MFA virtual untuk pengguna Akun AWS root (konsol) Anda di Panduan Pengguna IAM.

Buat pengguna dengan akses administratif

1. Aktifkan Pusat Identitas IAM.

   Untuk mendapatkan petunjuk, silakan lihat Mengaktifkan AWS IAM Identity Center di Panduan Pengguna AWS IAM Identity Center .
2. Di Pusat Identitas IAM, berikan akses administratif ke pengguna.

   Untuk tutorial tentang menggunakan Direktori Pusat Identitas IAM sebagai sumber identitas Anda, lihat Mengkonfigurasi akses pengguna dengan default Direktori Pusat Identitas IAM di Panduan AWS IAM Identity Center Pengguna.

Masuk sebagai pengguna dengan akses administratif

- Untuk masuk dengan pengguna Pusat Identitas IAM, gunakan URL masuk yang dikirim ke alamat email saat Anda membuat pengguna Pusat Identitas IAM.

  Untuk bantuan masuk menggunakan pengguna Pusat Identitas IAM, lihat Masuk ke portal AWS akses di Panduan AWS Sign-In Pengguna.

Tetapkan akses ke pengguna tambahan

1. Di Pusat Identitas IAM, buat set izin yang mengikuti praktik terbaik menerapkan izin hak istimewa paling sedikit.

   Untuk petunjuknya, lihat Membuat set izin di Panduan AWS IAM Identity Center Pengguna.
2. Tetapkan pengguna ke grup, lalu tetapkan akses masuk tunggal ke grup.

   Untuk petunjuk, lihat Menambahkan grup di Panduan AWS IAM Identity Center Pengguna.

(Opsional) Membuat dan menggunakan kunci terkelola pelanggan untuk enkripsi Amazon EBS

Enkripsi Amazon EBS adalah solusi enkripsi yang menggunakan kunci AWS KMS kriptografi untuk mengenkripsi volume Amazon EBS dan snapshot Amazon EBS Anda. Amazon EBS secara otomatis membuat kunci KMS AWS terkelola unik untuk enkripsi Amazon EBS di setiap Wilayah. Kunci KMS ini memiliki alias `aws/ebs`. Anda tidak dapat memutar kunci KMS default atau mengelola izinnya. Untuk fleksibilitas dan kontrol yang lebih besar atas kunci KMS yang digunakan untuk enkripsi Amazon EBS, Anda dapat mempertimbangkan untuk membuat dan menggunakan kunci yang dikelola pelanggan.

Untuk membuat dan menggunakan kunci terkelola pelanggan untuk enkripsi Amazon EBS

1. Buat kunci KMS enkripsi simetris.
2. Pilih kunci KMS sebagai kunci KMS default untuk enkripsi Amazon EBS.
3. Berikan izin kepada pengguna untuk menggunakan kunci KMS untuk enkripsi Amazon EBS.

(Opsional) Aktifkan blokir akses publik untuk snapshot Amazon EBS

Untuk mencegah berbagi snapshot secara publik, Anda sekarang dapat mengaktifkan blokir akses publik untuk snapshot. Setelah Anda mengaktifkan blokir akses publik untuk snapshot di Wilayah, setiap upaya untuk membagikan snapshot secara publik di Wilayah tersebut akan diblokir secara otomatis. Pengaktifan ini dapat membantu Anda meningkatkan keamanan snapshot dan untuk melindungi data snapshot Anda dari akses yang tidak terotorisasi atau tidak diinginkan.

Untuk informasi selengkapnya, lihat Memblokir akses publik untuk snapshot.

Console

Untuk mengaktifkan blokir akses publik untuk snapshot

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Dasbor EC2, lalu di Atribut akun (di sisi kanan), pilih Perlindungan dan keamanan data.
3. Di bagian Blokir akses publik untuk snapshot EBS, pilih Kelola.
4. Pilih Blokir akses publik dan kemudian pilih salah satu opsi berikut:
   - Blokir semua akses publik — Untuk memblokir semua pembagian snapshot untuk publik. Pengguna di akun tidak dapat meminta berbagi publik baru. Selain itu, snapshot yang sudah dibagikan secara publik diperlakukan sebagai privat dan tidak lagi tersedia secara publik.
   - Blokir semua pembagian publik — Untuk memblokir hanya pembagian snapshot untuk publik baru. Pengguna di akun tidak dapat meminta berbagi publik baru. Namun, snapshot yang sudah dibagikan secara publik, tetap tersedia untuk umum.
5. Pilih Perbarui.

AWS CLI

Untuk mengaktifkan blokir akses publik untuk snapshot

Gunakan perintah enable-snapshot-block-public-access. Untuk `--state`, tentukan salah satu dari nilai-nilai berikut:

- `block-all-sharing` — Untuk memblokir semua pembagian snapshot untuk publik. Pengguna di akun tidak dapat meminta berbagi publik baru. Selain itu, snapshot yang sudah dibagikan secara publik diperlakukan sebagai privat dan tidak lagi tersedia secara publik.
- `block-new-sharing` — Untuk memblokir hanya pembagian snapshot untuk publik baru. Pengguna di akun tidak dapat meminta berbagi publik baru. Namun, snapshot yang sudah dibagikan secara publik, tetap tersedia untuk publik.

```
aws ec2 enable-snapshot-block-public-access --state block-all-sharing|block-new-sharing
```

Volume Amazon EBS

Volume Amazon EBS adalah perangkat penyimpanan tingkat blok yang tahan lama yang dapat Anda pasang ke instans Anda. Setelah Anda memasang volume ke suatu instans, Anda dapat menggunakannya seperti Anda menggunakan hard drive fisik. Volume EBS bersifat fleksibel. Untuk volume generasi saat ini yang dipasang ke tipe instans generasi saat ini, Anda dapat secara dinamis meningkatkan ukuran, mengubah kapasitas IOPS yang tersedia, dan mengubah tipe volume pada volume produksi langsung.

Anda dapat menggunakan volume EBS sebagai penyimpanan utama untuk data yang memerlukan pembaruan rutin, seperti drive sistem untuk instans atau penyimpanan aplikasi basis data. Anda juga dapat menggunakannya untuk aplikasi yang membutuhkan banyak throughput dengan melakukan pemindaian disk secara terus-menerus. Volume EBS tetap bertahan secara independen dari siklus hidup instans EC2.

Anda dapat memasang beberapa volume EBS ke suatu instans. Volume dan instans harus berada dalam Zona Ketersediaan yang sama. Bergantung pada jenis volume dan instans, Anda dapat menggunakan Multi-Lampirkan untuk memasang volume ke beberapa instans secara bersamaan.

Amazon EBS menyediakan tipe volume berikut ini: SSD Tujuan Umum (`gp2` dan `gp3`), SSD IOPS yang Tersedia (`io1` dan `io2`), HDD Throughput Dioptimalkan (`st1`), dan Cold HDD (`sc1`), dan Magnetik (`standard`). Semuanya berbeda dalam karakteristik dan harga performa, memungkinkan Anda menyesuaikan performa dan biaya penyimpanan sesuai kebutuhan aplikasi Anda. Untuk informasi selengkapnya, lihat Tipe volume Amazon EBS.

Akun Anda memiliki batas total penyimpanan yang tersedia untuk Anda. Untuk informasi selengkapnya tentang pembatasan ini, dan cara meminta peningkatan dalam pembatasan Anda, lihat Titik akhir dan kuota Amazon EBS.

Untuk informasi selengkapnya tentang harga, lihat Harga Amazon EBS.

Daftar Isi

- Ganti volume Amazon EBS menggunakan snapshot sebelumnya
- Pantau volume Amazon EBS Anda
- Pengujian kesalahan pada Amazon EBS

Keuntungan menggunakan volume EBS

Volume EBS memberikan manfaat yang tidak disediakan oleh volume penyimpanan instans.

Manfaat

- Ketersediaan data
- Persistensi data
- Enkripsi data
- Keamanan data
- Snapshot
- Fleksibilitas

Ketersediaan data

Saat Anda membuat volume EBS, volume itu secara otomatis direplikasi dalam Zona Ketersediaannya untuk mencegah kehilangan data karena kegagalan komponen perangkat keras apa pun. Anda dapat memasang volume EBS ke instans EC2 apa pun di Zona Ketersediaan yang sama. Setelah Anda memasang volume, perangkat muncul sebagai perangkat blok asli yang serupa dengan hard drive atau perangkat fisik lainnya. Pada saat itu, instansnya dapat berinteraksi dengan volume sama seperti dengan drive lokal. Anda dapat terhubung ke instance dan memformat volume EBS dengan sistem file, seperti `Ext4` untuk instance Linux atau `NTFS` untuk instance Windows, dan kemudian menginstal aplikasi.

Jika Anda memasang beberapa volume ke perangkat yang telah diberi nama, Anda dapat menghapus data di seluruh volume untuk peningkatan performa I/O dan throughput.

Anda dapat memasang volume EBS `io1` dan `io2` hingga 16 instans berbasis Nitro. Untuk informasi selengkapnya, lihat Melampirkan volume ke beberapa instans dengan Multi-Lampiran Amazon EBS. Jika tidak, Anda dapat memasang volume EBS ke satu instans.

Anda dapat memperoleh data pantauan untuk volume EBS Anda, termasuk volume perangkat root untuk instans yang didukung EBS, tanpa biaya tambahan. Untuk informasi selengkapnya tentang

cara memantau metrik, lihat CloudWatch Metrik Amazon untuk Amazon EBS. Untuk informasi tentang pelacakan status volume Anda, lihat Amazon EventBridge untuk Amazon EBS.

Persistensi data

Volume EBS adalah penyimpanan off-instans yang dapat bertahan secara terpisah dari kehidupan sebuah instans. Anda terus membayar penggunaan volume selama data tetap ada.

Volume EBS yang dipasang ke instans yang berjalan dapat secara otomatis terlepas dari instans dengan datanya utuh saat instans dihentikan jika Anda menghapus tanda centang pada kotak centang Hapus saat Pengakhiran saat Anda mengonfigurasi volume EBS untuk instans Anda di konsol EC2. Volume kemudian dapat disambungkan kembali ke instans baru, memungkinkan pemulihan cepat. Jika kotak centang untuk Hapus saat Pengakhiran dicentang, volume akan dihapus pada saat pengakhiran instans EC2. Jika Anda menggunakan instans yang didukung EBS, Anda dapat menghentikan dan memulai ulang laporan tersebut yang memengaruhi data yang disimpan dalam volume terlampir. Volume tetap terpasang selama siklus stop-start. Hal ini memungkinkan Anda untuk memproses dan menyimpan data pada volume Anda secara tidak terbatas, hanya menggunakan sumber daya pemrosesan dan penyimpanan saat diperlukan. Data tetap ada pada volume hingga volume dihapus secara eksplisit. Penyimpanan blok fisik yang digunakan oleh volume EBS yang dihapus ditimpa dengan nol atau data pseudorandom kriptografi sebelum dialokasikan ke volume baru. Jika Anda berurusan dengan data sensitif, Anda harus mempertimbangkan untuk mengenkripsi data Anda secara manual atau menyimpan data pada volume yang dilindungi oleh enkripsi Amazon EBS. Untuk informasi selengkapnya, lihat Enkripsi EBS Amazon.

Secara default, volume EBS utama yang dibuat dan dipasang pada suatu instans saat peluncuran akan dihapus bila instans tersebut diakhiri. Anda dapat mengubah perilaku ini dengan mengubah nilai bendera `DeleteOnTermination` ke `false` saat Anda meluncurkan instansnya. Nilai yang dimodifikasi ini menyebabkan volume berlanjut bahkan setelah instans dihentikan, dan memungkinkan Anda untuk memasang volume ke instans lain.

Secara default, volume EBS utama yang dibuat dan dipasang pada suatu instans saat peluncuran akan dihapus bila instans tersebut dihentikan. Anda dapat mengubah perilaku ini dengan mengubah nilai bendera `DeleteOnTermination` ke `true` saat Anda meluncurkan instansnya. Nilai yang dimodifikasi ini menyebabkan volume dihapus ketika instans diakhiri.

Enkripsi data

Untuk enkripsi data yang disederhanakan, Anda dapat membuat volume EBS terenkripsi dengan fitur enkripsi Amazon EBS. Semua tipe volume EBS mendukung enkripsi. Anda dapat menggunakan

volume EBS terenkripsi untuk memenuhi berbagai persyaratan data-at-rest enkripsi untuk data dan aplikasi yang diatur/diaudit. Enkripsi Amazon EBS menggunakan algoritma 256-bit Advanced Encryption Standard (AES-256) dan infrastruktur kunci utama yang dikelola Amazon. Enkripsi terjadi pada server yang menghosting instans EC2, menyediakan enkripsi data-in-transit dari instans EC2 ke penyimpanan Amazon EBS. Untuk informasi selengkapnya, lihat Enkripsi EBS Amazon.

Enkripsi Amazon EBS digunakan AWS KMS keys saat membuat volume terenkripsi dan snapshot apa pun yang dibuat dari volume terenkripsi Anda. Pertama kali Anda membuat volume EBS terenkripsi di Wilayah, kunci KMS AWS terkelola default dibuat untuk Anda secara otomatis. Kunci ini digunakan untuk enkripsi Amazon EBS kecuali Anda membuat dan menggunakan kunci yang dikelola pelanggan. Membuat kunci terkelola pelanggan Anda sendiri memberi Anda lebih banyak fleksibilitas, termasuk kemampuan untuk membuat, memutar, menonaktifkan, menentukan kontrol akses, dan mengaudit kunci enkripsi yang digunakan untuk melindungi data Anda. Untuk informasi selengkapnya, lihat Panduan Developer AWS Key Management Service.

Keamanan data

Volume Amazon EBS disajikan kepada Anda sebagai perangkat blok mentah yang tidak terformat. Perangkat-perangkat ini adalah perangkat logis yang dibuat pada infrastruktur EBS dan layanan Amazon EBS akan memastikan bahwa perangkat-perangkat tersebut secara logis kosong (yakni bahwa, blok mentah tersebut sudah dikosongkan atau mengandung data pseudorandom secara kriptografis) sebelum digunakan atau digunakan kembali oleh pelanggan.

Jika Anda memiliki prosedur yang mengharuskan semua data dihapus menggunakan metode tertentu, baik setelah atau sebelum digunakan (atau keduanya), seperti yang dirinci dalam DoD 5220.22-M (Manual Operasi Program Keamanan Industri Nasional) atau NIST 800-88(Pedoman untuk Sanitisasi Media), Anda memiliki kemampuan untuk melakukannya di Amazon EBS. Aktivitas tingkat blok tersebut akan tercermin ke media penyimpanan yang mendasarinya dalam layanan Amazon EBS tersebut.

Snapshot

Amazon EBS memberikan kemampuan untuk membuat snapshot (cadangan) dari volume EBS apa pun dan menulis salinan data dalam volume ke Amazon S3, yang menyimpan salinan itu secara redundan di beberapa Zona Ketersediaan. Volume tidak perlu dipasang ke instans yang sedang berjalan untuk mengambil snapshot. Saat Anda melanjutkan menulis data ke volume, Anda dapat membuat snapshot volume secara berkala untuk digunakan sebagai garis dasar untuk volume baru. Snapshot ini dapat digunakan untuk membuat banyak volume EBS baru atau memindahkan volume di seluruh Zona Ketersediaan. Snapshot volume EBS terenkripsi secara otomatis dienkripsi.

Saat Anda membuat volume baru dari snapshot, itu adalah salinan persis dari volume asli pada saat snapshot diambil. Volume EBS yang dibuat dari snapshot terenkripsi dienkripsi secara otomatis dienkripsi. Dengan menentukan Zona Ketersediaan yang berbeda, Anda dapat menggunakan fungsionalitas ini untuk membuat volume duplikat dalam zona tersebut. Snapshot dapat dibagikan dengan AWS akun tertentu atau dipublikasikan. Saat Anda membuat snapshot, Anda dikenai biaya di Amazon S3 berdasarkan ukuran data yang dicadangkan, bukan ukuran volume sumber. Snapshot berikutnya dengan volume yang sama adalah snapshot inkremental. Snapshot tersebut hanya menyertakan data yang diubah dan baru yang ditulis ke volume sejak snapshot terakhir dibuat, dan Anda hanya dikenai biaya untuk data yang diubah dan baru ini.

Snapshot adalah pencadangan bertahap, yang berarti hanya blok pada volume yang berubah setelah snapshot terbaru Anda disimpan. Jika Anda memiliki volume dengan 100 GiB data, tetapi hanya 5 GiB data telah berubah sejak snapshot terakhir Anda, hanya 5 GiB dari data yang dimodifikasi ditulis ke Amazon S3. Meskipun snapshot disimpan secara bertahap, proses penghapusan snapshot dirancang agar Anda hanya mempertahankan snapshot terbaru.

Untuk membantu mengategorikan dan mengelola volume dan snapshot, Anda dapat menandainya dengan metadata pilihan Anda.

Untuk mencadangkan volume Anda secara otomatis, Anda dapat menggunakan Amazon Data Lifecycle Manager atau AWS Backup.

Fleksibilitas

Volume EBS mendukung perubahan konfigurasi langsung saat berada di produksi. Anda dapat memodifikasi tipe volume, ukuran volume, dan kapasitas IOPS tanpa gangguan layanan. Untuk informasi selengkapnya, lihat Ubah volume menggunakan Amazon EBS Elastic Volumes.

Tipe volume Amazon EBS

Amazon EBS menyediakan tipe volume berikut, yang berbeda dalam karakteristik performa dan harga, sehingga Anda dapat menyesuaikan performa dan biaya penyimpanan dengan kebutuhan aplikasi Anda.

> **Important**
>
> Ada beberapa faktor yang dapat memengaruhi performa volume EBS, seperti konfigurasi instans, karakteristik I/O, dan permintaan beban kerja. Untuk sepenuhnya menggunakan IOPS yang disediakan pada volume EBS, gunakan instans yang dioptimalkan EBS. Untuk

informasi selengkapnya tentang memaksimalkan volume EBS Anda, lihat Performa volume Amazon EBS.

Untuk informasi selengkapnya tentang harga, lihat Harga Amazon EBS.

Tipe volume

- Volume solid state drive (SSD)
- Volume hard disk drive (HDD)
- Volume generasi sebelumnya

Volume solid state drive (SSD)

Volume yang didukung SSD dioptimalkan untuk beban kerja transaksional yang melibatkan operasi baca/tulis yang sering dilakukan dengan ukuran I/O kecil, dengan IOPS sebanga atribut performa yang dominan. Tipe volume yang didukung SSD termasuk SSD Tujuan Umum dan SSD IOPS yang Tersedia. Berikut ini ringkasan kasus penggunaan dan karakteristik volume yang didukung SSD.

	Volume SSD Tujuan Umum		Volume SSD IOPS yang tersedia	
Tipe volume	`gp3`	`gp2`	`io2` Block Express [3]	`io1`
Daya tahan	Daya tahan 99,8% - 99,9% (tingkat kegagalan tahunan 0,1% - 0,2%)		Daya tahan 99,999% (tingkat kegagalan tahunan 0,001%)	Daya tahan 99,8% - 99,9% (tingkat kegagalan tahunan 0,1% - 0,2%)
Kasus penggunaan	• Beban kerja transaksional • Desktop virtual • Basis data instans tunggal berukuran sedang • Aplikasi interaktif latensi rendah • Volume boot		Beban kerja yang membutuhkan: • Latensi Submilidetik • Performa IOPS yang berkelanjutan	• Beban kerja yang memerlukan performa IOPS berkelanjutan atau lebih dari 16.000 IOPS

	Volume SSD Tujuan Umum		Volume SSD IOPS yang tersedia	
	• Lingkungan pengembangan dan pengujian		• Lebih dari 64.000 IOPS atau 1.000 MiB/dtk throughput	• Beban kerja basis data intensif I/O
Ukuran volume	1 GiB - 16 TiB		4 GiB - 64 TiB [4]	4 GiB - 16 TiB
Maks IOPS per volume	16.000 (64 KiB I/O)	16.000 (16 KiB I/O)	256.000 (16 KiB I/O) 5	64.000 (16 KiB I/O)
Throughput maksimal per volume	1.000 MiB/dtk	250 MiB/dtk [1]	4.000 MiB/dtk	1.000 MiB/dtk [2]
Multi-Lampiran Amazon EBS	Tidak didukung		Didukung	
Reservasi NVMe	Tidak didukung		Didukung	Tidak didukung
Volume boot	Didukung			

[1] Batasan throughput adalah antara 128 MiB/dtk dan 250 MiB/dtk, tergantung pada ukuran volume. Untuk informasi selengkapnya, lihat Performa volume gp2. Volume yang dibuat sebelum 3 Desember 2018 yang belum dimodifikasi sejak pembuatan mungkin tidak mencapai performa penuh kecuali Anda mengubah volume.

[2] Untuk mencapai throughput maksimum 1.000 MiB/s, volume harus disediakan dengan 64.000 IOPS dan harus dilampirkan ke instance yang dibangun pada Sistem Nitro. Volume yang dibuat sebelum 6 Desember 2017 yang belum dimodifikasi sejak pembuatan mungkin tidak mencapai performa penuh kecuali Anda memodifikasi volumenya.

[3] Semua volume `io2` yang dibuat setelah 21 November 2023 adalah volume `io2` Block Express. Volume `io2` yang dibuat sebelum 21 November 2023 dapat dikonversi ke volume `io2` Block Express dengan memodifikasi IOPS atau ukuran volume.

[4] Volume berukuran lebih dari 16 TiB hanya dapat dilampirkan ke instans yang dibangun di Sistem Nitro.

[5] Volume lebih dari 64.000 IOPS hanya dapat dilampirkan ke instans yang dibangun di atas Sistem Nitro. Volume hingga 64.000 IOPS dapat dilampirkan ke instans non-Nitro, tetapi mereka hanya dapat mencapai hingga 32.000 IOPS.

Untuk informasi selengkapnya tentang tipe volume yang didukung SSD, lihat berikut ini:

- Volume SSD Tujuan Umum
- Volume SSD IOPS yang tersedia

Volume hard disk drive (HDD)

Volume yang didukung HDD dioptimalkan untuk beban kerja streaming besar di mana atribut performa dominan adalah throughput. Tipe volume HDD termasuk HDD dengan Throughput Dioptimalkan dan HDD Dingin Berikut ini ringkasan kasus penggunaan dan karakteristik volume yang didukung SSD.

	Volume HDD Throughput Dioptimalkan	Volume Cold HDD
Tipe volume	`st1`	`sc1`
Daya tahan	Daya tahan 99,8% - 99,9% (tingkat kegagalan tahunan 0,1% - 0,2%)	
Kasus penggunaan	• Big data • Gudang data • Pemrosesan log	• Penyimpanan berorientasi throughput untuk data yang jarang diakses

	Volume HDD Throughput Dioptimalkan	Volume Cold HDD
		• Skenario di mana biaya penyimpanan terendah adalah penting
Ukuran volume	125 GiB - 16 TiB	
Maks IOPS per volume (1 MiB I/O)	500	250
Throughput maksimal per volume	500 MiB/dtk	250 MiB/dtk
Multi-Lampiran Amazon EBS	Tidak didukung	
Volume boot	Tidak didukung	

Untuk informasi selengkapnya tentang volume Hard disk drive (HDD), lihat Volume HDD dengan throughput yang dioptimalkan dan Cold HDD.

Volume generasi sebelumnya

Volume magnetik (`standard`) adalah volume generasi sebelumnya yang didukung oleh drive magnetik. Mereka cocok untuk beban kerja dengan set data kecil di mana data jarang diakses dan performanya bukan merupakan hal yang penting. Volume ini menghasilkan sekitar 100 IOPS secara rata-rata, dengan kapasitas lonjakan hingga ratusan IOPS, dan ukurannya dapat berkisar antara 1 GiB hingga 1 TiB.

> **Tip**
>
> Magnetik adalah tipe volume generasi sebelumnya. Jika Anda membutuhkan performa atau konsistensi performa yang lebih tinggi dibandingkan volume generasi sebelumnya, sebaiknya gunakan salah satu tipe volume yang lebih baru.

Tabel berikut menjelaskan tipe volume EBS generasi sebelumnya.

	Magnetik
Tipe volume	`standard`
Kasus penggunaan	Beban kerja di mana data jarang diakses
Ukuran volume	1 GiB-1 TiB
Maks IOPS per volume	40–200
Throughput maksimal per volume	40–90 MiB/dtk
Volume boot	Didukung

Untuk informasi selengkapnya, lihat Volume Generasi Sebelumnya.

Volume SSD Tujuan Umum

Volume SSD Tujuan Umum (gp2 dan gp3) didukung oleh solid-state drive (SSD). Harga dan performa diseimbangkan untuk berbagai macam beban kerja transaksional. Ini termasuk desktop virtual, basis data instans tunggal berukuran sedang, aplikasi interaktif sensitif latensi, lingkungan pengembangan dan pengujian, dan volume boot. Kami merekomendasikan volume ini untuk sebagian besar beban kerja.

Amazon EBS menawarkan tipe volume SSD Tujuan Umum berikut:

Tipe

- Volume SSD Tujuan Umum (gp3)
- Volume SSD Tujuan Umum (gp2)

Volume SSD Tujuan Umum (gp3)

Volume SSD Tujuan Umum (gp3) adalah generasi terbaru dari volume SSD Tujuan Umum, dan volume SSD dengan biaya terendah yang ditawarkan oleh Amazon EBS. Tipe volume ini membantu memberikan keseimbangan harga dan performa yang tepat untuk sebagian besar aplikasi. Ini juga membantu Anda menskalakan performa volume secara independen dari ukuran volume. Ini

berarti Anda dapat menyediakan performa yang diperlukan tanpa perlu menyediakan kapasitas penyimpanan blok tambahan. Selain itu, volume gp3 menawarkan harga 20 persen lebih rendah per GiB daripada volume SSD Tujuan Umum (gp2).

Volume gp3 memberikan latensi milidetik satu digit dan 99,8 persen hingga 99,9 persen daya tahan volume dengan tingkat kegagalan tahunan (AFR) tidak lebih tinggi dari 0,2 persen, yang berarti maksimum dua kegagalan volume per 1.000 volume berjalan selama periode satu tahun. AWS mendesain volume gp3 untuk memberikan kinerja yang disediakan 99 persen dari waktu.

Daftar Isi

Performa volume gp3

> **Tip**
>
> Volume gp3 tidak menggunakan performa lonjakan. Volume ini dapat mempertahankan performa IOPS yang tersedia dan throughput secara penuh tanpa batas waktu.

Performa IOPS

Volume gp3 memberikan performa IOPS dasar yang konsisten 3.000 IOPS, yang disertakan dengan harga penyimpanan. Anda dapat menyediakan IOPS tambahan (hingga maksimum 16.000) dengan biaya tambahan dengan rasio 500 IOPS per GiB ukuran volume. IOPS maksimum dapat disediakan untuk volume 32 GiB atau lebih besar (500 IOPS per GiB × 32 GiB = 16.000 IOPS).

Performa throughput

Volume gp3 memberikan performa throughput acuan yang konsisten sebesar 125 MiB/dtk, yang disertakan dengan harga penyimpanan. Anda dapat menyediakan throughput tambahan (hingga maksimum 1.000 MiB/dtk) dengan biaya tambahan dengan rasio 0,25 MiB/dtk per IOPS yang tersedia. Throughput maksimum dapat dikondisikan pada 4.000 IOPS atau lebih tinggi dan 8 GiB atau lebih besar (4.000 IOPS × 0,25 MiB/dtk per IOPS = 1.000 MiB/dtk).

Ukuran volume gp3

Ukuran volume gp3 dapat bervariasi dari 1 GiB hingga 16 TiB.

Migrasi ke gp3 dari gp2

Jika saat ini Anda menggunakan volume gp2, Anda dapat memigrasikan volume ke gp3 menggunakan operasi Ubah volume menggunakan Amazon EBS Elastic Volumes. Anda dapat menggunakan operasi Volume Elastis Amazon EBS untuk mengubah tipe volume, IOPS, dan throughput volume yang ada tanpa mengganggu instans Amazon EC2. Saat menggunakan konsol untuk membuat volume atau membuat AMI dari snapshot, SSD Tujuan Umum gp3 adalah pilihan default untuk tipe volume. Dalam kasus lain, gp2 adalah pilihan default. Dalam kasus ini, Anda dapat memilih gp3 sebagai tipe volume alih-alih menggunakan gp2.

Untuk mengetahui berapa banyak yang dapat Anda hemat dengan memigrasikan volume gp2 Anda ke gp3, gunakan kalkulator penghematan biaya migrasi Amazon EBS gp2 ke gp3.

Volume SSD Tujuan Umum (gp2)

Volume SSD Tujuan Umum menawarkan penyimpanan hemat biaya yang ideal untuk berbagai beban kerja. Dengan volume gp2, performa diskalakan dengan ukuran volume.

> **Tip**
>
> Volume gp3 adalah generasi terbaru dari volume SSD Tujuan Umum. Volume itu menawarkan penskalaan performa yang lebih dapat diprediksi dan harga yang lebih murah hingga 20 persen daripada volume gp2. Untuk informasi selengkapnya, lihat Volume SSD Tujuan Umum (gp3).
> Untuk mengetahui berapa banyak yang dapat Anda hemat dengan memigrasikan gp2 volume ke gp3, gunakan kalkulator penghematan biaya migrasi Amazon EBS gp2 ke gp3.

gp2volume memberikan latensi milidetik satu digit dan 99,8 persen hingga 99,9 persen daya tahan volume dengan tingkat kegagalan tahunan (AFR) tidak lebih tinggi dari 0,2 persen, yang berarti maksimum dua kegagalan volume per 1.000 volume berjalan selama periode satu tahun. AWS mendesain gp2 volume untuk memberikan kinerja yang disediakan 99 persen dari waktu.

Daftar Isi

- Ukuran volume gp2

Performa volume **gp2**

Performa IOPS

Skala performa IOPS dasar secara linear antara minimal 100 dan maksimum 16.000 dengan kecepatan 3 IOPS per GiB ukuran volume. Performa IOPS disediakan sebagai berikut:

- Volume 33,33 GiB dan yang lebih kecil disediakan dengan minimal 100 IOPS.
- Volume lebih besar dari 33,33 GiB disediakan dengan 3 IOPS per GiB ukuran volume hingga batas 16.000 IOPS, yang dicapai pada 5.334 GiB (3 X 5.334).
- Volume 5.334 GiB dan lebih besar disediakan dengan 16.000 IOPS.

Volume gp2 yang lebih kecil dari 1 TiB (dan yang disediakan dengan kurang dari 3.000 IOPS) dapat melonjak menjadi 3.000 IOPS bila diperlukan untuk jangka waktu yang lama. Kemampuan volume untuk meledak diatur oleh kredit I/O. Jika permintaan I/O lebih besar dari performa dasar, volume menghabiskan kredit I/O untuk melonjak ke tingkat performa yang sesuai (hingga 3.000 IOPS). Saat melonjak, kredit I/O tidak diakumulasikan dan dihabiskan pada tingkat IOPS yang digunakan di atas IOPS dasar (tingkat pengeluaran = IOPS lonjakan - IOPS dasar). Semakin banyak kredit I/O yang diperoleh volume, semakin lama volume tersebut dapat mempertahankan performa lonjakannya. Anda dapat menghitung Durasi lonjakan sebagai berikut:

```
                      (I/O credit balance)
Burst duration  =  ------------------------------
                  (Burst IOPS) - (Baseline IOPS)
```

Ketika permintaan I/O turun ke tingkat performa dasar atau lebih rendah, volume mulai mendapatkan kredit I/O pada tingkat 3 kredit I/O per GiB ukuran volume per detik. Volume memiliki batas akrual kredit I/O sebesar 5,4 juta kredit I/O, yang cukup untuk mempertahankan performa lonjakan maksimum 3.000 IOPS selama setidaknya 30 menit.

> **Note**
>
> Setiap volume menerima saldo kredit I/O awal sebesar 5,4 juta kredit I/O, yang memberikan siklus boot awal cepat dalam volume boot dan pengalaman bootstrapping yang baik untuk aplikasi lain.

Tabel berikut mencantumkan contoh ukuran volume dan performa garis dasar terkait dari volume, durasi lonjakan (saat memulai dengan 5,4 juta kredit I/O), dan waktu yang diperlukan untuk mengisi ulang saldo kredit I/O kosong.

Ukuran volume (GiB)	Performa dasar (IOPS)	Durasi lonjakan pada 3.000 IOPS (detik)	Waktu untuk mengisi ulang saldo kredit kosong (detik)
1 hingga 33,33	100	1,862	54.000
100	300	2.000	18.000
334 (Ukuran minimum untuk throughput maksimal)	1,002	2,703	5,389
750	2,250	7.200	2,400
1.000	3.000	T/A*	T/A*
5.334 (ukuran minimum untuk IOPS maks) dan lebih besar	16.000	T/A*	T/A*

* Performa dasar volume melebihi performa lonjakan maksimum.

Anda dapat memantau saldo kredit I/O untuk volume menggunakan `BurstBalance` metrik Amazon EBS di Amazon. CloudWatch Metrik ini menunjukkan persentase kredit I/O untuk `gp2` yang tersisa. Untuk informasi selengkapnya, lihat Karakteristik dan pemantauan Amazon EBS I/O. Anda dapat mengatur alarm yang memberi tahu Anda kapan nilai `BurstBalance` turun ke tingkat tertentu. Untuk informasi selengkapnya, lihat Membuat CloudWatch Alarm.

Performa throughput

Volume `gp2` menghasilkan throughput antara 128 MiB/dtk dan 250 MiB/dtk, tergantung ukuran volume. Performa throughput disediakan sebagai berikut:

- Volume yang besarnya 170 GiB dan lebih kecil menghasilkan throughput maksimal 128 MiB/dtk.

- Volume lebih besar dari 170 GiB tetapi lebih kecil dari 334 GiB dapat melonjak hingga throughput maksimal 250 MiB/dtk.
- Volume sebesar 334 GiB dan lebih besar menghasilkan 250 MiB/dtk.

Throughput untuk volume gp2 dapat dihitung menggunakan rumus berikut, hingga batas throughput 250 MiB/dtk:

```
Throughput in MiB/s = IOPS performance × I/O size in KiB / 1,024
```

Ukuran volume **gp2**

Volume gp2 dapat berkisar dalam ukuran dari 1 GiB hingga 16 TiB. Perlu diingat bahwa performa volume menskalakan secara linier dengan ukuran volume.

Volume SSD IOPS yang tersedia

Volume SSD IOPS yang tersedia didukung oleh solid-state drive (SSD). Volume tersebut adalah volume penyimpanan Amazon EBS beperforma tertinggi yang dirancang untuk beban kerja kritis, intensif IOPS, dan intensif throughput yang memerlukan latensi rendah. Volume SSD IOPS yang tersedia memberikan performa IOPS yang tersedia 99,9 persen waktu.

Amazon EBS menawarkan dua tipe volume SSD IOPS yang tersedia:

- Volume Block Express SSD (io2) IOPS yang tersedia
- Volume SSD IOPS yang tersedia (io1)

Volume Block Express SSD (**io2**) IOPS yang tersedia

Volume Block Express io2 dibangun pada penyimpanan server arsitektur Amazon EBS generasi berikutnya. Ini telah dibangun untuk tujuan memenuhi persyaratan kinerja aplikasi intensif I/O yang paling menuntut yang berjalan pada instance yang dibangun di atas Sistem Nitro. Dengan daya tahan tertinggi dan latensi terendah, Block Express sangat ideal untuk menjalankan beban kerja yang intensif performa, misi kritis, seperti Oracle, SAP HANA, Microsoft SQL Server, dan SAS Analytics.

Arsitektur Block Express meningkatkan performa dan skala volume io2. Server Block Express berkomunikasi dengan instans yang dibangun di atas Sistem Nitro menggunakan protokol jaringan Scalable Reliable Datagram (SRD). Antarmuka ini diimplementasikan dalam Kartu Nitro yang dikhususkan untuk fungsi I/O Amazon EBS pada perangkat keras host instans. Ini meminimalkan

penundaan I/O dan variasi latensi (jitter jaringan), yang memberikan performa yang lebih cepat dan lebih konsisten untuk aplikasi Anda.

Volume `io2` Block Express dirancang untuk memberikan 99,999 persen daya tahan volume dengan tingkat kegagalan tahunan (AFR) tidak lebih dari 0,001 persen, yang berarti satu kegagalan volume per 100.000 volume berjalan selama periode satu tahun. `io2` Volume Block Express cocok untuk beban kerja yang mendapatkan keuntungan dari volume tunggal yang memberikan latensi sub-milidetik, mendukung IOPS dan throughput yang lebih tinggi, dan kapasitas yang lebih besar dari volume gp3.

Volume Block Express SSD IOPS yang tersedia (`io2`) memberikan performa IOPS yang Tersedia 99,9 persen waktu.

`io2`Volume Block Express didukung pada semua instans yang dibangun di Sistem Nitro. Untuk informasi selengkapnya, lihat `io2` Volume Blok Ekspres.

Topik

- Pertimbangan
- Kinerja

Pertimbangan

- Volume `io2` Block Express tersedia di Wilayah berikut: AS Timur (Ohio) | AS Timur (Virginia Utara) | AS Barat (California Utara) | AS Barat (Oregon) | Asia Pasifik (Hong Kong) | Asia Pasifik (Mumbai) | Asia Pasifik (Seoul) | Asia Pasifik (Singapura) | Asia Pasifik (Sydney) | Asia Pasifik (Tokyo) | Kanada (Pusat) | Eropa (Frankfurt) | Eropa (Irlandia) | Eropa (London) | Eropa (Stockholm) | Timur Tengah (Bahrain).
- Semua volume `io2` yang dibuat setelah 21 November 2023 adalah volume `io2` Block Express. Volume `io2` yang dibuat sebelum 21 November 2023 dapat dikonversi ke volume `io2` Block Express dengan memodifikasi IOPS atau ukuran volume.
- Instans yang dibangun di atas Sistem Nitro dapat dilampirkan ke volume hingga ukuran 64 TiB. Tipe instans lainnya dapat dilampirkan ke volume hingga 16 TiB dalam ukuran.
- Instans yang dibangun pada Sistem Nitro dapat dilampirkan ke volume yang disediakan hingga 256.000 IOPS. Tipe instans lainnya dapat dilampirkan ke volume yang disediakan hingga 64.000 IOPS, tetapi dapat mencapai hingga 32.000 IOPS.

- Untuk membuat volume `io2` terenkripsi, dengan ukuran lebih besar dari 16 TiB atau IOPS lebih besar dari 64.000, dari snapshot yang tidak terenkripsi atau snapshot terenkripsi bersama, Anda harus:
  1. Membuat salinan terenkripsi dari snapshot itu di akun Anda
  2. Menggunakan salinan snapshot itu untuk membuat volume

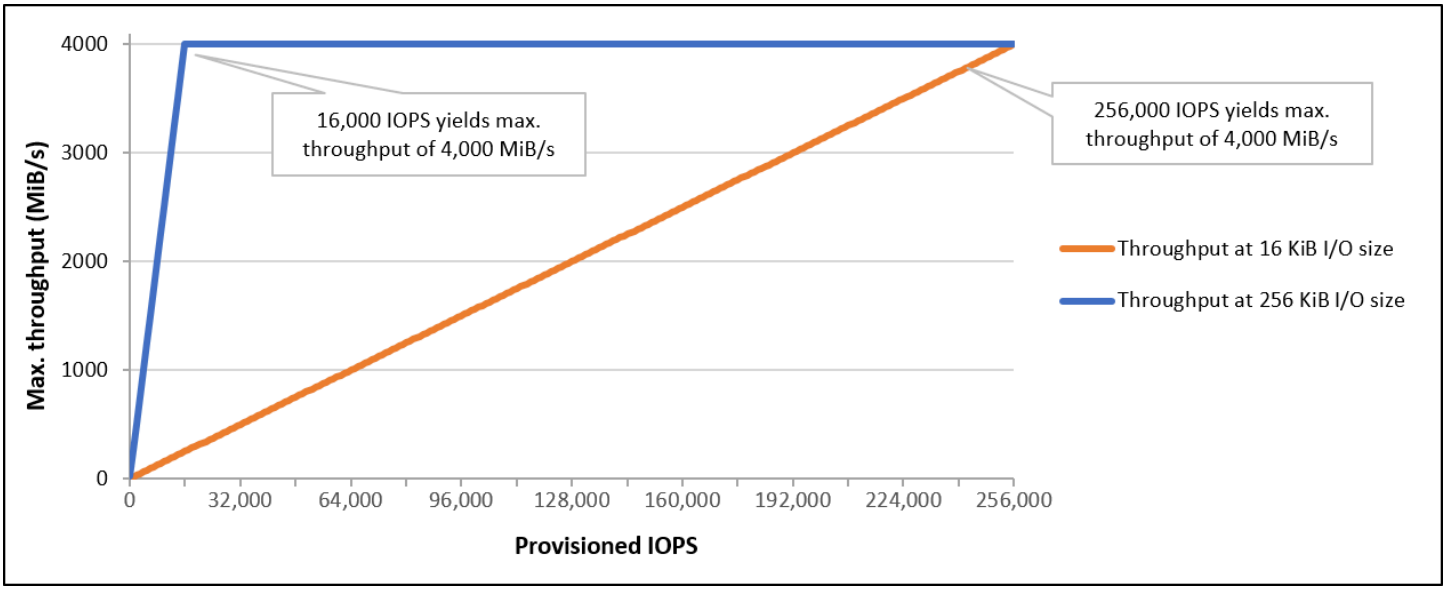
Kinerja

Dengan `io2` Block Express volume, Anda dapat menyediakan volume dengan:

- Latensi rata-rata sub-milidetik
- Kapasitas penyimpanan hingga 64 TiB (65.536 GiB)
- IOPS yang tersedia hingga 256.000, dengan rasio IOPS:GIB adalah 1.000:1. IOPS maksimum dapat disediakan dengan volume 256 GiB dan lebih besar (1.000 IOPS × 256 GiB = 256.000 IOPS).

> **Note**
>
> Anda dapat mencapai hingga 256.000 IOPS dengan instans yang dibangun di atas Sistem Nitro. Pada instans lain, Anda dapat mencapai performa hingga 32.000 IOPS.

- Volume throughput hingga 4.000 MiB/dtk. Throughput diskalakan secara proporsional hingga 0,256 MiB/dtk per IOPS yang tersedia. Throughput maksimum dapat dicapai pada 16.000 IOPS atau lebih tinggi.



Volume SSD IOPS yang tersedia (**`io1`**)

Volume SSD IOPS yang tersedia (`io1`) dirancang untuk memenuhi kebutuhan beban kerja intensif I/O, terutama beban kerja basis data, yang sensitif terhadap performa dan konsistensi penyimpanan. Volume SSD IOPS yang tersedia menggunakan tingkat IOPS yang konsisten, yang Anda tentukan saat membuat volume, dan Amazon EBS memberikan performa yang telah disediakan sebesar 99,9 persen.

`io1` volume dirancang untuk memberikan 99,8 hingga 99,9 persen daya tahan volume dengan tingkat kegagalan tahunan (AFR) tidak lebih dari 0,2 persen, yang berarti maksimum dua kegagalan volume per 1.000 volume berjalan selama periode satu tahun.

Volume `io1` tersedia untuk semua tipe instans Amazon EC2.

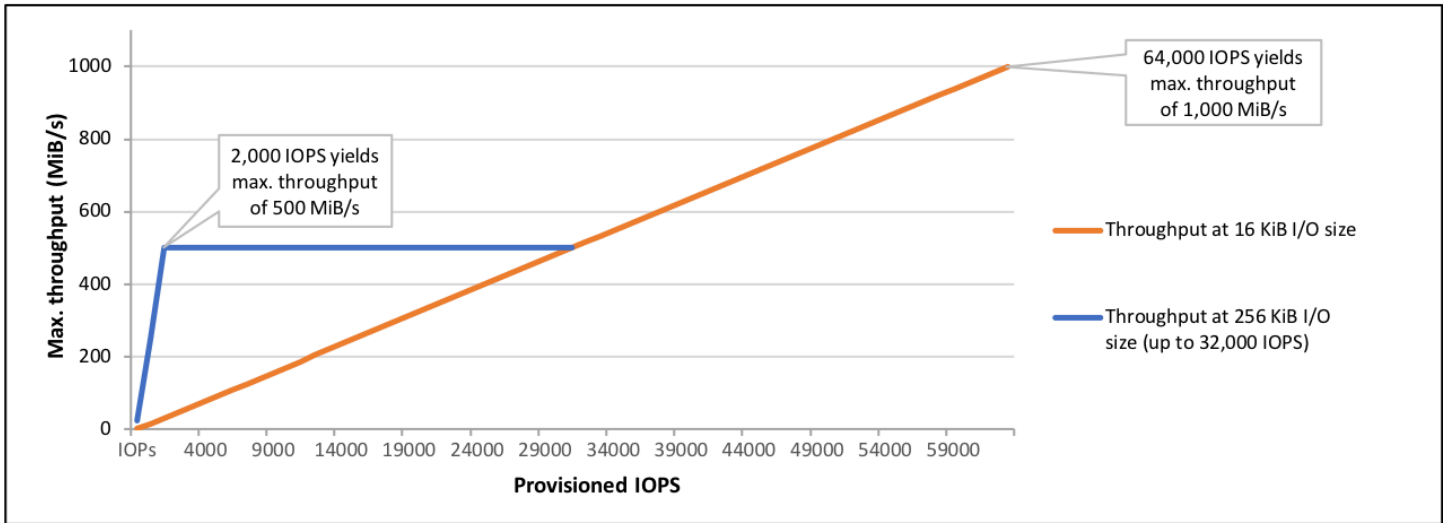
Kinerja

Ukuran volume `io1` dapat berkisar dari 4 GiB hingga 16 TiB dan Anda dapat menyediakan dari 100 IOPS hingga 64.000 IOPS per volume. Rasio maksimum IOPS yang tersedia dengan ukuran volume yang diminta (dalam GiB) adalah 50:1. Misalnya, volume `io1` 100 GiB dapat disediakan hingga 5.000 IOPS.

IOPS maksimum dapat ditetapkan untuk volume yang 1.280 GiB atau lebih besar (50 × 1.280 GiB = 64.000 IOPS).

- Volume `io1` yang disediakan hingga 32.000 IOPS mendukung ukuran I/O maksimum 256 KiB dan menghasilkan sebanyak 500 MiB/dtk throughput. Dengan ukuran I/O di maksimal, throughput puncak dicapai di 2.000 IOPS.
- Volume `io1` yang disediakan dengan lebih dari 32.000 IOPS (hingga maksimal 64.000 IOPS) menghasilkan peningkatan throughput linier pada tingkat 16 KiB per IOPS yang tersedia. Misalnya, volume yang disediakan dengan 48.000 IOPS dapat mendukung throughput hingga 750 MiB/dtk (16 KiB per IOPS yang tersedia x 48.000 IOPS yang tersedia = 750 MiB/dtk).
- Untuk mencapai throughput maksimum 1.000 MiB/dtk, volume harus disediakan dengan 64.000 IOPS (16 KiB per IOPS yang tersedia x 64.000 IOPS yang tersedia = 1.000 MiB/dtk).
- Anda dapat mencapai hingga 64.000 IOPS hanya pada instans yang dibangun di Sistem Nitro. Pada instans lain, Anda dapat mencapai performa hingga 32.000 IOPS.

. Grafik berikut menggambarkan karakteristik performa ini:



Pengalaman latensi per-I/O Anda tergantung pada IOPS yang tersedia dan profil beban kerja Anda. Untuk pengalaman latensi I/O terbaik, pastikan bahwa Anda menyediakan IOPS untuk memenuhi profil I/O beban kerja Anda.

Volume HDD dengan throughput yang dioptimalkan dan Cold HDD

Volume yang didukung HDD yang disediakan oleh Amazon EBS masuk ke dalam kategori berikut ini:

- HDD Throughput Dioptimalkan — HDD hemat biaya yang dirancang untuk beban kerja yang sering diakses dan membutuhkan banyak throughput.
- Cold HDD — Desain HDD hemat biaya untuk beban kerja yang jarang diakses.

Topik

Pembatasan pada throughput per-instans

Throughput untuk volume `st1` dan `sc1` selalu ditentukan oleh yang lebih kecil berikut ini:

- Batas throughput volume
- Batas throughput instans

Untuk seluruh volume Amazon EBS, kami sarankan Anda memilih instans EC2 yang sesuai dengan EBS untuk menghindari hambatan jaringan.

Volume HDD Throughput Dioptimalkan

Volume HDD Throughput yang Dioptimalkan (`st1`) menyediakan penyimpanan magnetik hemat biaya yang mendefinisikan performa dalam hal throighput daripada IOPS. Tipe volume ini cocok untuk beban kerja yang besar dan berurutan seperti Amazon EMR, ETL, gudang data, dan pemrosesan log. Volume `st1` yang dapat di-boot tidak didukung.

Volume HDD Throughput Dioptimalkan (`st1`), meskipun serupa dengan volume Cold HDD (`sc1`), dirancang untuk mendukung data yang sering diakses.

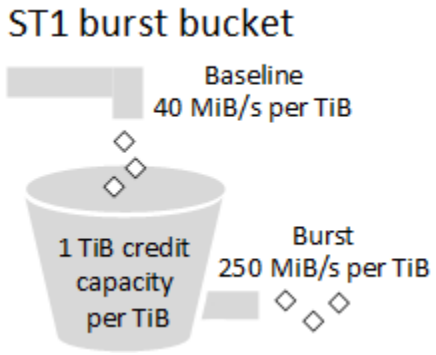
Tipe volume ini dioptimalkan untuk beban kerja yang melibatkan I/O berurutan yang besar, dan kami merekomendasikan agar pelanggan yang melakukan I/O kecil acak agar menggunakan `gp2`. Untuk informasi selengkapnya, lihat Inefisiensi baca/tulis kecil di HDD.

Volume HDD Throughput yang Dioptimalkan (`st1`) yang melekat pada instans yang dioptimalkan EBS dirancang untuk menawarkan performa yang konsisten, memberikan setidaknya 90 persen dari performa throughput yang diharapkan selalu sebesar 99 persen pada tahun tertentu.

Kredit throughput dan performa lonjakan

Seperti gp2, st1 menggunakan model bucket lonjakan untuk performa. Ukuran volume menentukan throughput tingkat dasar volume Anda, yang merupakan tingkat di mana volume mengakumulasi kredit throughput. Ukuran volume juga menentukan throughput lonjakan volume Anda, yang merupakan tingkat di mana Anda dapat menghabiskan kredit saat tersedia. Volume yang lebih besar memiliki garis dasar dan throughput lonjakan yang lebih tinggi. Makin banyak kredit volume Anda, makin lama volume tersebut dapat mendorong I/O pada tingkat lonjakan.

Diagram berikut menunjukkan perilaku bucket lonjakan untuk st1.



Tergantung throughput dan batas kredit throughput, throughput volume st1 yang tersedia dinyatakan dengan rumus berikut:

```
(Volume size) × (Credit accumulation rate per TiB) = Throughput
```

Untuk volume st1 1-TiB, throughput lonjakan dibatasi menjadi 250 MiB/dtk, bucket terisi dengan kredit di 40 MiB/dtk, dan dapat menampung kredit hingga senilai 1 TiB.

Volume yang lebih besar menskalakan batas ini secara linier, dengan throughput dibatasi maksimal 500 MiB/dtk. Setelah bucket habis, throughput dibatasi sesuai tingkat dasar sebesar 40 MiB/dtk per TiB.

Ukuran volume berkisar dari 0,125 TiB hingga 16 TiB, throughput awal bervariasi dari 5 MiB/dtk hingga maksimum 500 MiB/dtk, yang dicapai di 12,5 TiB sebagai berikut:

```
            40 MiB/s
12.5 TiB × ---------- = 500 MiB/s
             1 TiB
```

Throughput lonjakan bervariasi dari 31 MiB/dtk hingga batas 500 MiB/dtk, yang dicapai pada 2 TiB sebagai berikut:

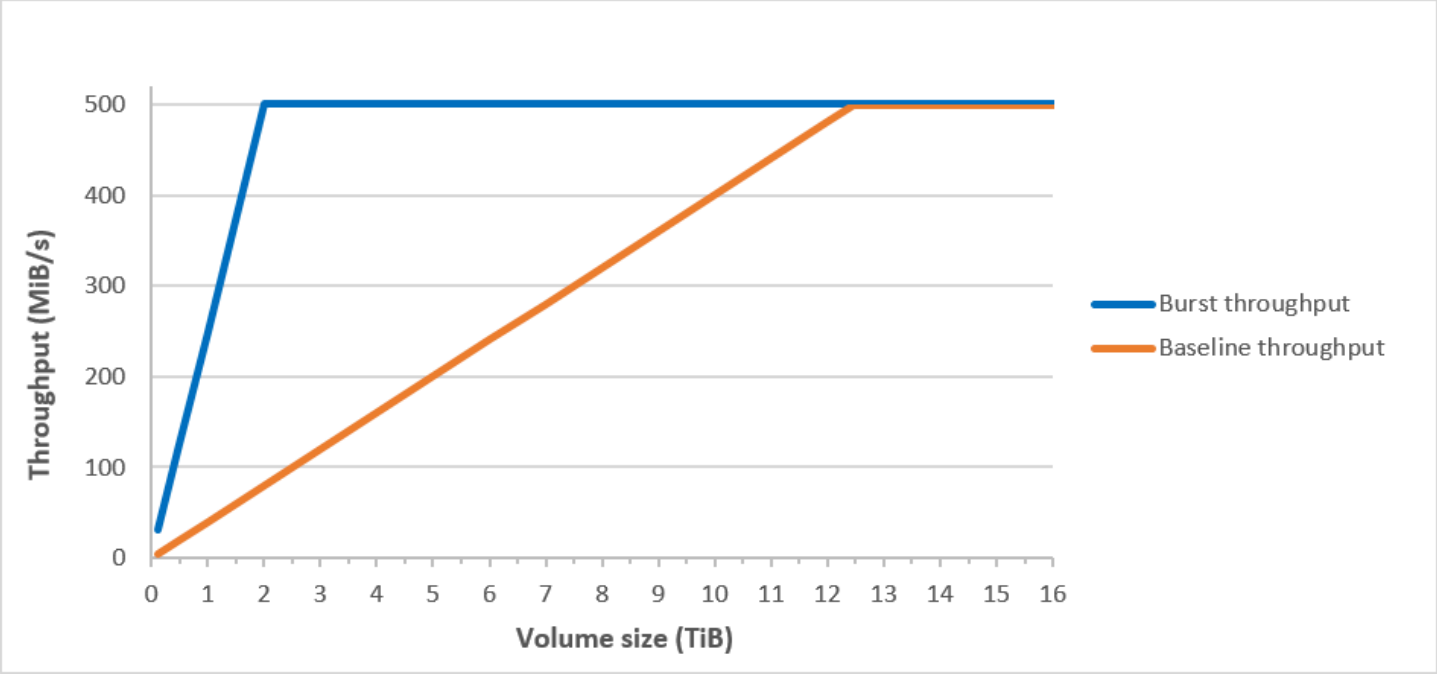
```
         250 MiB/s
2 TiB × ---------- = 500 MiB/s
          1 TiB
```

Tabel berikut ini menyatakan rentang lengkap nilai throughput dasar dan lonjakan untuk `st1`.

Ukuran volume (TiB)	Throughput dasar ST1 (MiB/dtk)	Throughput lonjakan ST1 (MiB/dtk)
0,125	5	31
0,5	20	125
1	40	250
2	80	500
3	120	500
4	160	500
5	200	500
6	240	500
7	280	500
8	320	500
9	360	500
10	400	500
11	440	500
12	480	500
12,5	500	500

Ukuran volume (TiB)	Throughput dasar ST1 (MiB/dtk)	Throughput lonjakan ST1 (MiB/dtk)
13	500	500
14	500	500
15	500	500
16	500	500

Diagram berikut membuat plot nilai tabel:



> **Note**
>
> Saat Anda membuat snapshot dari volume HDD Throughput Dioptimalkan (`st1`), performa dapat menurun sejauh nilai dasar volume saat snapshot sedang berlangsung.

Untuk informasi tentang penggunaan CloudWatch metrik dan alarm untuk memantau keseimbangan bucket burst Anda, lihat. Pantau saldo bucket lonjakan untuk volume

Volume Cold HDD

Volume Cold HDD (`sc1`) menyediakan penyimpanan magnetik hemat biaya yang mendefinisikan performa dalam hal throughput daripada IOPS. Dengan batas throughput yang lebih rendah dari `st1`, `sc1` cocok untuk beban kerja cold-data yang besar dan berurutan. Jika Anda memerlukan akses yang jarang ke data Anda dan ingin menghemat biaya, `sc1` menyediakan penyimpanan blok murah. Volume `sc1` yang dapat di-boot tidak didukung.

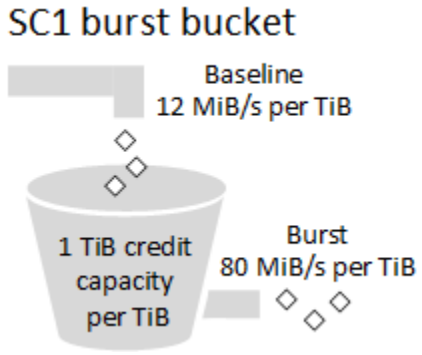
Volume Cold HDD (`sc1`), meskipun serupa dengan volume HDD Throughput Dioptimalkan (`st1`), dirancang untuk mendukung data yang jarang diakses.

> **Note**
>
> Tipe volume ini dioptimalkan untuk beban kerja yang melibatkan I/O berurutan yang besar, dan kami merekomendasikan agar pelanggan yang melakukan I/O kecil acak agar menggunakan gp2. Untuk informasi selengkapnya, lihat Inefisiensi baca/tulis kecil di HDD.

Volume Cold HDD (`sc1`) yang melekat pada instans yang dioptimalkan EBS dirancang untuk menawarkan performa yang konsisten, memberikan setidaknya 90 persen dari performa throughput yang diharapkan selalu 99 persen pada tahun tertentu.

Kredit throughput dan performa lonjakan

Seperti gp2, `sc1` menggunakan model bucket lonjakan untuk performa. Ukuran volume menentukan throughput tingkat dasar volume Anda, yang merupakan tingkat di mana volume mengakumulasi kredit throughput. Ukuran volume juga menentukan throughput lonjakan volume Anda, yang merupakan tingkat di mana Anda dapat menghabiskan kredit saat tersedia. Volume yang lebih besar memiliki garis dasar dan throughput lonjakan yang lebih tinggi. Makin banyak kredit volume Anda, makin lama volume tersebut dapat mendorong I/O pada tingkat lonjakan.



Tergantung pada batas throughput dan kredit throughput, throughput volume `sc1` dinyatakan dengan rumus berikut:

```
(Volume size) × (Credit accumulation rate per TiB) = Throughput
```

Untuk volume `sc1` 1-TiB, throughput lonjakan dibatasi menjadi 80 MiB/dtk, bucket terisi dengan kredit di 12 MiB/dtk, dan dapat menampung kredit hingga senilai 1 TiB.

Volume yang lebih besar menskalakan batas ini secara linier, dengan throughput dibatasi maksimum 250 MiB/dtk. Setelah bucket habis, throughput dibatasi sesuai tingkat dasar sebesar 12 MiB/dtk per TiB.

Ukuran volume berkisar dari 0,125 TiB hingga 16 TiB, throughput acuan bervariasi dari 1,5 MiB/dtk hingga maksimum 192 MiB/dtk, yang dicapai pada 16 TiB sebagai berikut:

```
           12 MiB/s
16 TiB × ---------- = 192 MiB/s
            1 TiB
```

Throughput lonjakan bervariasi dari 10 MiB/dtk hingga batas 250 MiB/dtk, yang dicapai pada 3,125 TiB sebagai berikut:

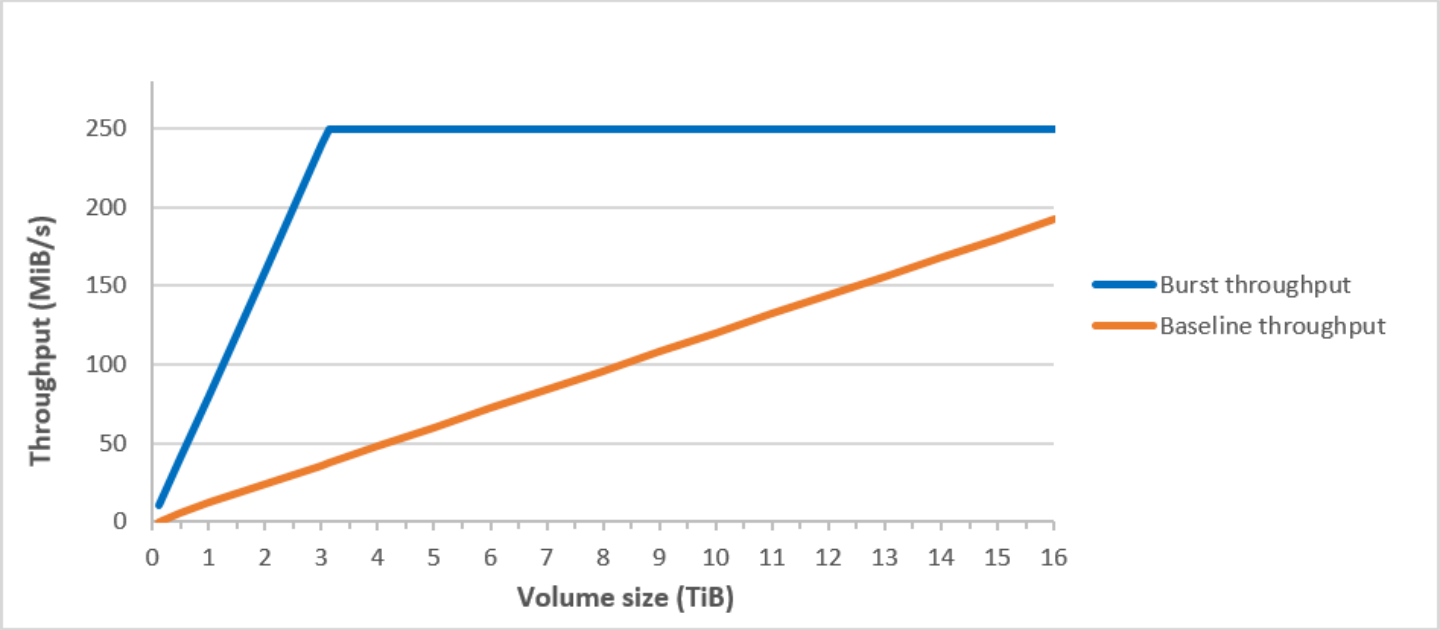
```
             80 MiB/s
3.125 TiB × ----------- = 250 MiB/s
              1 TiB
```

Tabel berikut ini menyatakan rentang lengkap nilai throughput dasar dan lonjakan untuk `sc1`:

Ukuran Volume (TiB)	Throughput Dasar SC1 (MiB/dtk)	Throughput Lonjakan SC1 (MiB/dtk)
0,125	1.5	10
0,5	6	40
1	12	80
2	24	160
3	36	240

Ukuran Volume (TiB)	Throughput Dasar SC1 (MiB/dtk)	Throughput Lonjakan SC1 (MiB/dtk)
3.125	37,5	250
4	48	250
5	60	250
6	72	250
7	84	250
8	96	250
9	108	250
10	120	250
11	132	250
12	144	250
13	156	250
14	168	250
15	180	250
16	192	250

Diagram berikut membuat plot nilai tabel:



> **Note**
>
> Saat Anda membuat snapshot volume Cold HDD (`sc1`), performa dapat menurun sejauh nilai dasar volume saat snapshot sedang berlangsung.

Untuk informasi tentang penggunaan CloudWatch metrik dan alarm untuk memantau keseimbangan bucket burst Anda, lihat. Pantau saldo bucket lonjakan untuk volume

Pertimbangan performa saat menggunakan volume HDD

Untuk hasil throughput optimal yang menggunakan volume HDD, rencanakan beban kerja Anda dengan mempertimbangkan hal-hal berikut.

Membandingkan HDD Throughput Dioptimalkan dan Cold HDD

Ukuran bucket `st1` dan `sc1` bervariasi sesuai dengan ukuran volume, dan bucket penuh berisi token yang cukup untuk pemindaian volume penuh. Namun, volume `st1` dan `sc1` yang lebih besar membutuhkan waktu yang lebih lama untuk menyelesaikan pemindaian volume karena batas throughput per-instans dan per-volume. Volume yang terpasang pada instans yang lebih kecil dibatasi berdasarkan pada throughput per instans daripada batas throughput `st1` atau `sc1`.

Keduanya `st1` dan `sc1` dirancang untuk konsistensi performa 90 persen dari hasil throughput lonjakan 99 persen. Periode yang tidak dipatuhi kurang lebih didistribusikan secara seragam, menargetkan 99 persen total throughput yang diharapkan setiap jam.

Secara umum, waktu pemindaian dinyatakan dengan rumus ini:

```
 Volume size
------------ = Scan time
 Throughput
```

Misalnya, mempertimbangkan jaminan konsistensi performa dan optimisasi lainnya, pelanggan `st1` dengan volume 5-TiB dapat melakukan pemindaian volume penuh dalam 2,91 hingga 3,27 jam.

- Waktu pemindaian optimal

```
   5 TiB            5 TiB
----------- = ------------------ = 10,486 seconds = 2.91 hours
 500 MiB/s     0.00047684 TiB/s
```

- Waktu pemindaian maksimal

```
  2.91 hours
-------------- = 3.27 hours
 (0.90)(0.99) <-- From expected performance of 90% of burst 99% of the time
```

Demikian pula, pelanggan `sc1` dengan volume 5TiB dapat melakukan pemindaian volume penuh dalam waktu 5,83 hingga 6,54 jam.

- Waktu pemindaian optimal

```
   5 TiB             5 TiB
----------- = ------------------- = 20972 seconds = 5.83 hours
 250 MiB/s     0.000238418 TiB/s
```

- Waktu pemindaian maksimal

```
  5.83 hours
-------------- = 6.54 hours
 (0.90)(0.99)
```

Tabel berikut menunjukkan waktu pemindaian ideal untuk volume berbagai ukuran, dengan asumsi bucket penuh dan throughput instans yang cukup.

Ukuran volume (TiB)	Waktu pemindaian ST1 dengan lonjakan (jam)*	Waktu pemindaian SC1 dengan lonjakan (jam)*
1	1.17	3,64
2	1.17	3,64
3	1,75	3,64
4	2.33	4.66
5	2,91	5.83
6	3.50	6,99
7	4.08	8.16
8	4.66	9.32
9	5.24	10.49
10	5.83	11.65
11	6.41	12.82
12	6,99	13,98
13	7.57	15.15
14	8.16	16.31
15	8.74	17.48
16	9.32	18.64

* Waktu pemindaian ini mengasumsikan kedalaman antrean rata-rata (dibulatkan ke bilangan bulat terdekat) sebesar empat atau lebih ketika melakukan I/O berurutan sebesar 1 MiB.

Oleh karena itu, jika Anda memiliki beban kerja berorientasi throughput yang perlu diselesaikan dengan cepat (hingga 500 MiB/dtk), atau memerlukan beberapa pemindaian volume penuh sehari,

gunakan `st1`. Jika Anda mengoptimalkan untuk biaya, data Anda relatif jarang diakses, dan Anda tidak memerlukan performa pemindaian lebih dari 250 MiB, gunakan `sc1`.

Inefisiensi baca/tulis kecil di HDD

Model performa untuk volume `st1` dan `sc1` dioptimalkan untuk I/O berurutan, mendukung beban kerja dengan throughput yang tinggi, menawarkan performa yang dapat diterima di beban kerja dengan campuran IOPS dan throughput yang sesuai, dan memisahkan beban kerja yang kecil, I/O acak.

Misalnya, permintaan I/O sebesar 1 MiB atau kurang dihitung sebagai kredit I/O 1 MiB. Namun, jika I/O bersifat berurutan, maka keduanya digabungkan menjadi blok I/O 1 MiB dan dihitung hanya sebagai kredit I/O 1 MiB.

Pantau saldo bucket lonjakan untuk volume

Anda dapat memantau tingkat burst bucket untuk `st1` dan `sc1` volume menggunakan `BurstBalance` metrik Amazon EBS yang tersedia di Amazon CloudWatch. Metrik ini menunjukkan kredit throughput untuk `st1` dan `sc1` yang tersisa di bucket lonjakan. Untuk informasi selengkapnya tentang `BurstBalance` metrik dan metrik lain yang terkait dengan I/O, lihat. Karakteristik dan pemantauan Amazon EBS I/O CloudWatch juga memungkinkan Anda untuk mengatur alarm yang memberi tahu Anda ketika `BurstBalance` nilainya jatuh ke tingkat tertentu. Untuk informasi selengkapnya, lihat Membuat CloudWatch Alarm.

Batasan ukuran dan konfigurasi volume EBS

Ukuran volume Amazon EBS dibatasi oleh fisika dan aritmatika penyimpanan data blok, serta oleh keputusan implementasi sistem operasi (OS) dan perancang sistem file. AWS memberlakukan batasan tambahan pada ukuran volume untuk menjaga keandalan layanannya.

Bagian-bagian berikut menjelaskan faktor terpenting yang membatasi ukuran volume EBS yang dapat digunakan dan menawarkan rekomendasi untuk mengonfigurasi volume EBS Anda.

Daftar Isi

- Ukuran blok data

Kapasitas penyimpanan

Tabel berikut merangkum jadwal penyimpanan teoretis dan yang diimplementasikan untuk sistem file yang paling umum digunakan di Amazon EBS, dengan asumsi ukuran blok sebesar 4.096 bita.

Skema pembagiar	Blok maksimal yang dapat dihitung	Ukuran maks teoretis (blok × ukuran blok)	Ekst4 menerapka n ukuran maksimal*	XFS menerapka n ukuran maksimal**	NTFS menerapka n ukuran maksimal	Max yang didukung oleh EBS
MBR	2^{32}	2 TiB	2 TiB	2 TiB	2 TiB	2 TiB
GPT	2^{64}	64 ZiB	1 EiB = 1024^2 TiB (50 TiB disertifi kasi pada RHEL7)	500 TiB (disertifikasi pada RHEL7)	256 TiB	64 TiB †

* https://ext4.wiki.kernel.org/index.php/Ext4_Howto dan https://access.redhat.com/solutions/1532

** https://access.redhat.com/solutions/1532

† Volume `io2` Block Express mendukung hingga 64 TiB untuk partisi GPT. Untuk informasi selengkapnya, lihat Volume Block Express SSD (`io2`) IOPS yang tersedia.

Pembatasan layanan

Amazon EBS merupakan abstrak penyimpanan pusat data yang didistribusikan secara besar-besaran ke dalam hard disk virtual. Ke sistem operasi yang terpasang pada instans EC2 volume EBS terlampir tampaknya berupa hard disk fisik yang berisi sektor disk 512-bita. OS tersebut mengelola alokasi blok (atau klaster) data ke sektor virtual tersebut melalui pemanfaatan manajemen penyimpanan. Alokasi tersebut sesuai dengan skema partisi volume, seperti master boot record

(MBR) atau GUID partition table (GPT), dan sesuai kemampuan sistem file yang terpasang (ext4, NTFS, dan seterusnya).

EBS tidak mengetahui data yang terkandung di sektor disk virtual; tapi hanya memastikan integritas sektor. Ini berarti bahwa AWS tindakan dan tindakan OS tidak tergantung satu sama lain. Saat Anda memilih ukuran volume, perhatikan kemampuan dan batasan keduanya, seperti dalam kasus berikut:

- Saat ini EBS mendukung ukuran volume maksimum 64 TiB. Artinya, Anda dapat membuat volume EBS sebesar 64 TiB, tetapi apakah OS tersebut mengakui semua kapasitas itu tergantung pada karakteristik desainnya sendiri dan bagaimana volumenya dipartisi.
- Volume boot harus menggunakan skema partisi MBR atau GPT. AMI yang Anda luncurkan instance menentukan mode boot dan selanjutnya skema partisi yang digunakan untuk volume boot.

  Dengan MBR, volume boot dibatasi hingga 2 TiB.

  Dengan GPT, volume boot dapat mencapai ukuran hingga 64 TiB saat digunakan dengan mode boot GRUB2 (Linux) atau UEFI (Windows).

  Untuk informasi selengkapnya, lihat Buat volume Amazon EBS tersedia untuk digunakan.
- Volume non-boot yang 2 TiB (2048 GiB) atau lebih besar harus menggunakan tabel partisi GPT untuk mengakses seluruh volume.

Skema partisi

Di antara dampak lainnya, skema pembagian menentukan berapa banyak blok data logis yang dapat ditangani secara unik dalam satu volume. Untuk informasi selengkapnya, lihat Ukuran blok data. Skema partisi umum yang digunakan adalah Master Boot Record (MBR) dan tabel partisi GUID (GPT). Perbedaan penting antara skema ini dapat dirangkum sebagai berikut.

MBR

MBR menggunakan struktur data 32-bit untuk menyimpan alamat blok. Ini berarti bahwa setiap blok data dipetakan dengan salah satu dari 2^{32} bilangan bulat yang mungkin. Ukuran maksimum volume yang dapat dihitung diberikan dengan formula berikut ini:

$$2^{32} \times \text{Block size}$$

Ukuran blok untuk volume MBR secara konvensional dibatasi sebesar 512 bita. Oleh karena itu:

$$2^{32} \times 512 \text{ bytes} = 2 \text{ TiB}$$

Solusi teknik untuk meningkatkan batas 2-TiB ini untuk volume MBR belum memenuhi adopsi industri yang tersebar luas. Akibatnya, Linux dan Windows tidak pernah mendeteksi volume MBR sebagai lebih besar dari 2 TiB bahkan AWS jika menunjukkan ukurannya menjadi lebih besar.

GPT

GPT menggunakan struktur data 64-bit untuk menyimpan alamat blok. Ini berarti bahwa setiap blok data dipetakan dengan salah satu dari 2^{64} bilangan bulat yang mungkin. Ukuran maksimum volume yang dapat dihitung diberikan dengan formula berikut ini:

$$2^{64} \times \text{Block size}$$

Ukuran blok untuk volume GPT umumnya 4.096 bita. Oleh karena itu:

$$\begin{aligned} & 2^{64} \times 4{,}096 \text{ bytes} \\ &= 2^{64} \times 2^{12} \text{ bytes} \\ &= 2^{70} \times 2^{6} \text{ bytes} \\ &= 64 \text{ ZiB} \end{aligned}$$

Sistem komputer dunia nyata tidak mendukung apa pun yang dekat dengan maksimum teoretis ini. Ukuran sistem file yang diterapkan saat ini dibatasi hingga 50 TiB untuk ext4 dan 256 TiB untuk NTFS.

Ukuran blok data

Penyimpanan data di hard drive modern dikelola melalui pengalamatan blok logis, lapisan abstraksi yang memungkinkan sistem operasi membaca dan menulis data dalam blok logis tanpa mengetahui banyak tentang perangkat keras yang mendasarinya. OS bergantung pada perangkat penyimpanan untuk memetakan blok-blok tersebut ke sektor fisik. EBS mengiklankan sektor 512-bita ke sistem operasi, yang membaca dan menulis data ke disk menggunakan blok data yang memiliki ukuran sektor ganda.

Ukuran default industri untuk blok data logis saat ini adalah 4.096 bita (4 KiB). Karena beban kerja tertentu mendapatkan keuntungan dari ukuran blok yang lebih kecil atau lebih besar, sistem file mendukung ukuran blok non-default yang dapat ditentukan selama pemformatan. Skenario di mana ukuran blok non-default harus digunakan berada di luar ruang lingkup topik ini, tetapi pilihan

ukuran blok memiliki konsekuensi bagi kapasitas penyimpanan volume. Tabel berikut menunjukkan kapasitas penyimpanan sebagai fungsi ukuran blok:

Ukuran blok	Ukuran volume maksimal
4 KiB (default)	16 TiB
8 KiB	32 TiB
16 KiB	64 TiB
32 KiB	128 TiB
64 KiB (maksimal)	256 TiB

Batas yang dikenakan EBS pada ukuran volume (64 TiB) saat ini setara dengan ukuran maksimum yang diaktifkan oleh blok data sebesar 16 KiB.

Amazon EBS dan NVMe

Volume EBS terekspos karena perangkat blok NVMe pada instans dibangun di Nitro System.

Panduan performa EBS yang disebutkan dalam Detail Produk Amazon EBS adalah valid dari antarmuka perangkat-blok.

Instans Linux

Nama perangkat adalah`/dev/nvme0n1`,`/dev/nvme1n1`, dan sebagainya. Nama perangkat yang Anda tentukan di pemetaan perangkat blok akan diganti menggunakan nama perangkat NVMe (`/dev/nvme[0-26]n1`). Driver perangkat blok dapat menentukan nama perangkat NVMe dengan urutan yang berbeda dari yang Anda tentukan untuk volume dalam pemetaan perangkat blok.

Instans Windows

Saat Anda memasang volume ke instans, Anda menyertakan nama perangkat untuk volume tersebut. Nama perangkat ini digunakan oleh Amazon EC2 Driver blokir perangkat untuk instans menetapkan nama volume aktual saat memasang volume, dan nama yang ditetapkan dapat berbeda dari nama yang digunakan Amazon EC2.

Daftar Isi

Instal atau tingkatkan driver NVMe

Untuk mengakses volume NVMe, driver NVMe harus diinstal. Instans dapat mendukung volume EBS NVMe, volume penyimpanan instans NVMe, kedua tipe volume NVMe, atau tanpa volume NVMe. Untuk informasi selengkapnya, lihat Ringkasan fitur jaringan dan penyimpanan.

Instans Linux

AMI berikut menyertakan driver NVMe yang diperlukan:

- Amazon Linux 2
- Amazon Linux AMI 2018.03
- Ubuntu 14.04 atau yang lebih baru dengan kernel `linux-aws`

> **Note**
>
> AWS Jenis instance berbasis Graviton memerlukan Ubuntu 18.04 atau yang lebih baru dengan kernel `linux-aws`

- Red Hat Enterprise Linux 6.5 atau lebih baru
- Red Hat Enterprise Linux 7.4 atau yang lebih baru
- SUSE Linux Enterprise Server 12 SP2 atau setelahnya
- CentOS 7.4.1708 atau setelahnya
- FreeBSD 11.1 atau setelahnya
- Debian GNU/Linux 9 atau setelahnya

Untuk mengonfirmasi bahwa instans Anda memiliki driver NVMe

Anda dapat mengonfirmasi bahwa instans Anda memiliki driver NVMe menggunakan perintah berikut.

- Amazon Linux, RHEL, CentOS, dan SUSE Linux Server

```
$ modinfo nvme
```

  Jika instans memiliki driver NVMe, perintah mengembalikan informasi tentang driver.

- Amazon Linux 2 dan Ubuntu

```
$ ls /sys/module/ | grep nvme
```

  Jika instans memiliki driver NVMe, perintah akan mengembalikan driver yang diinstal.

Untuk memperbarui driver NVMe

Jika instans Anda memiliki driver NVMe, Anda dapat memperbarui driver ke versi terbaru dengan menggunakan prosedur berikut.

1. Terhubung ke instans Anda.
2. Perbarui cache paket Anda untuk mendapatkan pembaruan paket yang diperlukan sebagai berikut.

   - Untuk Amazon Linux 2, Amazon Linux, CentOS, dan Red Hat Enterprise Linux:

     ```
     [ec2-user ~]$ sudo yum update -y
     ```

   - Untuk Ubuntu dan Debian:

     ```
     [ec2-user ~]$ sudo apt-get update -y
     ```

3. Ubuntu 16.04 dan yang lebih baru mencakup paket `linux-aws`, yang berisi driver NVMe dan ENA yang diwajibkan oleh instans berbasis Nitro. Mutakhirkan paket `linux-aws` untuk menerima versi terbaru sebagai berikut.

   ```
   [ec2-user ~]$ sudo apt-get install --only-upgrade -y linux-aws
   ```

   Untuk Ubuntu 14.04, Anda dapat menginstal paket `linux-aws` sebagai berikut:

   ```
   [ec2-user ~]$ sudo apt-get install linux-aws
   ```

4. Lakukan boot ulang instans untuk memuat versi kernel terbaru.

```
sudo reboot
```

5. Hubungkan kembali ke instans Anda setelah boot ulang.

Instans Windows

AMI AWS Windows untuk Windows Server 2008 R2 dan yang lebih baru termasuk driver AWS NVMe. Jika Anda tidak menggunakan AMI AWS Windows terbaru yang disediakan oleh Amazon, lihat Menginstal atau memutakhirkan driver AWS NVMe menggunakan PowerShell dalam Panduan Pengguna Amazon EC2.

Identifikasi perangkat EBS

EBS menggunakan virtualisasi I/O tunggal (SR-IOV) untuk menyediakan lampiran volume pada instans berbasis Nitro menggunakan spesifikasi NVMe. Perangkat ini mengandalkan driver NVMe standar pada sistem operasi. Driver ini biasanya menemukan perangkat terpasang selama boot instans, dan membuat simpul perangkat berdasarkan urutan respons perangkat, bukan pada cara perangkat ditentukan dalam pemetaan perangkat blok.

Instans Linux

<y>Di Linux, nama perangkat NVMe mengikuti pola`/dev/nvme<x>n<y>`, di mana <x>urutan enumerasi, dan, untuk EBS, adalah 1. Terkadang, perangkat dapat merespons penemuan dalam urutan yang berbeda di awal instans berikutnya, yang menyebabkan nama perangkat berubah. Selain itu, nama perangkat yang ditetapkan oleh driver perangkat blok dapat berbeda dari nama yang ditentukan dalam pemetaan perangkat blok.

Kami menyarankan agar Anda menggunakan pengidentifikasi stabil untuk volume EBS dalam instans Anda, seperti salah satu dari berikut ini:

- Untuk instans berbasis Nitro, pemetaan perangkat blok yang ditentukan di konsol Amazon EC2 ketika Anda memasang volume EBS atau selama Panggilan API `AttachVolume` atau `RunInstances` dicatat dalam bidang data khusus vendor pada identifikasi pengendali NVMe. Dengan AMI Amazon Linux setelah versi 2017.09.01, kami menyediakan aturan `udev` yang membaca data ini dan membuat tautan simbolis ke pemetaan perangkat blok.

- ID volume EBS dan titik pemasangan stabil di antara perubahan status instans. Nama perangkat NVMe dapat berubah tergantung pada urutan respons perangkat selama boot instans. Sebaiknya gunakan ID volume EBS dan titik pemasangan untuk identifikasi perangkat yang konsisten.
- Volume EBS NVMe memiliki ID volume EBS yang ditetapkan sebagai nomor seri dalam identifikasi perangkat. Gunakan perintah `lsblk -o +SERIAL` untuk mencantumkan nomor seri.
- Format nama perangkat NVMe dapat bervariasi tergantung pada apakah volume EBS dilampirkan selama atau setelah peluncuran instans. Nama perangkat NVMe untuk volume yang dilampirkan setelah peluncuran instans menyertakan prefiks `/dev/`, sedangkan nama perangkat NVMe untuk volume yang dilampirkan selama peluncuran instans tidak menyertakan prefiks `/dev/`. Jika Anda menggunakan AMI Amazon Linux atau FreeBSD, gunakan perintah `sudo ebsnvme-id /dev/nvme0n1 -u` untuk nama perangkat NVMe yang konsisten. Untuk distribusi lain, gunakan perintah `sudo nvme id-ctrl -v /dev/nvme0n1` untuk menentukan nama perangkat NVMe.
- Saat perangkat diformat, UUID akan dihasilkan yang akan bertahan selama masa pakai sistem file. Label perangkat dapat ditetapkan pada saat yang sama. Untuk informasi selengkapnya, lihat Buat volume Amazon EBS tersedia untuk digunakan dan Boot dari volume yang salah.

AMI Amazon Linux

Dengan AMI Amazon Linux 2017.09.01 atau yang lebih baru (termasuk Amazon Linux 2), Anda dapat menjalankan perintah ebsnvme-id sebagai berikut untuk memetakan nama perangkat NVMe ke ID volume dan nama perangkat:

Contoh berikut menunjukkan perintah dan output untuk volume yang dilampirkan selama peluncuran instans. Harap diperhatikan bahwa nama perangkat NVMe tidak menyertakan prefiks `/dev/`.

```
[ec2-user ~]$ sudo /sbin/ebsnvme-id /dev/nvme0n1
Volume ID: vol-01324f611e2463981
sda
```

Contoh berikut menunjukkan perintah dan output untuk volume yang dilampirkan setelah peluncuran instans. Harap diperhatikan bahwa nama perangkat NVMe menyertakan prefiks `/dev/`.

```
[ec2-user ~]$ sudo /sbin/ebsnvme-id /dev/nvme1n1
Volume ID: vol-064784f1011136656
/dev/sdf
```

Amazon Linux juga membuat tautan simbolik dari nama perangkat di pemetaan perangkat blok (misalnya, `/dev/sdf`), ke nama perangkat NVMe.

AMI FreeBSD

Memulai dengan FreeBSD 12.2-RELEASE, Anda dapat menjalankan perintah ebsnvme-id seperti yang ditunjukkan di atas. Berikan nama perangkat NVMe (misalnya `nvme0`) atau perangkat disk (misalnya `nvd0` atau `nda0`). FreeBSD juga membuat tautan simbolis ke perangkat disk (misalnya *`volume_id`* `/dev/aws/disk/ebs/`).

AMI Linux Lainnya

Dengan versi kernel 4.2 atau lebih baru, Anda dapat menjalankan perintah nvme id-ctrl sebagai berikut untuk memetakan perangkat NVMe ke ID volume. Pertama, instal paket baris perintah NVMe, `nvme-cli`, menggunakan alat manajemen paket untuk distribusi Linux Anda. Untuk petunjuk pengunduhan dan penginstalan untuk distribusi lainnya, lihat dokumentasi khusus untuk distribusi Anda.

Contoh berikut mendapatkan ID volume dan nama perangkat NVMe untuk volume yang terpasang saat peluncuran instans. Harap diperhatikan bahwa nama perangkat NVMe tidak menyertakan prefiks `/dev/`. Nama perangkat tersedia melalui ekstensi khusus vendor pengendali NVMe (bite 384:4095 dari identifikasi pengendali):

```
[ec2-user ~]$ sudo nvme id-ctrl -v /dev/nvme0n1
NVME Identify Controller:
vid     : 0x1d0f
ssvid   : 0x1d0f
sn      : vol01234567890abcdef
mn      : Amazon Elastic Block Store
...
0000: 2f 64 65 76 2f 73 64 6a 20 20 20 20 20 20 20 20 "sda..."
```

Contoh berikut mendapatkan ID volume dan nama perangkat NVMe untuk volume yang terpasang saat peluncuran instans. Harap diperhatikan bahwa nama perangkat NVMe menyertakan prefiks `/dev/`.

```
[ec2-user ~]$ sudo nvme id-ctrl -v /dev/nvme1n1
NVME Identify Controller:
vid     : 0x1d0f
ssvid   : 0x1d0f
sn      : volabcdef01234567890
mn      : Amazon Elastic Block Store
...
```

```
0000: 2f 64 65 76 2f 73 64 6a 20 20 20 20 20 20 20 20 "/dev/sdf..."
```

Perintah lsblk mencantumkan perangkat yang tersedia dan titik pemasangannya (jika ada). Ini membantu Anda menentukan nama perangkat yang tepat untuk digunakan. Dalam contoh ini, `/dev/nvme0n1p1` dipasang sebagai perangkat root dan `/dev/nvme1n1` dilampirkan tetapi tidak terpasang.

```
[ec2-user ~]$ lsblk
NAME          MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
nvme1n1       259:3   0  100G  0 disk
nvme0n1       259:0   0    8G  0 disk
  nvme0n1p1   259:1   0    8G  0 part /
  nvme0n1p128 259:2   0    1M  0 part
```

Instans Windows

Anda dapat menjalankan perintah **`ebsnvme-id`** untuk memetakan nomor disk perangkat NVMe untuk ID volume EBS dan nama perangkat. Secara default, semua perangkat NVMe EBS dienumerasi. Anda dapat melewatkan nomor disk untuk informasi enumerasi perangkat tertentu. `ebsnvme-id`Alat ini termasuk dalam AMI Windows Server terbaru yang AWS disediakan di`C:\PROGRAMDATA\AMAZON\Tools`.

Dimulai dengan paket `1.5.0`, driver AWS NVMe versi terbaru `ebsnvme-id` alat diinstal oleh paket driver. Versi terbaru hanya tersedia dalam paket driver. Tautan unduhan mandiri untuk alat `ebsnvme-id` ini tidak akan lagi menerima pembaruan. Versi terakhir yang tersedia melalui tautan mandiri adalah `1.1.0`, yang dapat diunduh menggunakan tautan ebsnvme-id.zip dan mengekstrak konten ke instans Amazon EC2 Anda untuk mendapatkan akses ke `ebsnvme-id.exe`.

```
PS C:\Users\Administrator\Desktop> ebsnvme-id.exe
Disk Number: 0
Volume ID: vol-0d6d7ee9f6e471a7f
Device Name: sda1

Disk Number: 1
Volume ID: vol-03a26248ff39b57cf
Device Name: xvdd

Disk Number: 2
Volume ID: vol-038bd1c629aa125e6
Device Name: xvde
```

```
Disk Number: 3
Volume ID: vol-034f9d29ec0b64c89
Device Name: xvdb

Disk Number: 4
Volume ID: vol-03e2dbe464b66f0a1
Device Name: xvdc
PS C:\Users\Administrator\Desktop> ebsnvme-id.exe 4
Disk Number: 4
Volume ID: vol-03e2dbe464b66f0a1
Device Name: xvdc
```

Bekerja dengan volume EBS NVMe

Untuk memformat dan memasang Volume EBS NVMe, lihat Buat volume Amazon EBS tersedia untuk digunakan.

Instans Linux

Jika Anda menggunakan Linux kernel 4.2 atau yang lebih baru, setiap perubahan yang Anda lakukan pada ukuran volume EBS NVMe secara otomatis diterapkan dalam instans tersebut. Untuk kernel Linux lama, Anda mungkin perlu memisahkan dan memasang volume EBS atau boot ulang instans agar perubahan ukuran dapat diterapkan. Dengan Linux kernel 3.19 atau versi yang lebih baru, Anda dapat menggunakan perintah hdparm sebagai berikut untuk memaksa pemindaian ulang perangkat NVMe:

```
[ec2-user ~]$ sudo hdparm -z /dev/nvme1n1
```

Saat Anda melepaskan volume EBS NVMe, instansnya tidak memiliki kesempatan untuk membersihkan cache sistem file atau metadata sebelum memisahkan volume. Oleh karena itu, sebelum Anda melepaskan Volume EBS NVMe, Anda harus terlebih dahulu menyikronkan dan melepaskannya. Jika volume tidak dapat dilepaskan, Anda dapat mencoba perintah `force-detach` seperti yang dijelaskan dalam Lepaskan volume Amazon EBS dari instans.

Instans Windows

AMI AWS Windows terbaru berisi driver AWS NVMe yang diperlukan oleh jenis instans yang mengekspos volume EBS sebagai perangkat blok NVMe. Namun, jika Anda mengubah ukuran volume root Anda di sistem Windows, Anda harus memindai ulang volume agar perubahan ini dapat diterapkan di dalam instans. Jika Anda meluncurkan instans dari AMI yang berbeda, instans tersebut

mungkin tidak berisi driver AWS NVMe yang diperlukan. Jika instans Anda tidak memiliki driver AWS NVMe terbaru, Anda harus menginstalnya. Untuk informasi selengkapnya, lihat driver AWS NVMe untuk instance Windows.

Waktu habis operasi I/O

Sebagian besar sistem operasi menentukan waktu habis untuk operasi I/O yang dikirimkan ke perangkat NVMe.

Instans Linux

Di Linux, volume EBS yang dilampirkan ke instance berbasis NITRO menggunakan driver NVMe default yang disediakan oleh sistem operasi. Sebagian besar sistem operasi menentukan waktu habis untuk operasi I/O yang dikirimkan ke perangkat NVMe. Waktu habis adalah 30 detik dan dapat diubah menggunakan parameter boot `nvme_core.io_timeout`. Untuk sebagian besar kernel Linux sebelum versi 4.6, parameter ini adalah `nvme.io_timeout`.

Jika latensi I/O melebihi nilai parameter waktu habis, maka driver NVMe Linux gagal dalam I/O dan mengembalikan kesalahan ke sistem file atau aplikasi. Bergantung pada operasi I/O, sistem file atau aplikasi Anda dapat mencoba kembali kesalahan tersebut. Dalam beberapa kasus, sistem file Anda mungkin dipasang ulang sebagai hanya-baca.

Untuk pengalaman yang serupa dengan volume EBS yang dilampirkan pada instans Xen, kami menyarankan agar mengatur `nvme_core.io_timeout` ke nilai tertinggi yang mungkin. Untuk kernel saat ini, maksimalnya adalah 4294967295, sedangkan untuk kernel sebelumnya maksimal adalah 255. Tergantung pada versi Linux, batas waktu mungkin sudah diatur ke nilai maksimum yang mendukung. Misalnya, batas waktu diatur ke 4294967295 secara default untuk AMI Amazon Linux 2017.09.01 dan yang lebih baru.

Anda dapat memverifikasi nilai maksimum untuk distribusi Linux Anda dengan menulis nilai yang lebih tinggi dari nilai maksimum hingga `/sys/module/nvme_core/parameters/io_timeout` yang disarankan dan memeriksa kesalahan Hasil numerik di luar rentang saat mencoba untuk menyimpan file.

Instans Windows

Pada Windows, batas waktu default adalah 60 detik dan maksimum adalah 255 detik. Anda dapat memodifikasi pengaturan registri kelas disk `TimeoutValue` menggunakan prosedur yang diuraikan dalam Entri Daftar untuk Driver SCSI Miniport.

Perintah Abort

Perintah `Abort` adalah perintah Admin NVMe yang dikeluarkan untuk membatalkan perintah tertentu yang sebelumnya dikirimkan ke pengendali. Perintah ini biasanya dikeluarkan oleh driver perangkat ke perangkat penyimpanan yang telah melampaui ambang batas waktu operasi I/O. Tipe instans Amazon EC2 yang mendukung perintah `Abort` secara default akan membatalkan perintah tertentu yang sebelumnya dikirimkan ke pengendali perangkat Amazon EBS terlampir di mana perintah `Abort` akan dikeluarkan.

Tipe instans berikut mendukung perintah `Abort` untuk semua volume Amazon EBS terlampir secara default: `R5b`, `R6i`, `M6i`, `M6a`, `C6gn`, `C6i`, `X2gd`, `X2iezn`, `Im4gn`, `Is4gen`.

Tipe instans lain tidak mengambil tindakan ketika perintah `Abort` dikeluarkan untuk volume Amazon EBS terlampir.

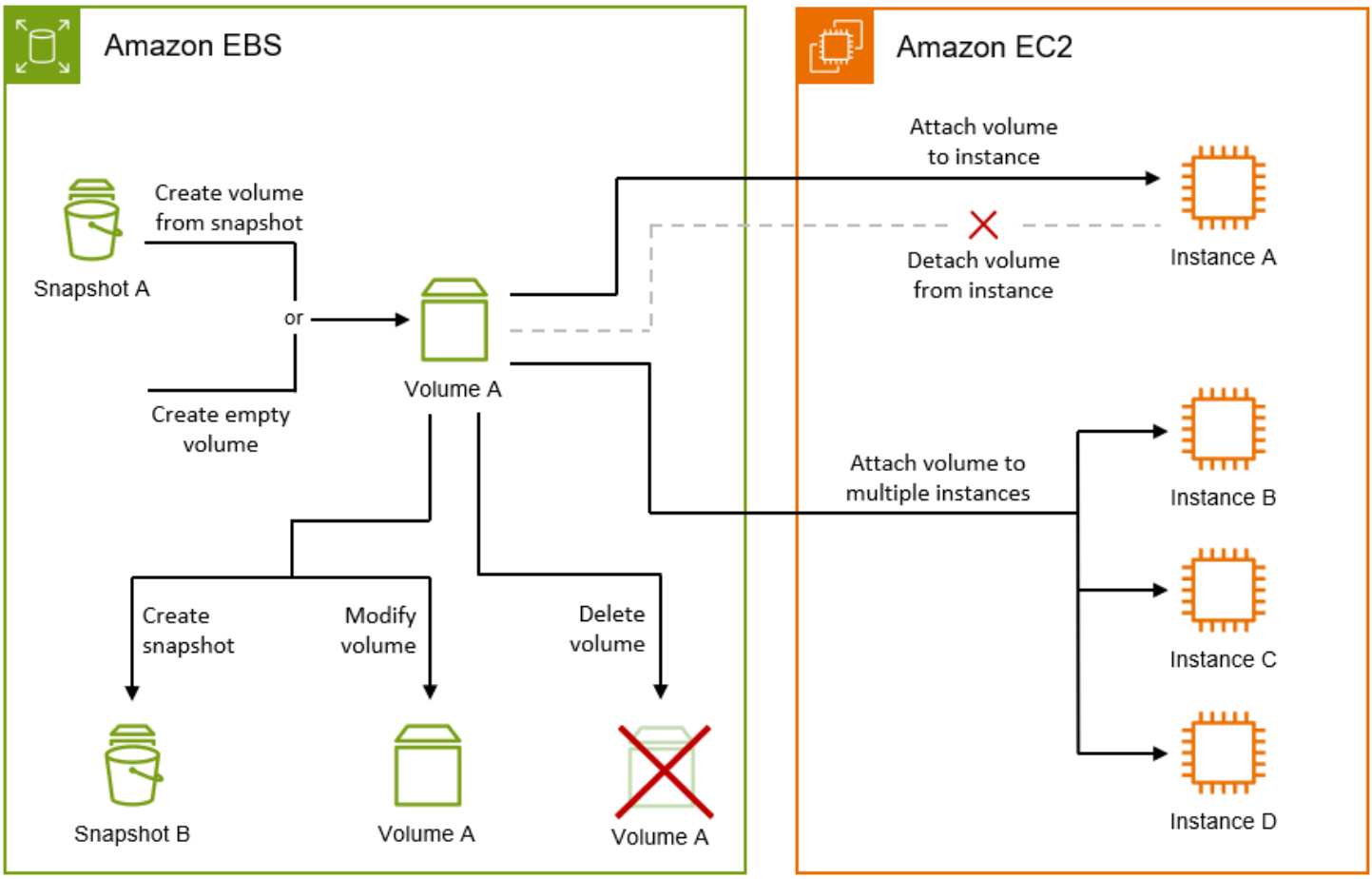
Perangkat Amazon EBS dengan perangkat NVMe versi `1.4` atau yang lebih tinggi mendukung perintah `Abort`.

Untuk informasi selengkapnya, lihat bagian 5.1 Membatalkan perintah pada Spesifikasi Dasar NVM Express.

Siklus hidup volume Amazon EBS

Siklus hidup volume Amazon EBS dimulai dengan proses pembuatan. Anda dapat membuat volume dari snapshot Amazon EBS atau Anda dapat membuat volume kosong. Sebelum Anda dapat menggunakan volume Anda, Anda harus melampirkannya ke satu atau beberapa instans Amazon EC2 yang berada di Availability Zone yang sama dengan volume. Anda dapat melampirkan beberapa volume ke sebuah instance. Jika diperlukan, Anda dapat melepaskan volume dari satu instance dan kemudian melampirkannya ke instance lain. Jika persyaratan penyimpanan Anda berubah, Anda dapat mengubah ukuran atau kinerja volume kapan saja. Anda dapat membuat point-in-time cadangan volume Anda dengan membuat snapshot Amazon EBS. Jika Anda tidak lagi membutuhkan volume, Anda dapat menghapusnya untuk berhenti menimbulkan biaya penyimpanan terkait.

Gambar berikut menunjukkan tindakan yang dapat Anda lakukan pada volume sebagai bagian dari siklus hidup volume.



Ada juga tugas yang Anda lakukan dengan menghubungkan ke instance dan menjalankan perintah sistem operasi. Misalnya, memformat volume, memasang volume, mengelola partisi, dan melihat ruang disk kosong.

Tugas

- Buat volume Amazon EBS
- Lampirkan volume Amazon EBS ke instans
- Melampirkan volume ke beberapa instans dengan Multi-Lampiran Amazon EBS
- Buat volume Amazon EBS tersedia untuk digunakan
- Melihat informasi tentang volume Amazon EBS
- Ubah volume menggunakan Amazon EBS Elastic Volumes
- Lepaskan volume Amazon EBS dari instans
- Menghapus volume Amazon EBS

Buat volume Amazon EBS

Anda dapat membuat sebuah volume Amazon EBS lalu melampirkannya ke instans EC2 apa pun di Zona Ketersediaan yang sama. Jika Anda membuat volume EBS terenkripsi, Anda hanya dapat memasangnya ke tipe instans yang didukung. Untuk informasi selengkapnya, lihat Tipe instans yang didukung.

Jika Anda sedang membuat volume untuk skenario penyimpanan beperforma tinggi, Anda harus memastikan untuk menggunakan volume SSD IOPS yang Tersedia (`io1` atau `io2`) dan memasangnya pada instans dengan bandwidth yang memadai bagi aplikasi Anda, seperti instans yang mengoptimalkan EBS. Saran yang sama berlaku untuk volume HDD Throughput Dioptimalkan (`st1`) dan Cold HDD (`sc1`).

> **Note**
>
> Jika Anda membuat volume untuk penggunaan dengan instans Windows, dan lebih besar dari 2048 GiB (atau merupakan volume yang lebih kecil dari 2048 GiB tetapi mungkin akan meningkat nanti), pastikan Anda mengonfigurasi volume untuk menggunakan tabel partisi GPT. Untuk informasi selengkapnya, lihat Dukungan Windows untuk hard disk yang lebih besar dari 2 TB.

Volume EBS yang kosong akan mencapai performa maksimalnya saat tersedia dan tidak memerlukan inisialisasi (sebelumnya dikenal sebagai pemanasan awal). Namun, blok penyimpanan pada volume yang dibuat dari snapshot harus inisialisasi (ditarik turun dari Amazon S3 dan ditulis ke volume) sebelum Anda dapat mengakses blok. Tindakan awal ini membutuhkan waktu dan dapat menyebabkan peningkatan yang signifikan dalam latensi operasi I/O, pada kali pertama setiap blok diakses. Performa volume dicapai setelah semua blok diunduh dan ditulis ke volume. Untuk sebagian besar aplikasi, amortisasi biaya ini selama masa pakai volume dapat diterima. Untuk menghindari performa awal ini terjadi dalam lingkungan produksi, Anda dapat memaksa inisialisasi seketika dari seluruh volume atau memungkinkan pemulihan snapshot cepat. Untuk informasi selengkapnya, lihat Inisialisasi volume Amazon EBS.

Metode pembuatan volume

- Buat dan pasangkan volume EBS saat Anda meluncurkan instans dengan menentukan pemetaan perangkat blok. Untuk informasi selengkapnya, lihat Meluncurkan instance menggunakan wizard instans peluncuran baru dan Memblokir pemetaan perangkat.

- Buat volume EBS kosong dan melampirkannya ke instans yang berjalan. Untuk informasi selengkapnya, lihat Buat volume kosong di bawah ini.
- Buat volume EBS dari snapshot yang dibuat sebelumnya dan lampirkan ke instans yang berjalan. Untuk informasi selengkapnya, lihat Membuat volume dari snapshot di bawah ini.

Topik

Buat volume kosong

Volume kosong menerima performa maksimum mereka saat volume tersedia dan tidak memerlukan inisialisasi.

Anda dapat membuat volume EBS kosong menggunakan salah satu metode berikut.

Console

Untuk membuat volume EBS kosong menggunakan konsol

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Volume.
3. Pilih Buat Volume.
4. Untuk Tipe volume, pilih tipe volume yang akan dibuat. Untuk informasi selengkapnya, lihat Tipe volume Amazon EBS.

   SSD Tujuan Umum `gp3` adalah pilihan default.
5. Untuk Ukuran, masukkan ukuran volume, dalam satuan GiB. Untuk informasi selengkapnya, lihat Batasan ukuran dan konfigurasi volume EBS.
6. (`io1`, `io2`, dan `gp3` saja) Untuk IOPS, masukkan jumlah maksimum operasi input/output per detik (IOPS) yang harus disediakan oleh volume.
7. (`gp3` saja) Untuk Throughput, masukkan throughput yang harus disediakan volume, dalam satuan MiB/dtk.
8. Untuk Zona Ketersediaan, pilih Zona Ketersediaan tempat pembuatan volume. Volume hanya dapat dilampirkan pada instans yang berada di Zona Ketersediaan yang sama.
9. Untuk ID Snapshot, pertahankan nilai default (Jangan buat volume dari snapshot).

10. (`io1` dan `io2` saja) Untuk mengaktifkan volume Multi-Lampiran Amazon EBS, pilih Aktifkan Multi-Lampiran. Untuk informasi selengkapnya, lihat Melampirkan volume ke beberapa instans dengan Multi-Lampiran Amazon EBS.
11. Atur status enkripsi untuk volume.

    Jika akun Anda diaktifkan untuk enkripsi secara default, enkripsi diaktifkan secara otomatis dan Anda tidak dapat menonaktifkannya. Anda dapat memilih kunci KMS untuk mengenkripsi volume.

    Jika akun Anda tidak diaktifkan untuk enkripsi secara default, enkripsi bersifat opsional. Untuk mengenkripsi volume, untuk Enkripsi, pilih Enkripsi volume ini lalu pilih kunci KMS yang akan digunakan untuk mengenkripsi volume.

    > **Note**
    >
    > Volume yang dienkripsi hanya dapat dilampirkan ke instans yang mendukung enkripsi Amazon EBS. Untuk informasi selengkapnya, lihat Enkripsi EBS Amazon.

12. (Opsional) Untuk menetapkan tag khusus ke volume, di bagian Tag, pilih Tambahkan tag, lalu masukkan kunci tag dan pasangan nilai.
13. Pilih Buat Volume.

    > **Note**
    >
    > Volume siap digunakan saat Status volume mencantumkan tersedia.

14. Untuk menggunakan volume, tempelkan ke instans. Untuk informasi selengkapnya, lihat Lampirkan volume Amazon EBS ke instans.

AWS CLI

Untuk membuat volume EBS kosong menggunakan AWS CLI

Gunakan perintah create-volume.

Volume siap digunakan saat `state` berstatus `available`.

Tools for Windows PowerShell

Untuk membuat volume EBS kosong menggunakan Alat untuk Windows PowerShell

Gunakan perintah New-EC2Volume.

Volume siap digunakan saat `state` berstatus `available`.

Membuat volume dari snapshot

Volume yang dibuat dari snapshot dimuat dengan lambat di latar belakang. Artinya, tidak perlu menunggu untuk semua data yang akan ditransfer dari Amazon S3 ke volume EBS Anda sebelum instans dapat mulai mengakses volume terpasang dan semua datanya. Jika instans Anda mengakses data yang belum dimuat, volume segera mengunduh data yang diminta dari Amazon S3 lalu melanjutkan memuat sisa data volume di latar belakang. Performa volume dicapai setelah semua blok diunduh dan ditulis ke volume. Untuk menghindari benturan performa awal di lingkungan produksi, lihat Inisialisasi volume Amazon EBS.

Volume EBS baru yang dibuat dari snapshot terenkripsi secara otomatis dienkripsi. Anda juga dapat mengenkripsi volume on-the-fly sambil memulihkannya dari snapshot yang tidak terenkripsi. Volume yang dienkripsi hanya dapat dilampirkan ke instans yang mendukung enkripsi Amazon EBS. Untuk informasi selengkapnya, lihat Tipe instans yang didukung.

Anda dapat membuat volume dari snapshot menggunakan salah satu metode berikut.

Console

Untuk membuat volume EBS dari snapshot menggunakan konsol

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Volume.
3. Pilih Buat Volume.
4. Untuk Tipe volume, pilih tipe volume yang akan dibuat. Untuk informasi selengkapnya, lihat Tipe volume Amazon EBS.

   SSD Tujuan Umum `gp3` adalah pilihan default.
5. Untuk Ukuran, masukkan ukuran volume, dalam satuan GiB. Untuk informasi selengkapnya, lihat Batasan ukuran dan konfigurasi volume EBS.
6. (`io1`, `io2`, dan `gp3` saja) Untuk IOPS, masukkan jumlah maksimum operasi input/output per detik (IOPS) yang harus disediakan oleh volume.
7. (`gp3` saja) Untuk Throughput, masukkan throughput yang harus disediakan volume, dalam satuan MiB/dtk.

8. Untuk Zona Ketersediaan, pilih Zona Ketersediaan tempat pembuatan volume. Volume hanya dapat dipasang pada instans yang berada di Zona Ketersediaan yang sama.
9. Untuk ID Snapshot, pilih snapshot yang akan digunakan untuk membuat volume.
10. Atur status enkripsi untuk volume.

    Jika snapshot yang dipilih dienkripsi, atau jika akun Anda diaktifkan untuk enkripsi secara default, enkripsi diaktifkan secara otomatis dan Anda tidak dapat menonaktifkannya. Anda dapat memilih kunci KMS untuk mengenkripsi volume.

    Jika snapshot yang dipilih tidak dienkripsi dan akun Anda tidak diaktifkan untuk enkripsi secara default, enkripsi bersifat opsional. Untuk mengenkripsi volume, untuk Enkripsi, pilih Enkripsi volume ini lalu pilih kunci KMS yang akan digunakan untuk mengenkripsi volume.

    > **Note**
    >
    > Volume yang dienkripsi hanya dapat dilampirkan ke instans yang mendukung enkripsi Amazon EBS. Untuk informasi selengkapnya, lihat Enkripsi EBS Amazon.

11. (Opsional) Untuk menetapkan tag khusus ke volume, di bagian Tag, pilih Tambahkan tag, lalu masukkan kunci tag dan pasangan nilai.
12. Pilih Buat Volume.

    > **Note**
    >
    > Volume siap digunakan saat Status volume mencantumkan tersedia.

13. Untuk menggunakan volume, tempelkan ke instans. Untuk informasi selengkapnya, lihat Lampirkan volume Amazon EBS ke instans.

AWS CLI

Untuk membuat volume EBS dari snapshot menggunakan AWS CLI

Gunakan perintah create-volume.

Volume siap digunakan saat `state` berstatus `available`.

Tools for Windows PowerShell

Untuk membuat volume EBS dari snapshot menggunakan Tools for Windows PowerShell

Gunakan perintah New-EC2Volume.

Volume siap digunakan saat `state` berstatus `available`.

Lampirkan volume Amazon EBS ke instans

Anda dapat melampirkan volume EBS yang tersedia pada satu atau beberapa instans yang berada dalam Zona Ketersediaan yang sama dengan volume tersebut.

Untuk informasi tentang menambahkan volume EBS ke instans Anda saat peluncuran, lihat pemetaan perangkat pemblokiran instans.

Pertimbangan

- Tentukan berapa banyak volume yang dapat Anda pasangkan ke instans Anda. Jumlah maksimum volume Amazon EBS yang dapat dilampirkan ke instans bergantung pada tipe instans dan ukuran instans. Untuk informasi selengkapnya, lihat Batas volume instans.
- Tentukan apakah Anda dapat memasang volume Anda ke beberapa instans dan mengaktifkan Multi-Lampiran. Untuk informasi selengkapnya, lihat Melampirkan volume ke beberapa instans dengan Multi-Lampiran Amazon EBS.
- Jika sebuah volume dienkripsi, Anda hanya dapat melampirkannya ke instans yang mendukung enkripsi Amazon EBS. Untuk informasi selengkapnya, lihat Tipe instans yang didukung.
- Jika volume memiliki kode AWS Marketplace produk:
  - Anda dapat memasang volume hanya ke instans yang dihentikan.
  - Anda harus berlangganan AWS Marketplace kode yang ada di volume.
  - Konfigurasi instans, seperti jenis dan sistem operasinya, harus mendukung AWS Marketplace kode tertentu. Misalnya, Anda tidak dapat mengambil volume dari instans Windows dan menempelkannya ke instans Linux.
  - AWS Marketplace kode produk disalin dari volume ke instance.

Anda dapat melampirkan volume ke instans dengan menggunakan metode berikut ini.

Console

Untuk memasang volume EBS ke suatu instans menggunakan konsol

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.

2. Di panel navigasi, pilih Volume.
3. Pilih volume yang akan dilampirkan dan pilih Tindakan, Lampirkan volume.

> **Note**
>
> Anda hanya dapat melampirkan volume yang ada dalam status `Available`.

4. Untuk Instans, masukkan ID instans atau pilih instans dari daftar opsi.

> **Note**
>
> - Volume harus dilampirkan ke sebuah instans dalam Zona Ketersediaan yang sama.
> - Jika volume dienkripsi, Anda hanya dapat dilampirkannya ke instans yang mendukung enkripsi Amazon EBS. Untuk informasi selengkapnya, lihat Enkripsi EBS Amazon.

5. Untuk nama Perangkat, lakukan salah satu hal berikut:

   - Untuk volume root, pilih nama perangkat yang diperlukan dari bagian Reserved for root volume dalam daftar. Biasanya `/dev/sda1` atau `/dev/xvda` untuk instance Linux tergantung pada AMI, atau `/dev/sda1` untuk instance Windows.
   - Untuk volume data, pilih nama perangkat yang tersedia dari bagian Direkomendasikan untuk volume data dalam daftar.
   - Untuk menggunakan nama perangkat kustom, pilih Tentukan nama perangkat kustom, lalu masukkan nama perangkat yang akan digunakan.

   Nama perangkat ini digunakan oleh Amazon EC2 Driver perangkat blok untuk instans mungkin menetapkan nama perangkat yang berbeda saat melakukan pemasangan volume. Untuk informasi selengkapnya, lihat nama perangkat di instance Linux atau nama perangkat di instance Windows.
6. Pilih Lampirkan volume.
7. Sambungkan ke instans dan pasang volume. Untuk informasi selengkapnya, lihat Buat volume Amazon EBS tersedia untuk digunakan.

AWS CLI

Untuk melampirkan volume EBS ke instance menggunakan AWS CLI

Gunakan perintah attach-volume.

Tools for Windows PowerShell

Untuk melampirkan volume EBS ke instance menggunakan Tools for Windows PowerShell

Gunakan perintah Add-EC2Volume.

> **Note**
>
> - Jika Anda mencoba melampirkan sejumlah volume yang melebihi batas volume tipe instans, permintaan gagal. Untuk informasi selengkapnya, lihat Batas volume instans.
> - Dalam beberapa situasi, Anda mungkin menemukan volume selain volume yang terpasang pada `/dev/xvda` atau `/dev/sda` telah menjadi volume root dari instans Anda. Ini dapat terjadi ketika Anda telah memasang volume root dari instans lain, atau volume yang dibuat dari tangkapan layar volume root, ke suatu instans dengan volume root yang ada. Untuk informasi selengkapnya, lihat Boot dari volume yang salah.

Melampirkan volume ke beberapa instans dengan Multi-Lampiran Amazon EBS

Dengan Multi-Lampiran Amazon EBS, Anda dapat memasang satu volume SSD IOPS yang Tersedia (`io1` atau `io2`) ke banyak instans yang berada dalam Zona Ketersediaan yang sama. Anda dapat memasang beberapa volume dengan Multi-Lampiran diaktifkan ke suatu instans atau serangkaian instans. Setiap instans di mana volume terpasang memiliki izin baca dan tulis penuh untuk volume yang dibagikan. Multi-Lampiran membuat Anda mudah untuk mendapatkan ketersediaan aplikasi yang lebih tinggi dalam aplikasi yang mengelola operasi kerja yang dilakukan secara bersamaan.

Daftar Isi

- Memantau volume dengan Multi-Lampiran diaktifkan
- Harga dan penagihan
- Reservasi NVMe

Pertimbangan dan batasan

- Volume yang diaktifkan Multi-Lampirkan dapat dilampirkan hingga 16 instans yang dibangun di Sistem Nitro yang berada di Availability Zone yang sama.
- Instans Linux mendukung Multi-Attach diaktifkan `io1` dan `io2` volume. Instans Windows hanya mendukung `io2` volume yang diaktifkan Multi-Attach.
- Jumlah maksimum volume Amazon EBS yang dapat dilampirkan ke instans bergantung pada tipe instans dan ukuran instans. Untuk informasi selengkapnya, lihat batas volume instance.
- Multi-Lampiran didukung secara eksklusif pada Volume SSD IOPS yang tersedia (`io1` dan `io2`).
- Multi-Lampiran untuk volume `io1` hanya tersedia di Wilayah berikut: AS Timur (Virginia Utara), AS Barat (Oregon), dan Asia Pasifik (Seoul).

  Multi-Lampiran untuk `io2` tersedia di semua Wilayah yang mendukung `io2`.

  > **Note**
  >
  > Untuk performa, konsistensi, dan daya tahan yang lebih baik dengan biaya lebih rendah, kami sarankan Anda menggunakan volume `io2`.

- Volume `io1` dengan Multi-Lampiran diaktifkan tidak didukung dengan instans yang dibangun di atas Nitro System yang mendukung protokol jaringan Scalable Reliable Datagram (SRD) saja. Untuk menggunakan Multi-Lampiran dengan tipe instans ini, Anda harus menggunakan volume `io2` Block Express.
- Sistem file standar, seperti XFS dan EXT4, tidak dirancang untuk diakses secara bersamaan oleh beberapa server, seperti instans EC2. Anda harus menggunakan sistem file klaster untuk memastikan ketahanan dan keandalan data untuk beban kerja produksi Anda.
- Volume `io2` dengan Multi-Lampiran diaktifkan mendukung pagar I/O. Protokol fencing I/O mengendalikan akses tulis dalam lingkungan penyimpanan bersama untuk menjaga konsistensi data. Aplikasi Anda harus memberikan urutan penulisan untuk instans terlampir untuk menjaga konsistensi data. Untuk informasi selengkapnya, lihat Reservasi NVMe.

  Volume `io1` dengan Multi-Lampiran diaktifkan tidak mendukung pagar I/O.

- Volume dengan Multi-Lampiran diaktifkan tidak dapat dibuat sebagai volume boot.
- Volume dengan Multi-Lampiran diaktifkan dapat dilampirkan ke satu pemetaan perangkat blok per instans.
- Multi-Lampirkan tidak dapat diaktifkan selama peluncuran instans menggunakan konsol RunInstances Amazon EC2 atau API.
- Volume dengan Multi-Lampiran diaktifkan yang memiliki masalah di lapisan infrastruktur Amazon EBS tidak tersedia untuk semua instans yang dipasang. Masalah di Amazon EC2 atau lapisan jaringan mungkin hanya berdampak pada beberapa instans yang terpasang.
- Tabel berikut menunjukkan dukungan modifikasi volume untuk volume `io1` dan `io2` dengan Multi-Lampiran diaktifkan setelah pembuatan.

	Volume **`io2`**	Volume **`io1`**
Mengubah tipe volume	✗	✗
Mengubah ukuran volume	✓	✗
Mengubah IOPS yang tersedia	✓	✗
Aktifkan Multi-Lampiran	✓ *	✗
Nonaktifkan Multi-Lampiran	✓ *	✗

* Anda tidak dapat mengaktifkan atau menonaktifkan Multi-Lampiran saat volume dilampirkan ke suatu instans.

Kinerja

Setiap instans yang dilampirkan mampu mendorong performa IOPS maksimum hingga performa maksimal yang tersedia dari volume. Namun, performa agregat dari semua instans yang terlampir tidak dapat melebihi performa maksimal yang tersedia dari volume. Jika permintaan instans yang

terpasang untuk IOPS lebih tinggi dari volume IOPS yang Tersedia, volumenya tidak akan melebihi performa yang disediakan.

Misalnya, Anda membuat volume dengan Multi-Lampiran diaktifkan `io2` dengan `80,000` IOPS yang Tersedia dan memasangnya ke instans `m7g.large` yang mendukung hingga `40,000` IOPS, dan `r7g.12xlarge` instans yang mendukung hingga `60,000` IOPS. Setiap instans dapat mendorong IOPS maksimum karena kurang dari volume IOPS yang tersedia sebesar `80,000`. Namun, jika kedua instans mendorong I/O ke volume secara bersamaan, IOPS gabungannya tidak dapat melebihi performa IOPS yang disediakan volume yaitu sebesar `80,000`.

Untuk mencapai performa yang konsisten, praktik terbaik adalah menyeimbangkan I/O yang didorong dari instans yang terlampir di seluruh sektor volume dengan Multi-Lampiran diaktifkan.

Bekerja dengan Multi-Lampiran

Volume dengan Multi-Lampiran diaktifkan dapat dikelola dengan cara yang sama dengan pengelolaan volume Amazon EBS lainnya. Namun, untuk menggunakan fungsi Multi-Lampiran, Anda harus mengaktifkannya untuk volume. Saat Anda membuat volume baru, Multi-Lampiran dinonaktifkan secara default.

Daftar Isi

Aktifkan Multi-Lampiran

Untuk mengaktifkan Multi-Lampiran selama pembuatan volume. Gunakan salah satu metode berikut.

Console

Untuk mengaktifkan Multi-Lampiran selama pembuatan volume

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Volume.
3. Pilih Buat Volume.

4. Untuk Tipe volume, pilih SSD IOPS yang Tersedia (**io1**) atau SSD IOPS yang Tersedia (**io2**).
5. Untuk Ukuran dan IOPS, pilih ukuran volume yang diperlukan dan jumlah IOPS untuk disediakan.
6. Untuk Zona Ketersediaan, pilih Zona Ketersediaan yang sama dengan lokasi instans.
7. Untuk Multi-Lampiran Amazon EBS, pilih Aktifkan Multi-Lampiran.
8. (Opsional) Untuk ID Snapshot, pilih snapshot tempat pembuatan volume.
9. Atur status enkripsi untuk volume.

   Jika snapshot yang dipilih dienkripsi, atau jika akun Anda diaktifkan untuk enkripsi secara default, enkripsi diaktifkan secara otomatis dan Anda tidak dapat menonaktifkannya. Anda dapat memilih kunci KMS untuk mengenkripsi volume.

   Jika snapshot yang dipilih tidak dienkripsi dan akun Anda tidak diaktifkan untuk enkripsi secara default, enkripsi bersifat opsional. Untuk mengenkripsi volume, untuk Enkripsi, pilih Enkripsi volume ini lalu pilih kunci KMS yang akan digunakan untuk mengenkripsi volume.

   > **Note**
   >
   > Volume yang dienkripsi hanya dapat dilampirkan ke instans yang mendukung enkripsi Amazon EBS. Untuk informasi selengkapnya, lihat Enkripsi EBS Amazon.

10. (Opsional) Untuk menetapkan tag khusus ke volume, di bagian Tag, pilih Tambahkan tag, lalu masukkan kunci tag dan pasangan nilai.
11. Pilih Buat Volume.

Command line

Untuk mengaktifkan Multi-Lampiran selama pembuatan volume

Gunakan perintah create-volume dan tentukan parameter `--multi-attach-enabled`.

```
$ C:\> aws ec2 create-volume --volume-type io2 --multi-attach-enabled --size 100 --
iops 2000 --region us-west-2 --availability-zone us-west-2b
```

Anda juga dapat mengaktifkan Multi-Lampiran untuk volume `io2` setelah pembuatan, tetapi hanya jika volume tersebut tidak terhubung ke instans apa pun.

> **Note**
>
> Anda tidak dapat mengaktifkan Multi-Lampiran untuk volume `io1` setelah pembuatan.

Gunakan salah satu metode berikut untuk mengaktifkan Multi-Lampiran untuk volume `io2` setelah pembuatan.

Console

Untuk mengaktifkan Multi-Lampiran setelah pembuatan

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Volume.
3. Pilih volume dan pilih Tindakan, Ubah Volume.
4. Untuk Multi-Lampiran Amazon EBS, pilih Aktifkan Multi-Lampiran.
5. Pilih Ubah.

Command line

Untuk mengaktifkan Multi-Lampiran setelah pembuatan

Gunakan perintah modify-volume dan tentukan parameter `--multi-attach-enabled`.

```
$ C:\> aws ec2 modify-volume --volume-id vol-1234567890abcdef0 --multi-attach-
enabled
```

Nonaktifkan Multi-Lampiran

Anda dapat menonaktifkan Multi-Lampiran untuk volume `io2` hanya jika dilampirkan ke tidak lebih dari satu instans.

> **Note**
>
> Anda tidak dapat menonaktifkan Multi-Lampiran untuk volume `io1` setelah pembuatan.

Gunakan salah satu metode berikut untuk menonaktifkan Multi-Lampiran untuk sebuah volume `io2`.

Console

Untuk menonaktifkan Multi-Lampiran setelah pembuatan

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Volume.
3. Pilih volume dan pilih Tindakan, Ubah Volume.
4. Untuk Multi-Lampiran Amazon EBS, hapus Aktifkan Multi-Lampiran.
5. Pilih Ubah.

Command line

Untuk menonaktifkan Multi-Lampiran setelah pembuatan

Gunakan perintah modify-volume dan tentukan parameter `-no-multi-attach-enabled`.

```
$ C:\> aws ec2 modify-volume --volume-id vol-1234567890abcdef0 --no-multi-attach-
enabled
```

Lampirkan volume ke instans

Anda memasang volume dengan Multi-Lampiran diaktifkan pada sebuah instans dengan cara yang sama seperti Anda melampirkan volume EBS lainnya. Untuk informasi selengkapnya, lihat Lampirkan volume Amazon EBS ke instans.

Hapus saat penghentian

Volume dengan Multi-Lampiran diaktifkan dihapus pada saat pengakhiran instans jika instans terakhir yang dilampirkan diakhiri dan jika instans tersebut dikonfigurasi untuk menghapus volume pada saat pengakhiran. Jika volume terlampir ke banyak instans yang memiliki pengaturan pengakhiran saat pengakhiran yang berbeda dalam pemetaan perangkat blok volumenya, pengaturan pemetaan perangkat blok instans terakhir yang terlampir menentukan penghapusan pada perilaku pengakhiran.

Untuk memastikan penghapusan yang dapat diprediksi pada perilaku pengakhiran, aktifkan atau nonaktifkan penghapusan pada saat pengakhiran untuk semua instans tempat volume terpasang.

Secara default, ketika volume dilampirkan ke suatu instans, penghapusan pada pengaturan pengakhiran untuk pemetaan perangkat blok diatur ke palsu. Jika Anda ingin mengaktifkan

penghapusan saat pengakhiran untuk volume dengan Multi-Lampiran diaktifkan, ubah pemetaan perangkat blok.

Jika Anda ingin volume dihapus ketika instans yang terlampir diakhiri, aktifkan penghapusan pada saat pengakhiran pemetaan perangkat blok untuk semua instans yang terlampir. Jika Anda ingin mempertahankan volume setelah instans yang terpasang diakhiri, nonaktifkan penghapusan saat pengakhiran di pemetaan perangkat blok untuk semua instans yang terlampir. Untuk informasi selengkapnya, lihat Mempertahankan data saat instance dihentikan.

Anda dapat memodifikasi penghapusan suatu instans pada pengaturan pengakhiran saat peluncuran atau setelah diluncurkan. Jika Anda mengaktifkan atau menonaktifkan penghapusan pada pengakhiran selama peluncuran instans, pengaturan hanya berlaku untuk volume yang dilampirkan saat peluncuran. Jika Anda memasang volume ke suatu instans setelah peluncuran, Anda harus secara eksplisit mengatur penghapusan pada perilaku pengakhiran untuk volume tersebut.

Anda dapat mengubah pengaturan penghapusan saat pengakhiran suatu instans menggunakan alat baris perintah saja.

Untuk mengubah pengaturan penghapusan saat pengakhiran untuk instans yang sudah ada

Gunakan perintah modify-instance-attribute dan tentukan atribut `DeleteOnTermination` dalam `--block-device-mappings option`.

```
aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --block-device-
mappings file://mapping.json
```

Tentukan hal berikut dalam `mapping.json`.

```
[
  {
    "DeviceName": "/dev/sdf",
    "Ebs": {
      "DeleteOnTermination": true|false
    }
  }
]
```

Memantau volume dengan Multi-Lampiran diaktifkan

Anda dapat memantau volume yang diaktifkan Multi-Lampirkan menggunakan CloudWatch Metrik untuk volume Amazon EBS. Untuk informasi selengkapnya, lihat CloudWatch Metrik Amazon untuk Amazon EBS.

Data digabungkan di semua instans yang terlampir. Anda tidak dapat memantau metrik untuk setiap instans yang terlampir.

Harga dan penagihan

Tidak ada biaya tambahan untuk menggunakan Multi-Lampiran Amazon EBS. Anda dikenai biaya dengan tarif standar yang berlaku untuk volume SSD IOPS yang Tersedia (`io1` dan `io2`). Untuk informasi selengkapnya, lihat harga Amazon EBS.

Reservasi NVMe

Volume `io2` dengan Multi-Lampiran diaktifkan mendukung reservasi NVMe, yang merupakan set protokol pagar penyimpanan standar industri. Protokol ini memungkinkan Anda membuat dan mengelola reservasi yang mengontrol dan mengoordinasikan akses dari beberapa instans ke volume bersama. Reservasi digunakan oleh aplikasi penyimpanan bersama untuk memastikan konsistensi data.

Topik

- Persyaratan
- Mengaktifkan dukungan untuk reservasi NVMe
- Perintah Reservasi NVMe yang didukung
- Harga

Persyaratan

Reservasi NVMe didukung hanya dengan volume `io2` dengan Multi-Lampiran diaktifkan. Volume dengan Multi-Lampiran diaktifkan hanya dapat dilampirkan ke instans yang dibangun di Nitro system.

Reservasi NVMe didukung dengan sistem operasi berikut ini:

- SUSE Linux Enterprise 12 SP3 dan lebih baru
- RHEL 8.3 dan yang lebih baru
- Amazon Linux 2 dan yang lebih baru

- Windows Server 2016 dan setelahnya

> **Note**
>
> Untuk AMI Windows Server yang didukung tertanggal 2023.09.13 dan yang lebih baru, driver NVMe yang diperlukan disertakan. Untuk AMI sebelumnya, Anda harus memperbarui ke driver NVMe versi 1.5.0 atau yang lebih baru. Untuk informasi selengkapnya, lihat driver AWS NVMe untuk instance Windows.

Jika Anda menggunakan EC2Launch v2 untuk menginisialisasi disk, Anda harus meningkatkan ke versi 2.0.1521 atau yang lebih baru. Untuk informasi selengkapnya, lihat Mengonfigurasi instance Windows menggunakan EC2launch v2.

Mengaktifkan dukungan untuk reservasi NVMe

Dukungan untuk reservasi NVMe diaktifkan secara default untuk semua volume `io2` dengan Multi-Lampiran diaktifkan yang dibuat setelah 18 September 2023.

Untuk mengaktifkan dukungan reservasi NVMe untuk volume `io2` yang ada yang dibuat sebelum 18 September 2023, Anda harus melepaskan lampiran semua instans dari volume, kemudian memasang kembali instans yang diperlukan. Semua lampiran yang dibuat setelah melepaskan lampiran semua instans akan mengaktifkan reservasi NVMe.

Perintah Reservasi NVMe yang didukung

Amazon EBS mendukung perintah Reservasi NVMe berikut:

Registrasi Reservasi

Mendaftarkan, membatalkan pendaftaran, atau mengganti kunci reservasi. Kunci registrasi digunakan untuk mengidentifikasi dan mengautentikasi sebuah instans. Mendaftarkan kunci reservasi dengan volume menciptakan kairan antara instans dan volume. Anda harus mendaftarkan instans dengan volume sebelum instans itu dapat memperoleh reservasi.

Pemerolehan Reservasi

Memperoleh reservasi pada volume, mendahului reservasi yang disimpan di namespace, dan membatalkan reservasi yang disimpan pada volume. Jenis reservasi berikut dapat diperoleh:

- Tulis Reservasi Eksklusif

- Reservasi Akses Eksklusif
- Tulis Eksklusif - Hanya Reservasi Pendaftar
- Akses Eksklusif - Reservasi Khusus Pendaftar
- Tulis Eksklusif - Reservasi Semua Pendaftar
- Akses Eksklusif - Reservasi Semua Pendaftar

Rilis Reservasi

Merilis atau menghapus reservasi yang disimpan pada volume.

Laporan Reservasi

Menjelaskan status pendaftaran dan reservasi volume.

Harga

Tidak ada biaya tambahan untuk mengaktifkan dan menggunakan Multi-Lampiran.

Buat volume Amazon EBS tersedia untuk digunakan

Setelah Anda melampirkan volume Amazon EBS ke instans, volume tersebut akan ditampilkan sebagai perangkat pemblokiran. Anda dapat memformat volume dengan sembarang sistem file lalu memasangnya. Setelah Anda menyediakan volume EBS untuk digunakan, Anda dapat mengaksesnya dengan cara yang sama seperti Anda mengakses volume lainnya. Setiap data yang ditulis pada sistem file ini ditulis ke volume EBS dan terlihat untuk aplikasi yang menggunakan perangkat tersebut.

Anda dapat mengambil foto volume EBS untuk tujuan pencadangan atau menggunakannya sebagai dasar saat Anda membuat volume lain. Untuk informasi selengkapnya, lihat Snapshot Amazon EBS.

Jika volume EBS yang Anda persiapkan untuk digunakan lebih besar dari 2 TiB, Anda harus menggunakan skema partisi GPT untuk mengakses seluruh volume. Untuk informasi selengkapnya, lihat Batasan ukuran dan konfigurasi volume EBS.

Instans Linux

Format dan pasang volume yang terpasang

Misalkan, Anda memiliki instans EC2 dengan volume EBS untuk perangkat root, `/dev/xvda`, dan bahwa Anda baru saja melampirkan volume EBS kosong pada instans menggunakan `/dev/sdf`. Gunakan prosedur berikut untuk membuat volume baru terpasang tersedia untuk digunakan.

Untuk memformat dan memasang volume EBS di Linux

1. Connect ke instans Anda dengan menggunakan SSH. Untuk informasi selengkapnya, lihat Connect ke instans Linux Anda.
2. Perangkat dapat dilampirkan ke instans dengan nama perangkat yang berbeda dengan yang Anda tentukan dalam pemetaan perangkat blok. Untuk informasi selengkapnya, lihat nama perangkat di instance Linux. Gunakan perintah lsblk untuk melihat perangkat disk yang tersedia dan titik pemasangannya (jika ada) untuk membantu Anda menentukan nama perangkat yang tepat untuk digunakan. Output dari lsblk menghapus prefiks `/dev/` dari jalur perangkat lengkap.

   Berikut ini adalah contoh output untuk instance yang dibangun di atas Sistem Nitro, yang mengekspos volume EBS sebagai perangkat blok NVMe. Perangkat root adalah `/dev/nvme0n1`, yang memiliki dua partisi bernama `nvme0n1p1` dan `nvme0n1p128`. Volume yang terlampir adalah `/dev/nvme1n1`, yang tidak memiliki partisi dan belum dipasang.

   ```
   [ec2-user ~]$ lsblk
   NAME          MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
   nvme1n1       259:0    0  10G  0 disk
   nvme0n1       259:1    0   8G  0 disk
   -nvme0n1p1    259:2    0   8G  0 part /
   -nvme0n1p128  259:3    0   1M  0 part
   ```

   Berikut ini instans output untuk instans T2. Perangkat root adalah `/dev/xvda`, yang memiliki satu partisi bernama `xvda1`. Volume yang terlampir adalah `/dev/xvdf`, yang tidak memiliki partisi dan belum dipasang.

   ```
   [ec2-user ~]$ lsblk
   NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
   xvda    202:0    0    8G  0 disk
   -xvda1  202:1    0    8G  0 part /
   xvdf    202:80   0   10G  0 disk
   ```
3. Tentukan apakah ada sistem file pada volume. Volume baru adalah perangkat blok mentah, dan Anda harus membuat sistem file di dalamnya sebelum Anda dapat memasang dan menggunakannya. Volume yang dibuat dari snapshot mungkin sudah memiliki sistem file; jika Anda membuat sistem file baru di atas sistem file yang sudah ada, operasi akan menimpa data Anda.

Gunakan salah satu atau kedua metode berikut untuk menentukan apakah ada sistem file pada volume:

- Gunakan perintah file -s untuk mendapatkan informasi tentang perangkat spesifik, seperti tipe sistem file-nya. Jika output menunjukkan hanya `data`, seperti pada contoh output berikut, tidak ada sistem file di perangkat

```
[ec2-user ~]$ sudo file -s /dev/xvdf
/dev/xvdf: data
```

Jika perangkat memiliki sistem file, perintah akan menampilkan informasi tentang jenis sistem file. Misalnya, output berikut menunjukkan perangkat root dengan sistem file XFS.

```
[ec2-user ~]$ sudo file -s /dev/xvda1
/dev/xvda1: SGI XFS filesystem data (blksz 4096, inosz 512, v2 dirs)
```

- Gunakan perintah lsblk -f untuk mendapatkan informasi tentang semua perangkat yang terlampir pada instans.

```
[ec2-user ~]$ sudo lsblk -f
```

Misalnya, output berikut menunjukkan bahwa ada tiga perangkat yang dilampirkan ke instans—`nvme1n1`, `nvme0n1`, dan `nvme2n1`. Kolom pertama mencantumkan perangkat dan partisi mereka. Kolom FSTYPE menunjukkan jenis sistem file untuk setiap perangkat. Jika kolom kosong untuk perangkat tertentu, itu berarti perangkat tidak memiliki sistem file. Dalam hal ini, perangkat `nvme1n1` dan partisi `nvme0n1p1` pada perangkat `nvme0n1` keduanya diformat menggunakan sistem file XFS, sedangkan perangkat `nvme2n1` dan partisi `nvme0n1p128` pada perangkat `nvme0n1` tidak memiliki sistem file.

```
NAME  FSTYPE LABEL UUID      MOUNTPOINT
nvme1n1         xfs  7f939f28-6dcc-4315-8c42-6806080b94dd
nvme0n1
##nvme0n1p1 xfs     / 90e29211-2de8-4967-b0fb-16f51a6e464c         /
##nvme0n1p128
nvme2n1
```

Jika output dari perintah ini menunjukkan bahwa tidak ada sistem file pada perangkat, Anda harus membuatnya.

4. (Bersyarat) Jika Anda menemukan bahwa ada sistem file pada perangkat di langkah sebelumnya, lewati langkah ini. Jika Anda memiliki volume kosong, gunakan perintah mkfs -t untuk membuat sistem file pada volume.

   > **Warning**
   >
   > Jangan gunakan perintah ini jika Anda memasang volume yang sudah memiliki data di dalamnya (misalnya, volume yang dibuat dari snapshot). Jika tidak, Anda akan memformat volume dan menghapus data yang ada.

   ```
   [ec2-user ~]$ sudo mkfs -t xfs /dev/xvdf
   ```

   Jika Anda menerima pesan kesalahan bahwa `mkfs.xfs` tidak ditemukan, gunakan perintah berikut untuk menginstal alat XFS, lalu ulangi perintah sebelumnya:

   ```
   [ec2-user ~]$ sudo yum install xfsprogs
   ```

5. Gunakan perintah mkdir untuk membuat direktori titik pasang untuk volume. Titik pasang adalah tempat volume berada di struktur sistem file dan tempat Anda membaca serta menulis file setelah Anda memasang volume. Contoh berikut membuat direktori yang bernama `/data`.

   ```
   [ec2-user ~]$ sudo mkdir /data
   ```

6. Pasang volume atau partisi pada direktori titik pemasangan yang Anda buat pada langkah sebelumnya.

   Jika volume tidak memiliki partisi, gunakan perintah berikut dan tentukan nama perangkat untuk memasang seluruh volume.

   ```
   [ec2-user ~]$ sudo mount /dev/xvdf /data
   ```

   Jika volume memiliki partisi, gunakan perintah berikut dan tentukan nama partisi untuk memasang partisi.

```
[ec2-user ~]$ sudo mount /dev/xvdf1 /data
```

7. Tinjaulah izin file untuk pemasangan volume baru Anda untuk memastikan bahwa pengguna dan aplikasi Anda dapat menulis ke volume. Untuk informasi selengkapnya tentang izin file, lihat Keamanan file di Proyek Dokumentasi Linux.
8. Titik pemasangan tidak dipertahankan secara otomatis setelah melakukan boot ulang instans Anda. Untuk secara otomatis memasang volume EBS setelah boot ulang, lihat Otomatis memasang volume yang terlampir setelah boot ulang.

Otomatis memasang volume yang terlampir setelah boot ulang

Untuk memasang volume EBS yang terlampir pada setiap boot ulang sistem, tambahkan entri untuk perangkat ke file `/etc/fstab` Anda.

Anda dapat menggunakan nama perangkat, seperti `/dev/xvdf`, di `/etc/fstab`, tetapi sebaiknya gunakan pengidentifikasi yang unik universal (UUID) 128-bit. Nama perangkat dapat berubah, tetapi UUID tetap ada selama masa paruh. Dengan menggunakan UUID, Anda mengurangi kemungkinan sistem menjadi tidak dapat diaktifkan setelah konfigurasi ulang perangkat keras. Untuk informasi selengkapnya, lihat Identifikasi perangkat EBS.

Untuk otomatis memasang volume yang terlampir setelah boot ulang

1. (Opsional) Buat cadangan dari file `/etc/fstab` Anda yang dapat digunakan jika Anda secara tidak sengaja menghancurkan atau menghapus file ini saat mengedit.

   ```
   [ec2-user ~]$ sudo cp /etc/fstab /etc/fstab.orig
   ```

2. Gunakan perintah blkid untuk menemukan UUID perangkat. Buat catatan UUID perangkat yang ingin Anda pasang setelah boot ulang. Anda akan membutuhkannya dalam langkah berikut.

   Sebagai contoh, perintah berikut menunjukkan bahwa ada dua perangkat yang dipasang ke instans, dan menunjukkan UUID untuk kedua perangkat.

   ```
   [ec2-user ~]$ sudo blkid
   /dev/xvda1: LABEL="/" UUID="ca774df7-756d-4261-a3f1-76038323e572" TYPE="xfs"
    PARTLABEL="Linux" PARTUUID="02dcd367-e87c-4f2e-9a72-a3cf8f299c10"
   /dev/xvdf: UUID="aebf131c-6957-451e-8d34-ec978d9581ae" TYPE="xfs"
   ```

Untuk Ubuntu 18.04, gunakan perintah lsblk.

```
[ec2-user ~]$ sudo lsblk -o +UUID
```

3. Buka file `/etc/fstab` menggunakan editor teks, seperti nano atau vim.

```
[ec2-user ~]$ sudo vim /etc/fstab
```

4. Tambahkan entri berikut ke `/etc/fstab` untuk memasang perangkat di titik pemasangan yang ditentukan. Kolom tersebut adalah nilai UUID yang dikembalikan oleh blkid (atau lsblk untuk Ubuntu 18.04), titik pemasangan, sistem file, dan opsi pemasangan sistem file yang direkomendasikan. Untuk informasi lebih lanjut tentang bidang yang diperlukan, jalankan `man fstab` Untuk membuka fstab manual.

   Pada contoh berikut, kami memasang perangkat dengan UUID `aebf131c-6957-451e-8d34-ec978d9581ae` ke titik pemasangan `/data` dan kami menggunakan sistem file `xfs`. Kami juga menggunakan `defaults` dan `nofail` Bendera. Kami tentukan `0` untuk mencegah agar sistem file tidak dibuang, dan kami tentukan 2 untuk menunjukkan bahwa itu adalah perangkat non-root.

```
UUID=aebf131c-6957-451e-8d34-ec978d9581ae  /data  xfs  defaults,nofail  0  2
```

> **Note**
>
> Jika Anda pernah melakukan boot pada instans Anda tanpa melampirkan volume ini (misalnya, setelah memindahkan volume ke instans lain), opsi pemasangan `nofail` memungkinkan instans di-boot meskipun terdapat kesalahan saat memasang volume. Derivatif Debian, termasuk versi Ubuntu yang lebih awal dari 16.04, juga harus menambahkan opsi pemasangan `nobootwait`.

5. Untuk memverifikasi bahwa entri Anda bekerja, jalankan perintah berikut untuk melepas perangkat, kemudian memasang semua sistem file di `/etc/fstab`. Jika tidak ada kesalahan, file `/etc/fstab` akan baik-baik saja dan sistem file Anda akan memasangkannya secara otomatis setelah di-boot ulang.

```
[ec2-user ~]$ sudo umount /data
[ec2-user ~]$ sudo mount -a
```

Jika Anda menerima pesan kesalahan, atasi kesalahan dalam file.

> **Warning**
>
> Kesalahan dalam file `/etc/fstab` dapat membuat sistem tidak dapat di-boot. Jangan mematikan sistem yang memiliki kesalahan di `/etc/fstab` file Anda.

Jika Anda tidak yakin bagaimana cara memperbaiki kesalahan di `/etc/fstab` dan Anda telah membuat file cadangan di langkah pertama prosedur ini, Anda dapat memulihkan dari file cadangan menggunakan perintah berikut.

```
[ec2-user ~]$ sudo mv /etc/fstab.orig /etc/fstab
```

Instans Windows

Gunakan salah satu metode berikut untuk membuat volume tersedia pada instance Windows.

PowerShell

Untuk membuat semua volume EBS dengan partisi mentah tersedia untuk digunakan dengan Windows PowerShell

1. Masuk ke instans Windows menggunakan Remote Desktop. Untuk informasi selengkapnya, lihat Connect ke instans Windows Anda.
2. Pada taskbar, buka menu Start, dan pilih Windows. PowerShell
3. Gunakan serangkaian PowerShell perintah Windows yang disediakan dalam PowerShell prompt yang dibuka. Skrip tersebut melakukan tindakan-tindakan berikut ini secara default:
    1. Menghentikan layanan ShellHWDetection.
    2. Melakukan enumerasi disk yang gaya partisinya mentah.
    3. Membuat partisi baru yang mencakup ukuran maksimum yang akan didukung oleh disk dan jenis partisi.
    4. Menetapkan huruf drive yang tersedia.
    5. Memformat sistem file sebagai NTFS dengan label sistem file yang ditentukan.
    6. Memulai kembali layanan ShellHWDetection.

```
Stop-Service -Name ShellHWDetection
Get-Disk | Where PartitionStyle -eq 'raw' | Initialize-Disk -PartitionStyle MBR
 -PassThru | New-Partition -AssignDriveLetter -UseMaximumSize | Format-Volume -
FileSystem NTFS -NewFileSystemLabel "Volume Label" -Confirm:$false
Start-Service -Name ShellHWDetection
```

DiskPart command line tool

Untuk membuat volume EBS tersedia untuk digunakan dengan alat baris DiskPart perintah

1. Masuk ke instans Windows menggunakan Remote Desktop. Untuk informasi selengkapnya, lihat Connect ke instans Windows Anda.
2. Tentukan nomor disk yang ingin Anda sediakan:
    1. Buka menu Start, dan pilih Windows PowerShell.
    2. Gunakan Cmdlet `Get-Disk` untuk mengambil daftar disk yang tersedia.
    3. Dalam output perintah, perhatikan Nomor yang sesuai dengan disk yang Anda sediakan.
3. Buat file skrip untuk menjalankan DiskPart perintah:
    1. Buka menu Start, dan pilih File Explorer.
    2. Arahkan ke direktori, seperti C:\, untuk menyimpan file skrip.
    3. Pilih atau klik kanan ruang kosong di dalam folder untuk membuka kotak dialog, posisikan kursor di atas Baru untuk mengakses menu konteks, lalu pilih Dokumen Teks.
    4. Beri nama file teks `diskpart.txt`.
4. Tambahkan perintah berikut ke file skrip. Anda mungkin perlu memodifikasi nomor disk, jenis partisi, label volume, dan huruf drive. Skrip tersebut melakukan tindakan-tindakan berikut ini secara default:
    1. Memilih disk 1 untuk modifikasi.
    2. Mengonfigurasi volume untuk menggunakan struktur partisi master boot record (MBR).
    3. Memformat volume sebagai volume NTFS.
    4. Mengatur label volume.
    5. Menetapkan volume huruf drive.

> **Warning**
>
> Jika Anda memasang volume yang sudah memiliki data, jangan memformat ulang volume atau Anda akan menghapus data yang ada.

```
select disk 1
attributes disk clear readonly
online disk noerr
convert mbr
create partition primary
format quick fs=ntfs label="volume_label"
assign letter="drive_letter"
```

Untuk informasi selengkapnya, lihat DiskPart Sintaks dan Parameter.

5. Buka prompt perintah, arahkan ke folder tempat skrip berada, dan jalankan perintah berikut agar volume tersedia untuk digunakan pada disk yang ditentukan:

```
C:\> diskpart /s diskpart.txt
```

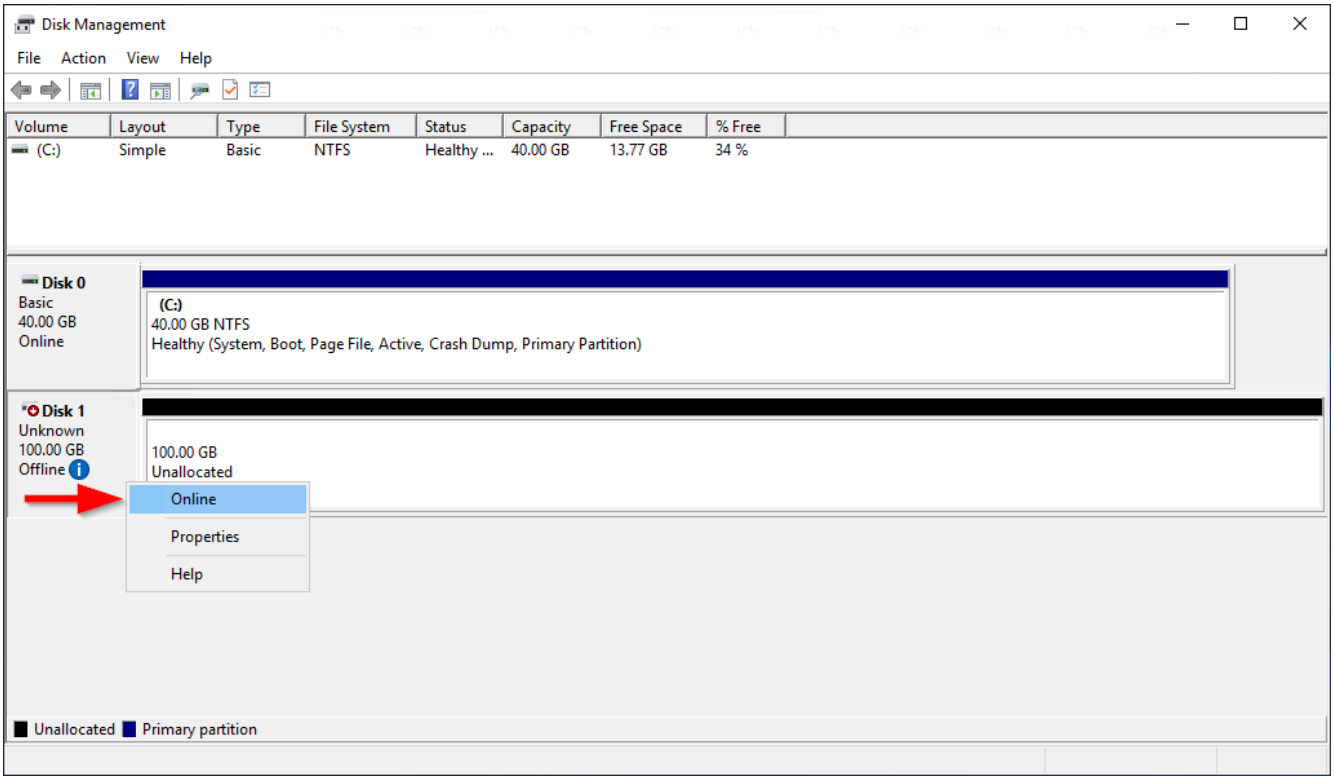
Disk Management utility

Untuk menjadikan volume EBS tersedia untuk digunakan dengan alat baris perintah DiskPart

1. Masuk ke instans Windows menggunakan Remote Desktop. Untuk informasi selengkapnya, lihat Connect ke instans Windows Anda.
2. Mulai utilitas Manajemen Disk. Pada bilah tugas, buka menu konteks (klik kanan) untuk logo Windows dan pilih Manajemen Disk.

> **Note**
>
> Di Windows Server 2008, pilih Mulai, Alat Administratif, Manajemen Komputer, Manajemen Disk.

3. Buat volume menjadi online. Di panel bawah, buka menu konteks (klik kanan) untuk panel kiri untuk disk untuk volume EBS. Pilih Online.

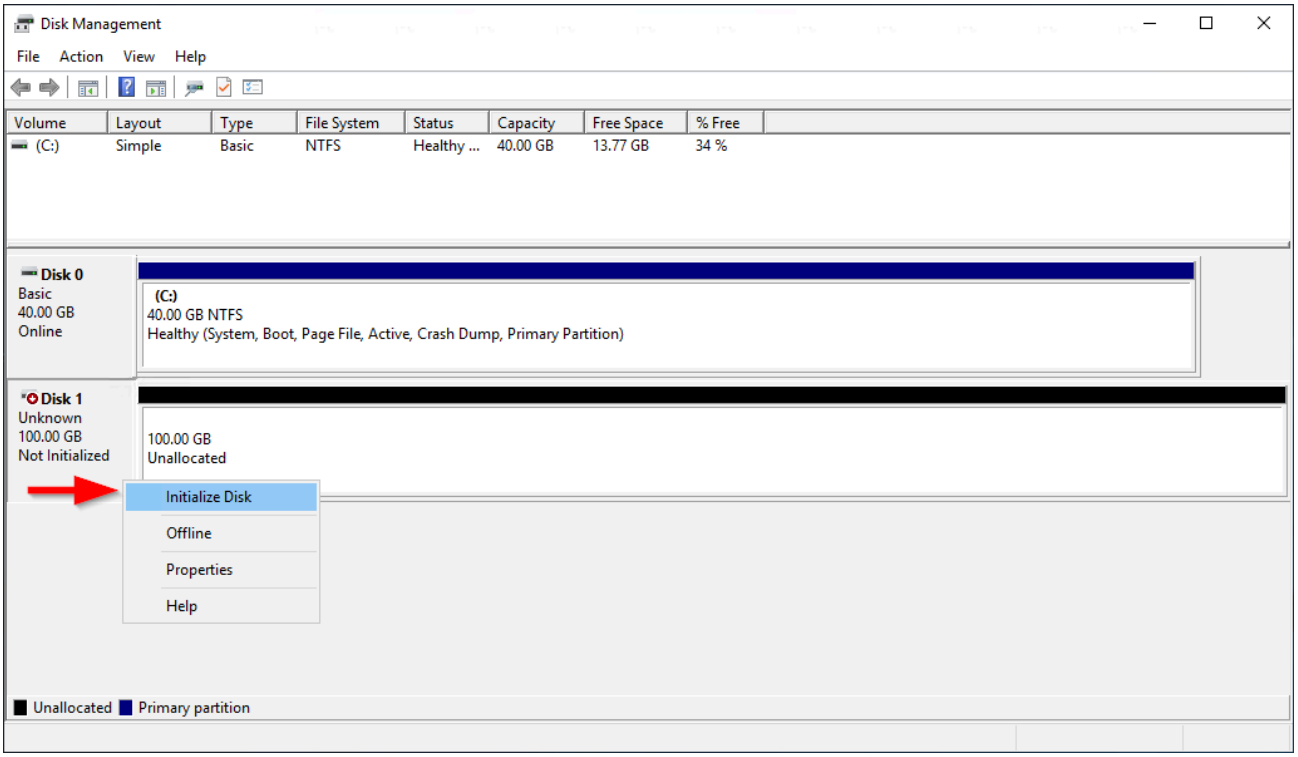


4. (Syarat) Jika disk tidak diinisialisasi, Anda harus menginisialisasinya sebelum Anda dapat menggunakannya. Jika disk sudah diinisialisasi, lewati langkah ini.

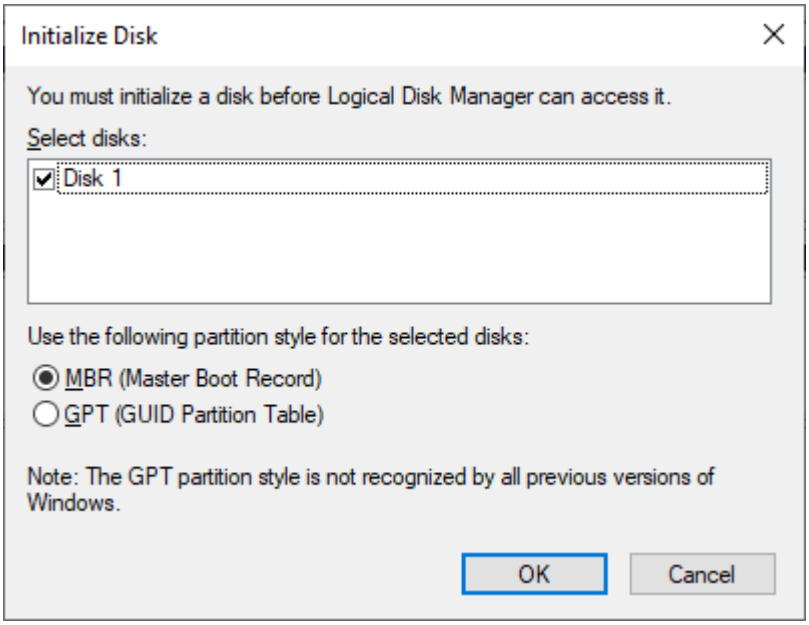
> **Warning**
>
> Jika Anda memasang volume yang sudah memiliki data di dalamnya (misalnya, set data publik, atau volume yang Anda buat dari snapshot), jangan memformat ulang volume atau data yang ada akan terhapus.

Jika disk tidak diinisialisasi, lakukan inisialisasi sebagai berikut:

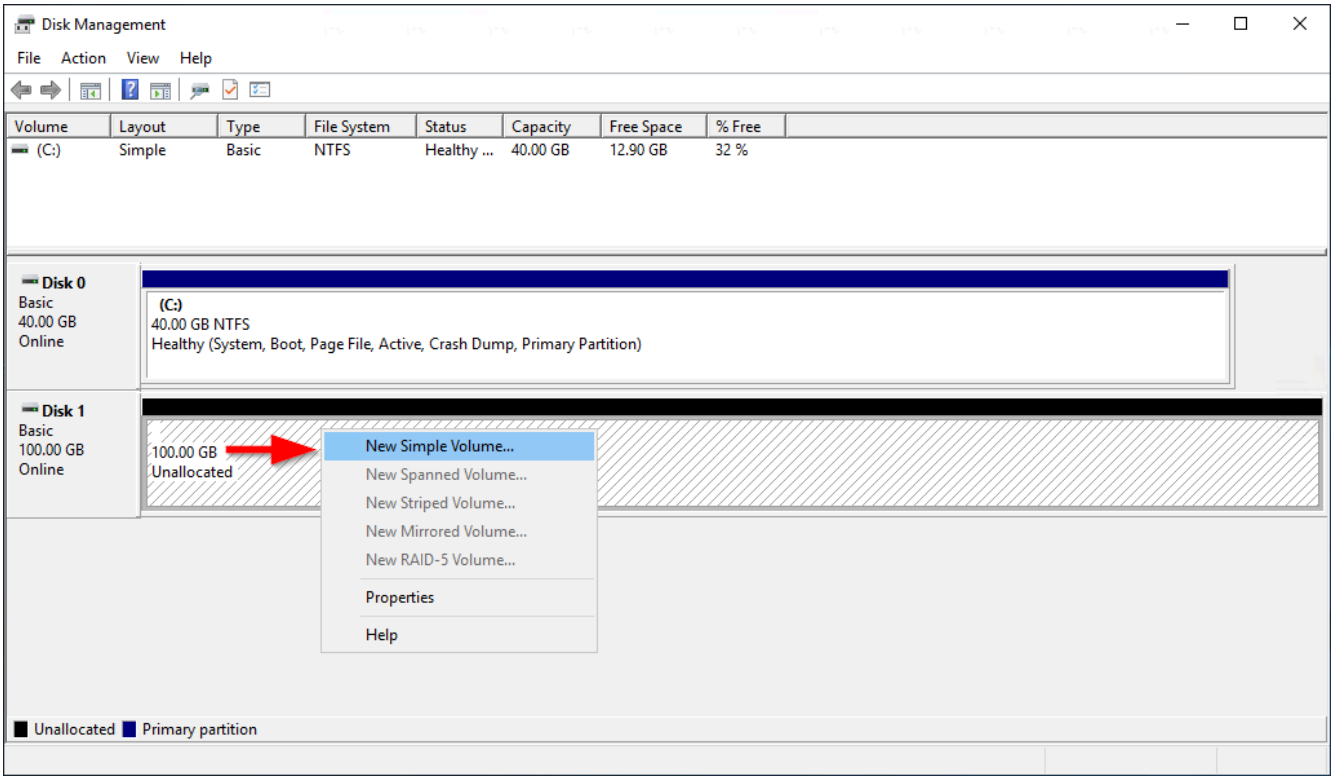
1. Buka menu konteks (klik kanan) untuk panel kiri untuk disk dan pilih Inisialisasi Disk.



2. Di kotak dialog Inisialisasi Disk, pilih gaya partisi dan pilih OKE.

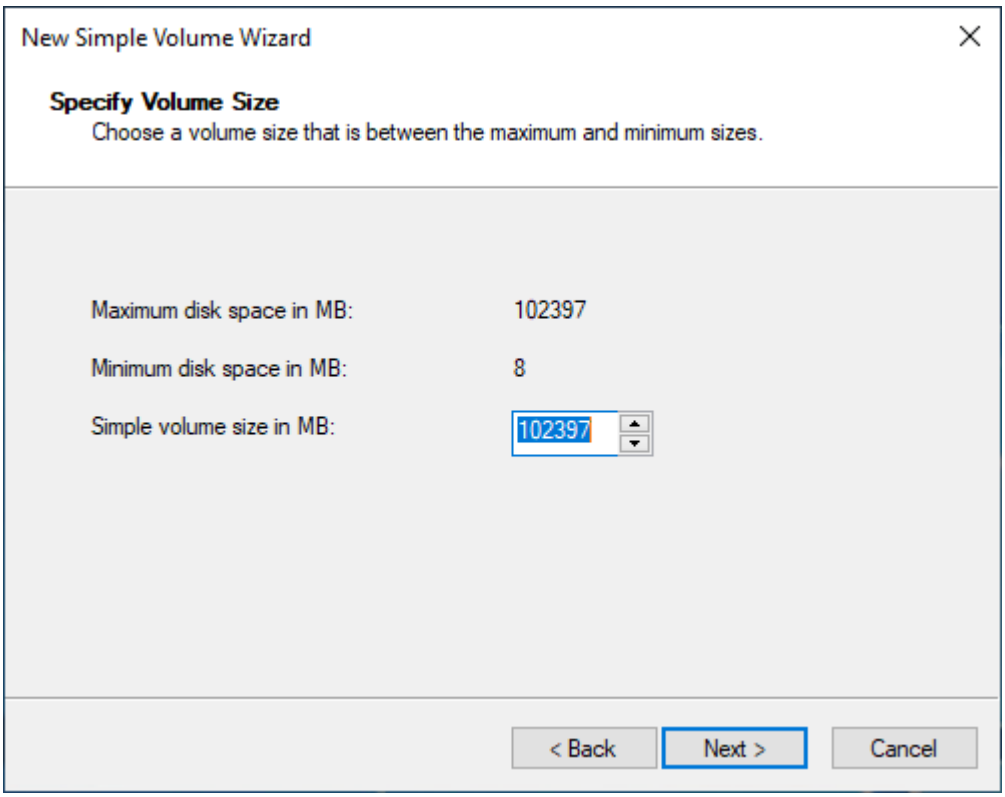


5. Buka menu konteks (klik kanan) untuk panel kanan untuk disk dan pilih Volume Sederhana Baru.

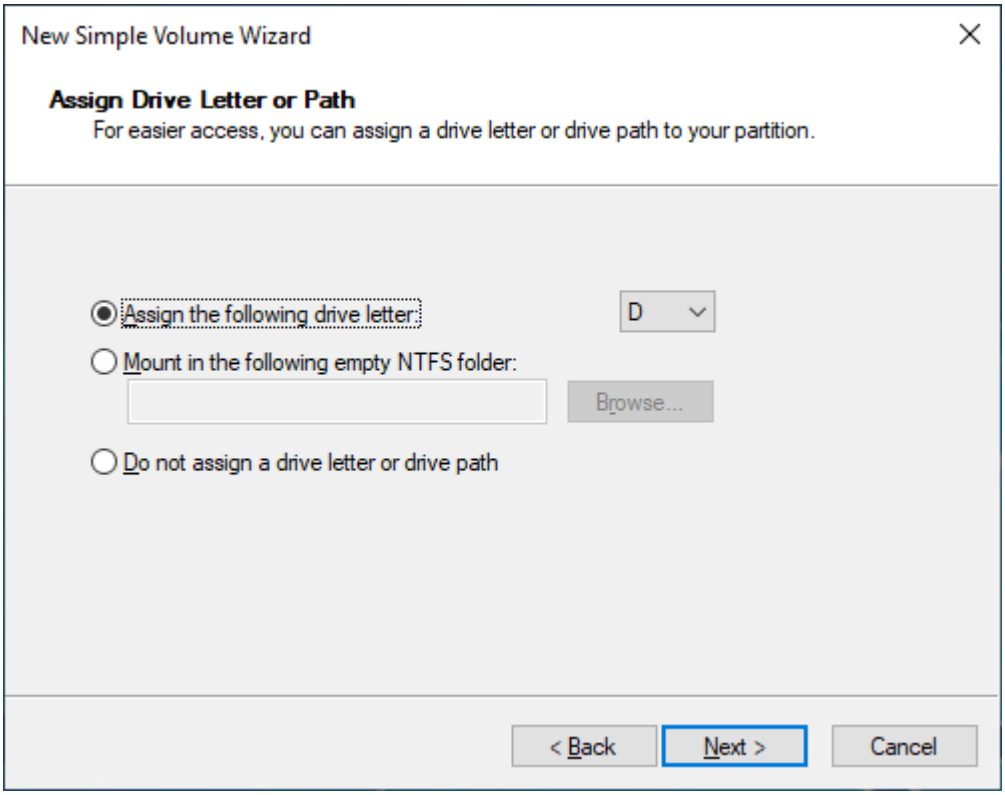


6. Di Wizard Volumer Sederhana Baru, pilih Berikutnya.

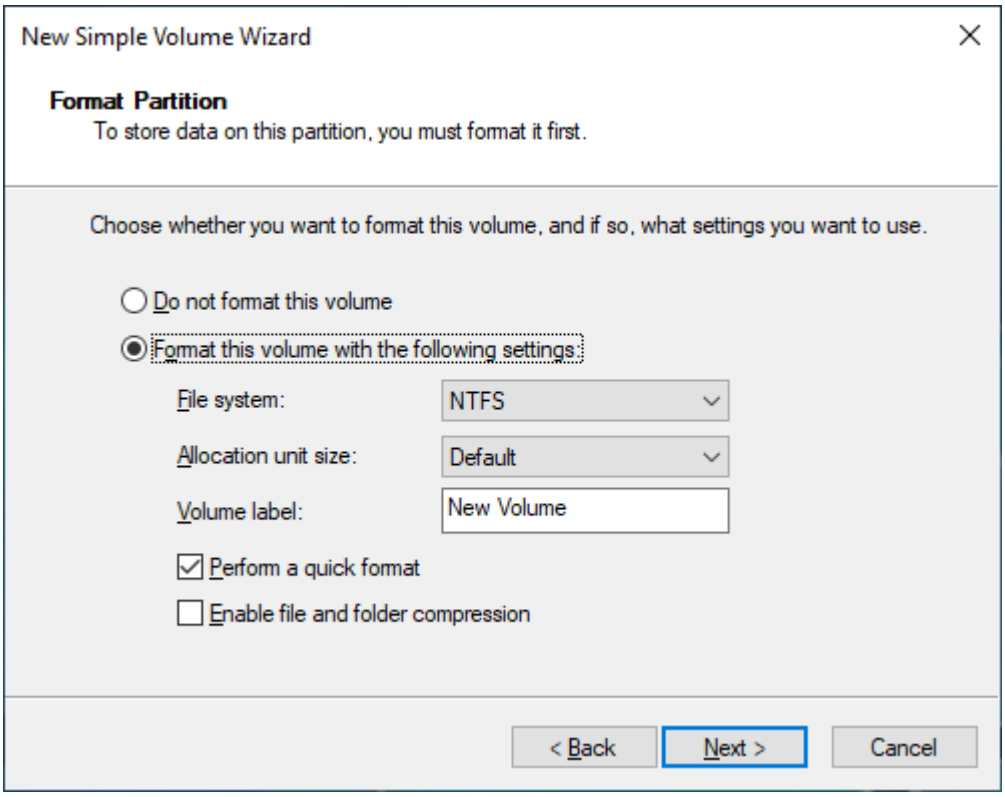
7. Jika Anda ingin mengubah nilai maksimum default, tentukan Ukuran volume sederhana dalam MB, lalu pilih Berikutnya.



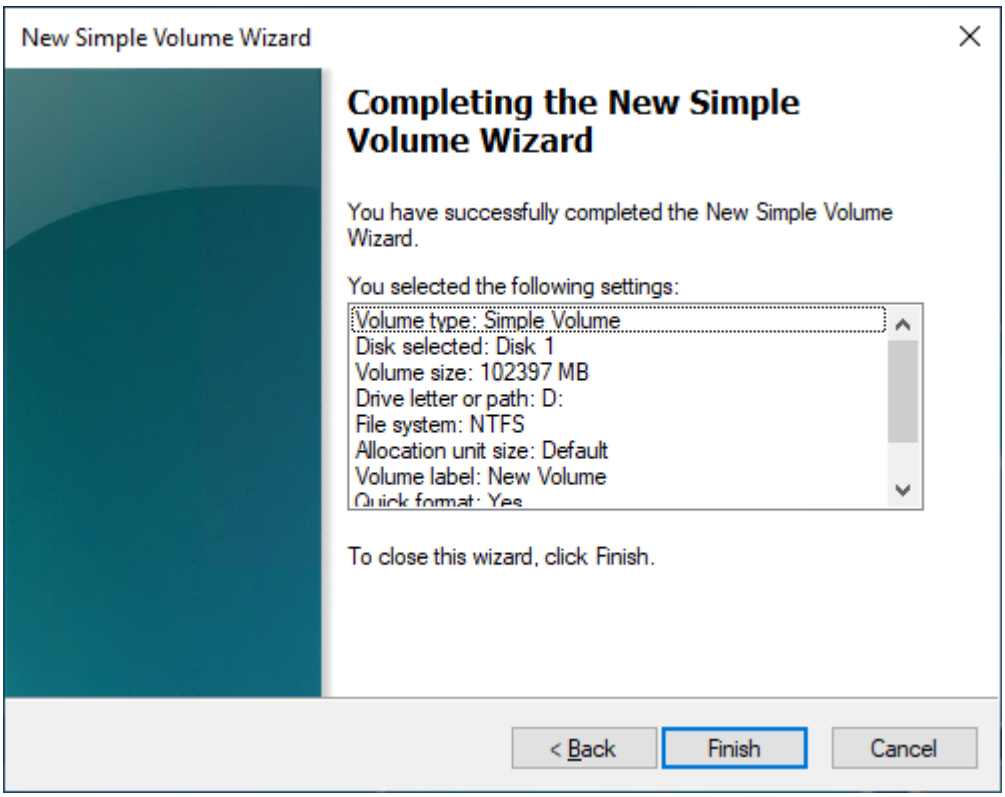
8. Tentukan huruf drive yang disukai, jika perlu, di dalam menu dropdown Tetapkan huruf drive berikut, lalu pilih Berikutnya.



9. Tentukan Label Volume dan sesuaikan pengaturan default seperlunya, lalu pilih Berikutnya.



10. Tinjau pengaturan Anda, lalu pilih Selesai untuk menerapkan modifikasi dan tutup wizard Volume Sederhana Baru.



Melihat informasi tentang volume Amazon EBS

Anda dapat melihat informasi deskriptif tentang volume EBS Anda. Misalnya, Anda dapat melihat informasi tentang semua volume di Wilayah tertentu atau melihat informasi terperinci tentang satu volume, termasuk ukurannya, jenis volume, apakah volume dienkripsi, kunci KMS mana yang digunakan untuk mengenkripsi volume, dan contoh spesifik yang dilampirkan volume.

Anda dapat memperoleh informasi tambahan tentang volume EBS Anda, seperti berapa banyak ruang disk yang tersedia, dari sistem operasi pada instans.

Topik

- Lihat informasi volume
- Status volume
- Lihat metrik volume
- Lihat ruang disk kosong

Lihat informasi volume

Anda dapat melihat informasi tentang volume menggunakan salah satu metode berikut.

Console

Untuk melihat informasi tentang volume menggunakan konsol

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Volume.
3. Untuk mengurangi daftar, Anda dapat memfilter volume menggunakan tanda dan atribut volume. Pilih bidang filter, pilih atribut tanda atau volume, lalu pilih nilai filter.
4. Untuk melihat informasi lebih lanjut tentang volume, pilih ID.

Untuk melihat volume EBS yang dilampirkan ke suatu instans menggunakan konsol

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Instans.
3. Pilih instans.
4. Pada tab Penyimpanan, bagian Perangkat blok mencantumkan volume yang dilampirkan ke instans. Untuk melihat informasi tentang volume tertentu, pilih ID-nya di kolom ID Volume.

Amazon EC2 Global View

Anda dapat menggunakan Amazon EC2 Global View untuk melihat volume Anda di semua Wilayah tempat akun AWS Anda diaktifkan. Untuk informasi selengkapnya, lihat Tampilan Global Amazon EC2.

AWS CLI

Untuk melihat informasi tentang volume EBS menggunakan AWS CLI

Gunakan perintah describe-volumes.

Tools for Windows PowerShell

Untuk melihat informasi tentang volume EBS menggunakan Alat untuk Windows PowerShell

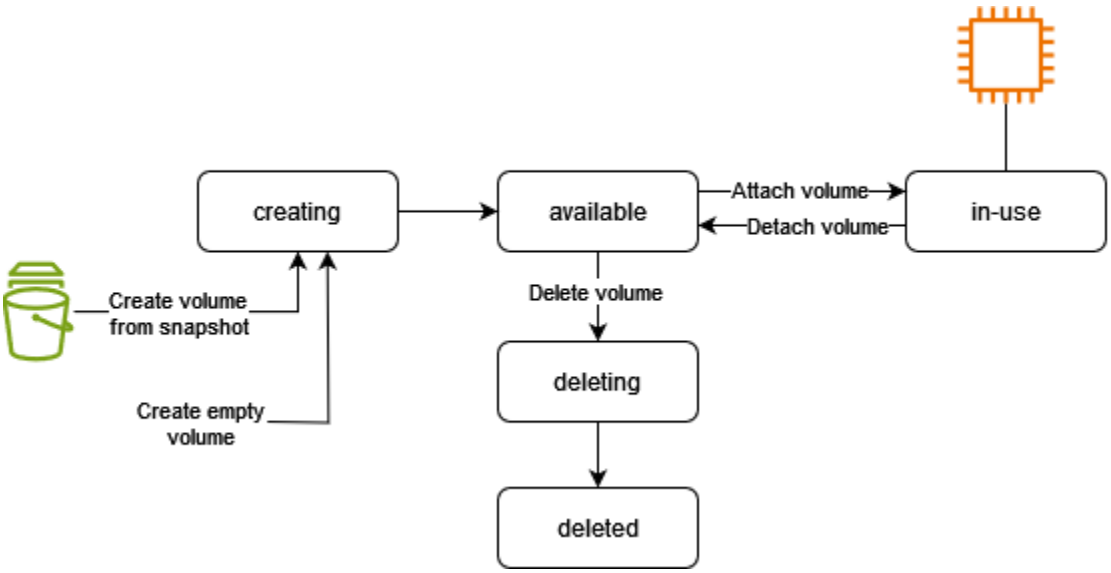
Gunakan perintah Get-EC2Volume.

Status volume

Status volume menjelaskan ketersediaan volume Amazon EBS. Anda dapat melihat status volume di kolom Status pada halaman Volume di konsol, atau dengan menggunakan perintah AWS CLI deskripsikan volume.

Volume Amazon EBS bertransisi melalui status yang berbeda dari saat dibuat hingga dihapus.

Ilustrasi berikut menunjukkan transisi antara status volume. Anda dapat membuat volume dari snapshot Amazon EBS atau membuat volume kosong. Saat Anda membuat volume, itu memasuki `creating` status. Setelah volume siap digunakan, ia memasuki `available` keadaan. Anda dapat melampirkan volume yang tersedia ke instance di Availability Zone yang sama dengan volume. Anda harus melepaskan volume sebelum Anda dapat melampirkannya ke instance lain atau menghapusnya. Anda dapat menghapus volume saat Anda tidak lagi membutuhkannya.



Tabel berikut merangkum status volume.

Status	Deskripsi
`creating`	Volume sedang dibuat.
`available`	Volume tidak terlampir pada suatu instans.
`in-use`	Volume terlampir pada suatu instans.
`deleting`	Volume sedang dihapus.

Status	Deskripsi
`deleted`	Volume dihapus.
`error`	Perangkat keras dasar yang terkait dengan volume EBS Anda gagal, dan data yang terkait dengan volume tidak dapat dipulihkan. Untuk informasi tentang cara memulihkan volume atau memulihkan data pada volume, lihat Volume EBS saya memiliki status "kesalahan".

Lihat metrik volume

Anda bisa mendapatkan informasi tambahan tentang volume EBS Anda dari Amazon CloudWatch. Untuk informasi selengkapnya, lihat CloudWatch Metrik Amazon untuk Amazon EBS.

Lihat ruang disk kosong

Instans Linux

Anda dapat memperoleh informasi tambahan tentang volume EBS Anda, seperti berapa banyak ruang disk yang tersedia, dari sistem operasi Linux di instans. Misalnya, gunakan perintah berikut:

```
[ec2-user ~]$ df -hT /dev/xvda1
Filesystem     Type      Size  Used Avail Use% Mounted on
/dev/xvda1     xfs       8.0G  1.2G  6.9G  15% /
```

> **Tip**
>
> Anda juga dapat menggunakan CloudWatch agen untuk mengumpulkan metrik penggunaan ruang disk dari instans Amazon EC2 tanpa menghubungkan ke instans. Untuk informasi selengkapnya, lihat Membuat file konfigurasi CloudWatch agen dan Menginstal CloudWatch agen di Panduan CloudWatch Pengguna Amazon. Jika Anda perlu memantau penggunaan ruang disk untuk beberapa instance, Anda dapat menginstal dan mengonfigurasi CloudWatch agen pada instance tersebut menggunakan Systems Manager. Untuk informasi selengkapnya, lihat Menginstal CloudWatch agen menggunakan Systems Manager.

Untuk informasi tentang melihat ruang disk kosong pada instance Windows, lihat Melihat ruang disk kosong di Panduan Pengguna Amazon EC2.

Instans Windows

Anda dapat memperoleh informasi tambahan tentang volume EBS Anda, seperti berapa banyak ruang disk yang tersedia, dari sistem operasi Windows pada instans. Misalnya, Anda dapat melihat ruang disk kosong dengan membuka File Explorer dan memilih PC ini.

Anda juga dapat melihat ruang kosong disk menggunakan perintah `dir` dan memeriksa baris terakhir dari output:

```
C:\> dir C:
 Volume in drive C has no label.
 Volume Serial Number is 68C3-8081

 Directory of C:\

03/25/2018  02:10 AM    <DIR>          .
03/25/2018  02:10 AM    <DIR>          ..
03/25/2018  03:47 AM    <DIR>          Contacts
03/25/2018  03:47 AM    <DIR>          Desktop
03/25/2018  03:47 AM    <DIR>          Documents
03/25/2018  03:47 AM    <DIR>          Downloads
03/25/2018  03:47 AM    <DIR>          Favorites
03/25/2018  03:47 AM    <DIR>          Links
03/25/2018  03:47 AM    <DIR>          Music
03/25/2018  03:47 AM    <DIR>          Pictures
03/25/2018  03:47 AM    <DIR>          Saved Games
03/25/2018  03:47 AM    <DIR>          Searches
03/25/2018  03:47 AM    <DIR>          Videos
               0 File(s)              0 bytes
              13 Dir(s)  18,113,662,976 bytes free
```

Anda juga dapat melihat ruang kosong disk menggunakan perintah `fsutil` berikut:

```
C:\> fsutil volume diskfree C:
Total # of free bytes        : 18113204224
Total # of bytes             : 32210153472
Total # of avail free bytes  : 18113204224
```

> **Tip**
>
> Anda juga dapat menggunakan CloudWatch agen untuk mengumpulkan metrik penggunaan ruang disk dari instans Amazon EC2 tanpa menghubungkan ke instans. Untuk informasi selengkapnya, lihat Membuat file konfigurasi CloudWatch agen dan Menginstal CloudWatch agen di Panduan CloudWatch Pengguna Amazon. Jika Anda perlu memantau penggunaan ruang disk untuk beberapa instance, Anda dapat menginstal dan mengonfigurasi CloudWatch agen pada instance tersebut menggunakan Systems Manager. Untuk informasi selengkapnya, lihat Menginstal CloudWatch agen menggunakan Systems Manager.

Untuk informasi tentang melihat ruang disk kosong pada instance Linux, lihat Melihat ruang disk kosong di Panduan Pengguna Amazon EC2.

Ubah volume menggunakan Amazon EBS Elastic Volumes

Dengan Volume Elastis Amazon EBS, Anda dapat meningkatkan ukuran volume, mengubah tipe volume, atau menyesuaikan performa volume EBS Anda. Jika instans Anda mendukung Volume Elastis, Anda dapat melakukannya tanpa melepas volume atau memulai ulang instans tersebut. Hal ini memungkinkan Anda untuk terus menggunakan aplikasi Anda saat perubahan berlaku.

Tidak ada biaya untuk mengubah konfigurasi volume. Anda dikenakan biaya untuk konfigurasi volume baru setelah modifikasi volume dimulai. Untuk informasi selengkapnya, lihat halaman Harga Amazon EBS.

Daftar Isi

Persyaratan untuk modifikasi volume EBS

Persyaratan dan batasan berikut berlaku ketika Anda memodifikasi volume Amazon EBS. Untuk mempelajari selengkapnya tentang persyaratan umum untuk volume EBS, lihat Batasan ukuran dan konfigurasi volume EBS.

Topik

- Tipe instans yang didukung
- Sistem operasi
- Batasan

Tipe instans yang didukung

Volume Elastis mendukung pada instans berikut:

- Semua instance generasi saat ini
- Instans generasi sebelumnya sebagai berikut: C1, C3, C4, G2, I2, M1, M3, M4, R3, dan R4

Jika tipe instans Anda tidak mendukung Volume Elastis, lihat Modifikasi volume EBS jika Volume Elastis tidak mendukungnya.

Sistem operasi

Persyaratan sistem operasi berikut berlaku:

Linux

AMI Linux memerlukan tabel partisi GUID (GPT) dan GRUB 2 untuk volume boot 2 TiB (2.048 GiB) atau lebih besar. Banyak AMI Linux saat ini yang masih menggunakan skema partisi MBR, yang hanya mendukung ukuran volume boot hingga 2 TiB. Jika instans Anda tidak melakukan boot dengan volume boot yang lebih besar dari 2 TiB, AMI yang Anda gunakan mungkin dibatasi pada ukuran volume boot kurang dari 2 TiB. Volume non-boot tidak memiliki batasan ini pada instans Linux. Untuk persyaratan yang memengaruhi volume Windows, lihat Persyaratan untuk volume Windows di Panduan Pengguna Amazon EC2.

Sebelum mencoba mengubah ukuran volume booting melebihi 2 TiB, Anda dapat menentukan apakah volume tersebut menggunakan partisi MBR atau GPT dengan menjalankan perintah berikut pada instans Anda:

```
[ec2-user ~]$ sudo gdisk -l /dev/xvda
```

Instans Amazon Linux dengan partisi GPT mengembalikan informasi berikut:

```
GPT fdisk (gdisk) version 0.8.10

  Partition table scan:
```

```
    MBR: protective
    BSD: not present
    APM: not present
    GPT: present

  Found valid GPT with protective MBR; using GPT.
```

Instans SUSE dengan partisi MBR mengembalikan informasi berikut:

```
GPT fdisk (gdisk) version 0.8.8

  Partition table scan:
    MBR: MBR only
    BSD: not present
    APM: not present
    GPT: not present
```

Windows

Secara default, Windows menginisialisasi volume dengan tabel partisi Master Boot Record (MBR). Karena MBR hanya mendukung volume yang lebih kecil dari 2 TiB (2.048 GiB), Windows mencegah Anda mengubah ukuran volume MBR melebihi batas ini. Dalam kasus seperti itu, opsi Perpanjang Volume dinonaktifkan di utilitas Manajemen Disk Windows. Jika Anda menggunakan AWS Management Console atau AWS CLI untuk membuat volume yang dipartisi MBR yang melebihi batas ukuran, Windows tidak dapat mendeteksi atau menggunakan ruang tambahan. Untuk persyaratan yang memengaruhi volume Linux, lihat Persyaratan untuk volume Linux di Panduan Pengguna Amazon EC2.

Untuk mengatasi keterbatasan ini, Anda dapat membuat volume baru yang lebih besar dengan tabel partisi GUID (GPT) dan menyalin data dari volume MBR asli.

Untuk membuat volume GPT

1. Buat volume kosong baru dengan ukuran yang diinginkan di Zona ketersediaan instans EC2 dan lampirkan ke instans Anda.

   > **Note**
   >
   > Volume baru tidak boleh berupa volume yang dipulihkan dari snapshot.

2. Masuk ke sistem Windows Anda dan buka Manajemen Disk (diskmgmt.exe).

3. Buka menu konteks (klik kanan) untuk disk baru dan pilih Online.
4. Di jendela Inisialisasi Disk, pilih disk baru dan pilih GPT (Tabel Partisi GUID), OK.
5. Saat inisialisasi selesai, salin data dari volume asli ke volume baru, menggunakan alat seperti robocopy atau teracopy.
6. Di Manajemen Disk, ubah huruf drive ke nilai yang sesuai dan ambil volume lama secara offline.
7. Di konsol Amazon EC2 lepaskan volume lama dari instans, boot ulang instans untuk memverifikasi bahwa berfungsi dengan benar, dan menghapus volume lama.

Batasan

- Ada batasan untuk penyimpanan agregat maksimum yang dapat diminta di modifikasi volume. Untuk informasi selengkapnya, lihat Kuota layanan Amazon EBS di Referensi Umum Amazon Web Services.
- Setelah memodifikasi volume, Anda harus menunggu setidaknya enam jam dan memastikan volume berada dalam status `in-use` atau `available` sebelum dapat memodifikasi volume yang sama.
- Mengubah volume EBS dapat memakan waktu mulai dari hitungan menit hingga hitungan jam, bergantung pada perubahan konfigurasi yang diterapkan. Volume EBS yang berukuran 1 TiB biasanya membutuhkan waktu hingga enam jam untuk dimodifikasi. Namun, volume yang sama dapat memakan waktu 24 jam atau lebih dalam situasi lain. Waktu yang diperlukan untuk memodifikasi volume tidak selalu berskala linier. Oleh karena itu, volume yang lebih besar mungkin membutuhkan waktu yang lebih sedikit, dan volume yang lebih kecil mungkin membutuhkan waktu yang lebih banyak.
- Jika volume dipasang sebelum 3 November 2016 pukul 23:40 UTC, Anda harus menginisialisasi dukungan Volume Elastis. Untuk informasi selengkapnya, lihat Menginisialisasi dukungan Volume Elastis.
- Jika Anda menemukan pesan kesalahan saat mencoba mengubah volume EBS, atau jika Anda memodifikasi volume EBS yang dilampirkan ke tipe instans generasi sebelumnya, lakukan salah satu langkah berikut:
  - Untuk volume non-root, lepaskan volume dari instans, terapkan modifikasi, kemudian lampirkan kembali volume.
  - Untuk volume root, hentikan instans, terapkan modifikasi, lalu mulai ulang instans.
- Waktu modifikasi bertambah untuk volume yang tidak diinisialisasi sepenuhnya. Untuk informasi selengkapnya, lihat Inisialisasi volume Amazon EBS.

- Ukuran volume baru tidak dapat melebihi kapasitas yang didukung dari sistem file dan skema partisi. Untuk informasi selengkapnya, lihat Batasan ukuran dan konfigurasi volume EBS.
- Jika Anda memodifikasi tipe volume, ukuran dan performa harus berada dalam batas tipe volume target. Untuk informasi selengkapnya, lihat Tipe volume Amazon EBS
- Anda tidak dapat mengurangi ukuran volume EBS. Namun, Anda dapat membuat volume yang lebih kecil dan kemudian memigrasikan data Anda ke sana menggunakan alat tingkat aplikasi seperti rsync (instance Linux) atau robocopy (instance Windows).
- Setelah menyediakan lebih dari 32.000 IOPS pada volume `io1` dan `io2` yang ada, Anda mungkin perlu melepaskan dan melampirkan kembali volume tersebut, atau memulai ulang instans untuk melihat peningkatan kinerja penuh.
- `io2`volume yang melekat pada instans yang dibangun di atas Nitro System mendukung ukuran hingga 64 TiB dan IOPS hingga 256.000 IOPS. `io2`volume yang melekat pada instans lain mendukung ukuran hingga 16 TiB dan IOPS hingga 64.000, tetapi dapat mencapai kinerja hingga 32.000 IOPS saja.
- Anda tidak dapat memodifikasi tipe volume dari volume `io2` yang diaktifkan Multi-Lampiran.
- Anda tidak dapat memodifikasi tipe volume, ukuran, atau IOPS yang tersedia dari volume `io1` yang diaktifkan Multi-Lampiran.
- Volume root tipe `io1`, `io2`, `gp2`, `gp3`, atau `standard` tidak dapat dimodifikasi menjadi volume `st1` atau `sc1`, bahkan jika itu dilepas dari instans.
- Meskipun instans `m3.medium` sepenuhnya mendukung modifikasi volume, `m3.large`, `m3.xlarge`, dan instans `m3.2xlarge` mungkin tidak mendukung semua fitur modifikasi volume.

Minta perubahan pada Volume EBS Anda

Dengan Elastic Volume, Anda dapat secara dinamis meningkatkan ukuran, menambah atau mengurangi kinerja, dan mengubah tipe volume dari volume Amazon EBS Anda tanpa melepaskannya.

Gunakan proses berikut ketika memodifikasi volume:

1. (Opsional) Sebelum memodifikasi volume yang berisi data berharga, praktik terbaiknya adalah membuat snapshot volume jika Anda perlu membatalkan perubahan. Untuk informasi selengkapnya, lihat Membuat snapshot Amazon EBS.
2. Minta modifikasi volume.

3. Memantau kemajuan modifikasi volume. Untuk informasi selengkapnya, lihat Pantau kemajuan modifikasi volume EBS.
4. Jika ukuran volume dimodifikasi, perluas sistem file volume untuk memanfaatkan peningkatan kapasitas penyimpanan. Untuk informasi selengkapnya, lihat Perluas sistem file setelah mengubah ukuran volume EBS.

Daftar Isi

Modifikasi volume EBS menggunakan Volume Elastis

Pertimbangan

Ingatlah hal-hal berikut ini saat memodifikasi volume:

- Setelah memodifikasi volume, Anda harus menunggu setidaknya enam jam dan memastikan volume berada dalam status `in-use` atau `available` sebelum dapat memodifikasi volume yang sama.
- Mengubah volume EBS dapat memakan waktu mulai dari hitungan menit hingga hitungan jam, bergantung pada perubahan konfigurasi yang diterapkan. Volume EBS yang berukuran 1 TiB biasanya membutuhkan waktu hingga enam jam untuk dimodifikasi. Namun, volume yang sama dapat memakan waktu 24 jam atau lebih dalam situasi lain. Waktu yang diperlukan untuk memodifikasi volume tidak selalu berskala linier. Oleh karena itu, volume yang lebih besar mungkin membutuhkan waktu yang lebih sedikit, dan volume yang lebih kecil mungkin membutuhkan waktu yang lebih banyak.
- Anda tidak dapat membatalkan permintaan modifikasi volume setelah dikirimkan.
- Anda hanya dapat meningkatkan ukuran volume. Anda tidak dapat mengurangi ukuran volume.
- Anda dapat meningkatkan atau mengurangi performa volume.
- Jika Anda tidak mengubah tipe volume, ukuran volume dan modifikasi performa harus dalam batas tipe volume saat ini. Jika Anda mengubah tipe volume, ukuran volume dan modifikasi performa harus dalam batas tipe volume target

- Jika Anda memodifikasi tipe volume dari gp2 ke gp3, dan Anda tidak menentukan performa IOPS atau throughput, Amazon EBS secara otomatis menetapkan performa yang setara dengan volume gp2 sumber, atau performa gp3 dasar, mana saja yang lebih tinggi.

  Misalnya, jika Anda memodifikasi volume gp2 500 GiB dengan throughput 250 MiB/dtk dan 1500 IOPS ke gp3 tanpa menentukan IOPS atau performa throughput, Amazon EBS secara otomatis menyediakan volume gp3 dengan 3000 IOPS (IOPS gp3 dasar) dan 250 MiB/dtk (untuk mencocokkan throughput volume gp2 sumber).

Untuk mengubah volume EBS, gunakan salah satu metode berikut.

Console

Untuk memodifikasi volume EBS menggunakan konsol

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Volume.
3. Pilih volume yang akan dimodifikasi dan pilih Tindakan, Ubah volume.
4. Layar Ubah volume menampilkan ID volume dan konfigurasi volume saat ini volume, termasuk tipe, ukuran, IOPS, dan throughput. Atur nilai konfigurasi baru sebagai berikut:
   - Untuk mengubah tipe, pilih nilai untuk Tipe Volume.
   - Untuk memodifikasi ukuran, masukkan nilai baru untuk Ukuran.
   - (gp3, io1, dan io2 saja) Untuk memodifikasi IOPS, masukkan nilai baru untuk IOPS.
   - (gp3 saja) Untuk memodifikasi throughput, masukkan nilai baru untuk Throughput.
5. Setelah Anda selesai mengubah pengaturan volume, pilih Modifikasi. Saat diminta konfirmasi, pilih Ubah.
6. **Important**

   Jika Anda telah meningkatkan ukuran volume, Anda juga harus memperluas partisi volume untuk memanfaatkan kapasitas penyimpanan tambahan. Untuk informasi selengkapnya, lihat Perluas sistem file setelah mengubah ukuran volume EBS.
7. (Hanya instance Windows) Jika Anda meningkatkan ukuran volume NVMe pada instance yang tidak memiliki driver AWS NVMe, Anda harus me-reboot instance untuk mengaktifkan

Windows untuk melihat ukuran volume baru. Untuk informasi selengkapnya tentang menginstal driver AWS NVMe, lihat driver AWS NVMe untuk instance Windows.

AWS CLI

Untuk memodifikasi volume EBS menggunakan AWS CLI

Gunakan perintah modify-volume untuk memodifikasi satu atau lebih pengaturan konfigurasi untuk volume. Misalnya, jika Anda memiliki volume tipe gp2 ukuran 100 GiB, perintah berikut mengubah konfigurasinya menjadi volume tipe `io1` 10.000 IOPS dan ukuran 200 GiB.

```
aws ec2 modify-volume --volume-type io1 --iops 10000 --size 200 --volume-
id vol-1111111111111111
```

Berikut ini adalah output contoh:

```
{
    "VolumeModification": {
        "TargetSize": 200,
        "TargetVolumeType": "io1",
        "ModificationState": "modifying",
        "VolumeId": "vol-1111111111111111",
        "TargetIops": 10000,
        "StartTime": "2017-01-19T22:21:02.959Z",
        "Progress": 0,
        "OriginalVolumeType": "gp2",
        "OriginalIops": 300,
        "OriginalSize": 100
    }
}
```

> **Important**
>
> Jika Anda telah meningkatkan ukuran volume, Anda juga harus memperluas partisi volume untuk memanfaatkan kapasitas penyimpanan tambahan. Untuk informasi selengkapnya, lihat Perluas sistem file setelah mengubah ukuran volume EBS.

Menginisialisasi dukungan Volume Elastis (jika diperlukan)

Sebelum Anda dapat memodifikasi volume yang telah dipasang ke suatu instans sebelum 3 November 2016 pada 23:40 UTC, Anda harus inisialisasi dukungan modifikasi volume menggunakan salah satu tindakan berikut:

- Lepaskan dan pasang volume
- Hentikan dan mulai instans

Gunakan salah satu prosedur berikut untuk menentukan apakah proses Anda siap untuk modifikasi volume.

Console

Untuk menentukan apakah instans Anda siap menggunakan konsol

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Instans.
3. Pilih ikon Tampilkan/Sembunyikan Kolom (roda gigi). Pilih Waktu peluncuran dan kemudian pilih Konfirmasi.
4. Sortir daftar instans berdasarkan Waktu Peluncuran yang berbeda. Untuk setiap instans yang dimulai sebelum tanggal batas, pilih Penyimpanan dan periksa Waktu lampiran untuk melihat kapan volume dilampirkan.

AWS CLI

Untuk menentukan apakah instans Anda siap menggunakan CLI

Gunakan perintah describe-instances berikut ini untuk menentukan apakah volume dipasang sebelum 3 November 2016 23:40 UTC.

```
aws ec2 describe-instances --query "Reservations[*].Instances[*].
[InstanceId,LaunchTime<='2016-11-01',BlockDeviceMappings[*]
[Ebs.AttachTime<='2016-11-01']]" --output text
```

Baris pertama output untuk setiap instans menampilkan ID dan apakah itu dimulai sebelum batas tanggal (Benar atau Salah). Baris pertama diikuti dengan satu baris atau lebih yang menunjukkan jika setiap volume EBS dipasang sebelum tanggal batas (Benar atau Salah). Dalam contoh output

berikut, Anda harus menginisialisasi modifikasi volume untuk instans pertama karena itu dimulai sebelum tanggal batas dan volume root dipasang sebelum tanggal batas. Instans lainnya sudah siap karena sudah dimulai setelah tanggal batas.

```
i-e905622e              True
True
i-719f99a8              False
True
i-006b02c1b78381e57     False
False
False
i-e3d172ed              False
True
```

Modifikasi volume EBS jika Volume Elastis tidak mendukungnya

Jika Anda menggunakan tipr instans yang didukung, Anda dapat menggunakan Volume Elastis untuk secara dinamis mengubah ukuran, performa, dan tipe volume Amazon EBS Anda tanpa melepaskannya.

Jika Anda tidak dapat menggunakan Volume Elastis, tetapi Anda perlu memodifikasi volume root (boot), Anda harus menghentikan instans, memodifikasi volume, kemudian memulai ulang instansnya.

Setelah instans dimulai, Anda dapat memeriksa ukuran sistem file untuk melihat apakah instans Anda mengenali ruang volume yang lebih besar. Di Linux, gunakan df -h perintah untuk memeriksa ukuran sistem file.

```
[ec2-user ~]$ df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/xvda1            7.9G  943M  6.9G  12% /
tmpfs                 1.9G     0  1.9G   0% /dev/shm
```

Jika ukuran tidak mencerminkan volume yang baru diperluas, Anda harus memperluas sistem file perangkat Anda sehingga instans Anda dapat menggunakan ruang baru. Untuk informasi selengkapnya, lihat Perluas sistem file setelah mengubah ukuran volume EBS.

Dengan contoh Windows, Anda mungkin harus membawa volume online untuk menggunakannya. Untuk informasi selengkapnya, lihat Buat volume Amazon EBS tersedia untuk digunakan. Anda tidak perlu memformat ulang volume.

Pantau kemajuan modifikasi volume EBS

Saat Anda memodifikasi volume EBS, perubahan itu akan melewati urutan status. Volume memasuki status `modifying`, status `optimizing`, dan terakhir status `completed`. Pada titik ini, volume siap untuk dimodifikasi selengkapnya.

> **Note**
>
> Jarang, AWS kesalahan sementara dapat mengakibatkan keadaan. `failed` Ini bukan indikasi kesehatan volume; itu hanya menunjukkan bahwa modifikasi terhadap volume gagal. Jika ini terjadi, coba kembali modifikasi volume.

Saat volume berada pada status `optimizing`, kinerja volume Anda ada di antara spesifikasi konfigurasi sumber dan target. Performa volume transisi tidak akan kurang dari performa volume sumber. Jika Anda menurunkan IOPS, performa volume transisi tidak kurang dari performa volume target.

Perubahan modifikasi volume berlaku sebagai berikut:

- Perubahan ukuran biasanya memakan waktu beberapa detik dan berlaku setelah volume bertransisi ke status `0ptimizing`.
- Perubahan Performa (IOPS) dapat berlangsung dari beberapa menit ke beberapa jam untuk menyelesaikan dan tergantung pada perubahan konfigurasi yang dibuat.
- Dalam beberapa kasus, konfigurasi baru dapat diterapkan lebih dari 24 jam, seperti ketika volume belum sepenuhnya diinisialisasi. Biasanya, volume 1-TiB yang digunakan sepenuhnya membutuhkan waktu sekitar 6 jam untuk bermigrasi ke konfigurasi performa baru.

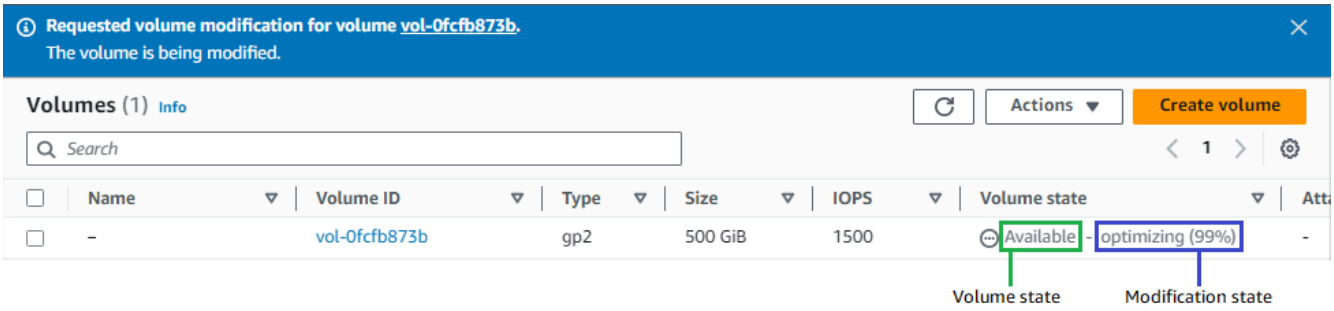
Gunakan salah satu metode berikut untuk memantau kemajuan perubahan suatu volume.

Console

Untuk memantau kemajuan suatu modifikasi menggunakan konsol Amazon EC2

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Volume.
3. Pilih volume.

4. Kolom Status Volume dan bidang Status volume di tab Detail berisi informasi dalam format berikut: *Status volume* - *Status modifikasi* (*Kemajuan modifikasi*%). Gambar berikut menunjukkan status modifikasi volume dan volume.



   Status volume yang mungkin adalah `creating`, `available`, `in-use`, `deleting`, `deleted`, dan `error`.

   Status modifikasi yang mungkin adalah `modifying`, `optimizing`, dan `completed`.

   Setelah modifikasi selesai, hanya status volume yang ditampilkan. Status modifikasi dan kemajuan tidak lagi ditampilkan.

AWS CLI

Untuk memantau kemajuan modifikasi menggunakan AWS CLI

Gunakan describe-volumes-modifications untuk melihat kemajuan dari satu atau lebih modifikasi volume. Contoh berikut menjelaskan modifikasi volume untuk dua volume.

```
aws ec2 describe-volumes-modifications --volume-
ids vol-1111111111111111 vol-2222222222222222
```

Dalam contoh output berikut, modifikasi volume masih dalam status `modifying`. Kemajuan dilaporkan dalam bentuk persentase.

```
{
    "VolumesModifications": [
        {
            "TargetSize": 200,
            "TargetVolumeType": "io1",
            "ModificationState": "modifying",
            "VolumeId": "vol-1111111111111111",
```

```
            "TargetIops": 10000,
            "StartTime": "2017-01-19T22:21:02.959Z",
            "Progress": 0,
            "OriginalVolumeType": "gp2",
            "OriginalIops": 300,
            "OriginalSize": 100
        },
        {
            "TargetSize": 2000,
            "TargetVolumeType": "sc1",
            "ModificationState": "modifying",
            "VolumeId": "vol-22222222222222222",
            "StartTime": "2017-01-19T22:23:22.158Z",
            "Progress": 0,
            "OriginalVolumeType": "gp2",
            "OriginalIops": 300,
            "OriginalSize": 1000
        }
    ]
}
```

Contoh berikutnya menggambarkan semua volume dengan status modifikasi `optimizing` atau `completed`, lalu memfilter dan memformat hasil untuk hanya menampilkan modifikasi yang dimulai pada atau setelah 1 Februari 2017:

```
aws ec2 describe-volumes-modifications --filters Name=modification-
state,Values="optimizing","completed" --query "VolumesModifications[?
StartTime>='2017-02-01'].{ID:VolumeId,STATE:ModificationState}"
```

Berikut ini adalah contoh output dengan informasi tentang dua volume:

```
[
    {
        "STATE": "optimizing",
        "ID": "vol-06397e7a0eEXAMPLE"
    },
    {
        "STATE": "completed",
        "ID": "vol-ba74e18c2aEXAMPLE"
    }
]
```

CloudWatch Events console

Dengan CloudWatch Acara, Anda dapat membuat aturan notifikasi untuk peristiwa modifikasi volume. Anda dapat menggunakan aturan Anda untuk membuat pesan notifikasi menggunakan Amazon SNS atau untuk menginvokasi Fungsi Lambda sebagai respons atas peristiwa yang cocok. Peristiwa dipancarkan atas dasar upaya terbaik.

Untuk memantau kemajuan modifikasi menggunakan CloudWatch Events

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Pilih Peristiwa, Buat aturan.
3. Untuk Bangun pola peristiwa untuk mencocokkan peristiwa berdasarkan layanan, pilih Pola peristiwa kustom.
4. Untuk Bangun pola peristiwa kustom, ganti konten dengan berikut dan pilih Simpan.

```
{
  "source": [
    "aws.ec2"
  ],
  "detail-type": [
    "EBS Volume Notification"
  ],
  "detail": {
    "event": [
      "modifyVolume"
    ]
  }
}
```

Berikut contoh data peristiwa:

```
{
   "version": "0",
   "id": "01234567-0123-0123-0123-012345678901",
   "detail-type": "EBS Volume Notification",
   "source": "aws.ec2",
   "account": "012345678901",
   "time": "2017-01-12T21:09:07Z",
   "region": "us-east-1",
   "resources": [
      "arn:aws:ec2:us-east-1:012345678901:volume/vol-03a55cf56513fa1b6"
```

```
    ],
    "detail": {
        "result": "optimizing",
        "cause": "",
        "event": "modifyVolume",
        "request-id": "01234567-0123-0123-0123-0123456789ab"
    }
}
```

Perluas sistem file setelah mengubah ukuran volume EBS

Setelah Anda meningkatkan ukuran volume EBS, Anda harus memperluas partisi dan sistem file ke ukuran baru yang lebih besar. Anda dapat melakukan ini segera setelah volume memasuki status `optimizing`.

Sebelum Anda mulai

- Buat snapshot dari volume, jika Anda perlu mengembalikan perubahan Anda. Untuk informasi selengkapnya, lihat Membuat snapshot Amazon EBS.
- Konfirmasikan bahwa modifikasi volume berhasil dan berada dalam status `optimizing` atau `completed`. Untuk informasi selengkapnya, lihat Pantau kemajuan modifikasi volume EBS.
- Pastikan volume dilampirkan ke instans dan diformat dan dipasang. Untuk informasi selengkapnya, lihat Format dan pasang volume yang terpasang.
- (Hanya instance Linux) Jika Anda menggunakan volume logis pada volume Amazon EBS, Anda harus menggunakan Logical Volume Manager (LVM) untuk memperluas volume logis. Untuk petunjuk tentang cara melakukan ini, lihat bagian Perpanjang volume logis di Bagaimana cara membuat volume logis LVM pada seluruh volume EBS? AWS Artikel Pusat Pengetahuan.

Instans Linux

> **Note**
>
> Instruksi berikut memandu Anda melalui proses perluasan sistem file XFS dan Ext4 untuk Linux. Untuk informasi tentang memperluas sistem file yang berbeda, lihat dokumentasinya.

Sebelum Anda dapat memperluas sistem file di Linux, Anda harus memperpanjang partisi, jika volume Anda memilikinya.

Perluas sistem file volume EBS

Gunakan prosedur berikut untuk memperluas sistem file untuk volume yang diubah ukurannya.

Perhatikan bahwa penamaan perangkat dan partisi berbeda untuk instance Xen dan instance yang dibangun di Sistem Nitro. Untuk menentukan apakah instans Anda berbasis Xen atau berbasis Nitro, gunakan perintah describe-instans-types AWS CLI sebagai berikut:

```
[ec2-user ~]$ aws ec2 describe-instance-types --instance-type instance_type --query
 "InstanceTypes[].Hypervisor"
```

`nitro` menunjukkan bahwa instans Anda berbasis Nitro. `xen` atau `xen-on-nitro` menunjukkan bahwa instans Anda berbasis Xen.

Untuk memperluas sistem file volume EBS

1. Terhubung ke instans Anda.
2. Ubah ukuran partisi, jika diperlukan. Untuk melakukannya:

   a. Periksa apakah volume memiliki partisi. Gunakan perintah lsblk.

      Nitro instance example

      Dalam contoh output berikut, volume root (`nvme0n1`) memiliki dua partisi (`nvme0n1p1` dan `nvme0n1p128`), sedangkan volume tambahan (`nvme1n1`) tidak memiliki partisi.

      ```
      [ec2-user ~]$ sudo lsblk
      NAME          MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
      nvme1n1       259:0    0  30G  0 disk /data
      nvme0n1       259:1    0  16G  0 disk
      ##nvme0n1p1   259:2    0   8G  0 part /
      ##nvme0n1p128 259:3    0   1M  0 part
      ```

      Xen instance example

      Dalam contoh output berikut, volume root (`xvda`) memiliki satu partisi (`xvda1`), sedangkan volume tambahan (`xvdf`) tidak memiliki partisi.

      ```
      [ec2-user ~]$ sudo lsblk
      NAME    MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
      xvda    202:0    0  16G  0 disk
      ```

```
##xvda1 202:1    0   8G  0 part /
xvdf    202:80   0  24G  0 disk
```

Jika volume memiliki partisi, lanjutkan prosedur dari langkah berikut (2b). Jika volume tidak memiliki partisi, lewati langkah 2b, 2c, dan 2d, dan lanjutkan prosedur dari langkah 3.

> **Tip pemecahan masalah**
>
> Jika Anda tidak melihat volume dalam output perintah, pastikan volume terlampir ke instans serta diformat dan dipasang.

b. Periksa apakah partisi perlu diperpanjang. Pada output perintah lsblk dari langkah sebelumnya, bandingkan ukuran partisi dan ukuran volume.

   Jika ukuran partisi lebih kecil dari ukuran volume, lanjutkan ke langkah berikutnya. Jika ukuran partisi sama dengan ukuran volume, partisi tidak dapat diperpanjang.

   > **Tip pemecahan masalah**
   >
   > Jika volume masih mencerminkan ukuran aslinya, konfirmasikan bahwa modifikasi volume berhasil.

c. Perluas partisi. Gunakan perintah growpart dan tentukan partisi yang akan diperpanjang.

   Nitro instance example

   Misalnya, untuk memperluas partisi bernama `nvme0n1p1`, gunakan perintah berikut.

   > **Important**
   >
   > Perhatikan ruang antara nama perangkat (`nvme0n1`) dan nomor partisi (1).

   ```
   [ec2-user ~]$ sudo growpart /dev/nvme0n1 1
   ```

   Xen instance example

   Misalnya, untuk memperluas partisi bernama `xvda1`, gunakan perintah berikut.

> **Important**
>
> Perhatikan ruang antara nama perangkat (`xvda`) dan nomor partisi (1).

```
[ec2-user ~]$ sudo growpart /dev/xvda 1
```

> **Tip pemecahan masalah**
>
> - `mkdir: cannot create directory ‘/tmp/growpart.31171’: No space left on device FAILED: failed to make temp dir`: Menunjukkan bahwa tidak ada cukup ruang disk kosong pada volume bagi growpart untuk membuat direktori sementara yang diperlukan untuk melakukan pengubahan ukuran. Kosongkan ruang disk, kemudian coba lagi.
> - `must supply partition-number`: Menunjukkan bahwa Anda menentukan partisi yang salah. Gunakan perintah lsblk untuk mengonfirmasi nama partisi, dan pastikan Anda memasukkan spasi antara nama perangkat dan nomor partisi.
> - `NOCHANGE: partition 1 is size 16773087. it cannot be grown`: Menunjukkan bahwa partisi sudah memperluas seluruh volume dan tidak dapat diperpanjang. Konfirmasikan bahwa modifikasi volume berhasil.

d. Verifikasi bahwa partisi telah diperpanjang. Gunakan perintah lsblk. Ukuran partisi sekarang harus sama dengan ukuran volume.

Nitro instance example

Contoh output berikut menunjukkan bahwa volume (`nvme0n1`) dan partisi (`nvme0n1p1`) berukuran sama (`16 GB`).

```
[ec2-user ~]$ sudo lsblk
NAME          MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
nvme1n1       259:0    0  30G  0 disk /data
nvme0n1       259:1    0  16G  0 disk
##nvme0n1p1   259:2    0  16G  0 part /
##nvme0n1p128 259:3    0   1M  0 part
```

Xen instance example

Contoh output berikut menunjukkan bahwa volume (`xvda`) dan partisi (`xvda1`) berukuran sama (`16 GB`).

```
[ec2-user ~]$ sudo lsblk
NAME    MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvda    202:0    0  16G  0 disk
##xvda1 202:1    0  16G  0 part /
xvdf    202:80   0  24G  0 disk
```

3. Perluas sistem file.

   a. Dapatkan nama, ukuran, tipe, dan titik pemasangan untuk sistem file yang perlu Anda perluas. Gunakan perintah df -hT.

   Nitro instance example

   Contoh output berikut menunjukkan bahwa sistem file `/dev/nvme0n1p1` berukuran 8 GB, bertipe `xfs`, dan titik pemasangannya adalah `/`.

   ```
   [ec2-user ~]$ df -hT
   Filesystem      Type  Size  Used Avail Use% Mounted on
   /dev/nvme0n1p1  xfs   8.0G  1.6G  6.5G  20% /
   /dev/nvme1n1    xfs   8.0G   33M  8.0G   1% /data
   ...
   ```

   Xen instance example

   Contoh output berikut menunjukkan bahwa sistem file `/dev/xvda1` berukuran 8 GB, bertipe `ext4`, dan titik pemasangannya adalah `/`.

   ```
   [ec2-user ~]$ df -hT
   Filesystem      Type   Size    Used   Avail   Use%   Mounted on
   /dev/xvda1      ext4   8.0G    1.9G   6.2G    24%    /
   /dev/xvdf1      xfs    24.0G   45M    8.0G    1%     /data
   ...
   ```

   b. Perintah untuk memperluas sistem file berbeda bergantung pada jenis sistem file. Pilih perintah yang benar berikut berdasarkan tipe sistem file yang Anda ketahui di langkah sebelumnya.

- [Sistem file XFS] Gunakan perintah xfs_growfs dan tentukan titik pemasangan sistem file yang Anda ketahui pada langkah sebelumnya.

  Nitro and Xen instance example

  Misalnya, untuk memperluas sistem file yang dipasang pada `/`, gunakan perintah berikut.

  ```
  [ec2-user ~]$ sudo xfs_growfs -d /
  ```

  Tip pemecahan masalah
  - `xfs_growfs: /data is not a mounted XFS filesystem`: Menunjukkan bahwa Anda menentukan titik pemasangan yang salah, atau sistem file bukan XFS. Untuk memverifikasi titik pemasangan dan tipe sistem file, gunakan perintah df -hT.
  - `data size unchanged, skipping`: Menunjukkan bahwa sistem file sudah memperluas seluruh volume. Jika volume tidak memiliki partisi, konfirmasikan bahwa modifikasi volume berhasil. Jika volume memiliki partisi, pastikan partisi diperluas seperti yang dijelaskan pada langkah 2.

- [Sistem file Ext4] Gunakan perintah resize2fs dan tentukan nama sistem file yang Anda ketahui pada langkah sebelumnya.

  Nitro instance example

  Misalnya, untuk memperluas sistem file yang dipasang yang bernama `/dev/nvme0n1p1`, gunakan perintah berikut.

  ```
  [ec2-user ~]$ sudo resize2fs /dev/nvme0n1p1
  ```

  Xen instance example

  Misalnya, untuk memperluas sistem file yang dipasang yang bernama `/dev/xvda1`, gunakan perintah berikut.

  ```
  [ec2-user ~]$ sudo resize2fs /dev/xvda1
  ```

> **Tip pemecahan masalah**
>
> - `resize2fs: Bad magic number in super-block while trying to open /dev/xvda1`: Menunjukkan bahwa sistem file bukan Ext4. Untuk memverifikasi tipe sistem file, gunakan perintah df -hT.
> - `open: No such file or directory while opening /dev/xvdb1`: Menunjukkan bahwa Anda menentukan partisi yang salah. Untuk memverifikasi partisi, gunakan perintah df -hT.
> - `The filesystem is already 3932160 blocks long. Nothing to do!`: Menunjukkan bahwa sistem file sudah memperluas seluruh volume. Jika volume tidak memiliki partisi, konfirmasikan bahwa modifikasi volume berhasil. Jika volume memiliki partisi, pastikan partisi diperluas seperti yang dijelaskan pada langkah 2.

- [Sistem file lainnya] Lihat dokumentasi untuk sistem file Anda untuk mendapatkan petunjuk.

c. Verifikasi bahwa sistem file telah diperluas. Gunakan perintah df -hT dan konfirmasikan bahwa ukuran sistem file sama dengan ukuran volume.

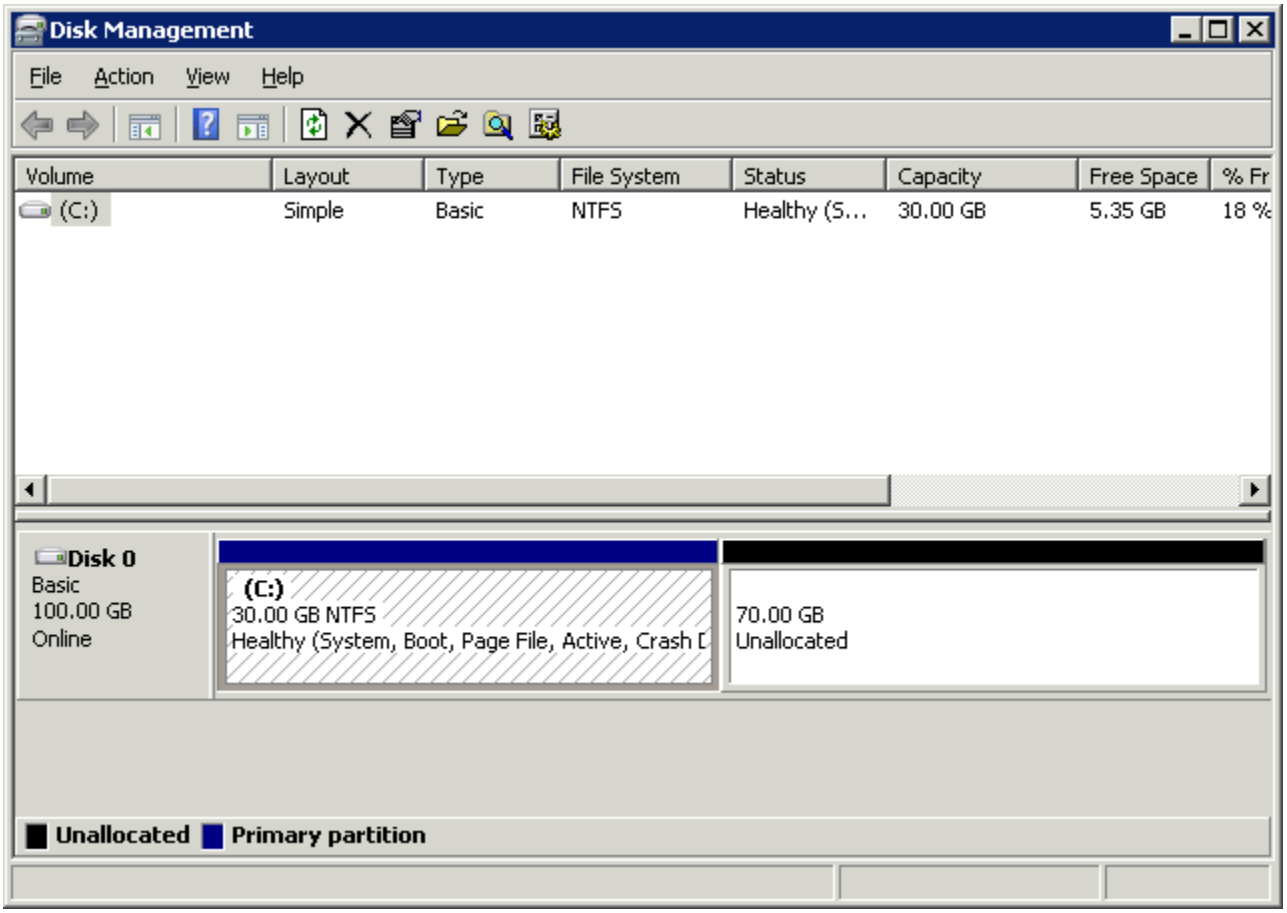
Instans Windows

Gunakan salah satu metode berikut untuk memperluas sistem file pada instance Windows.

Disk Management utility

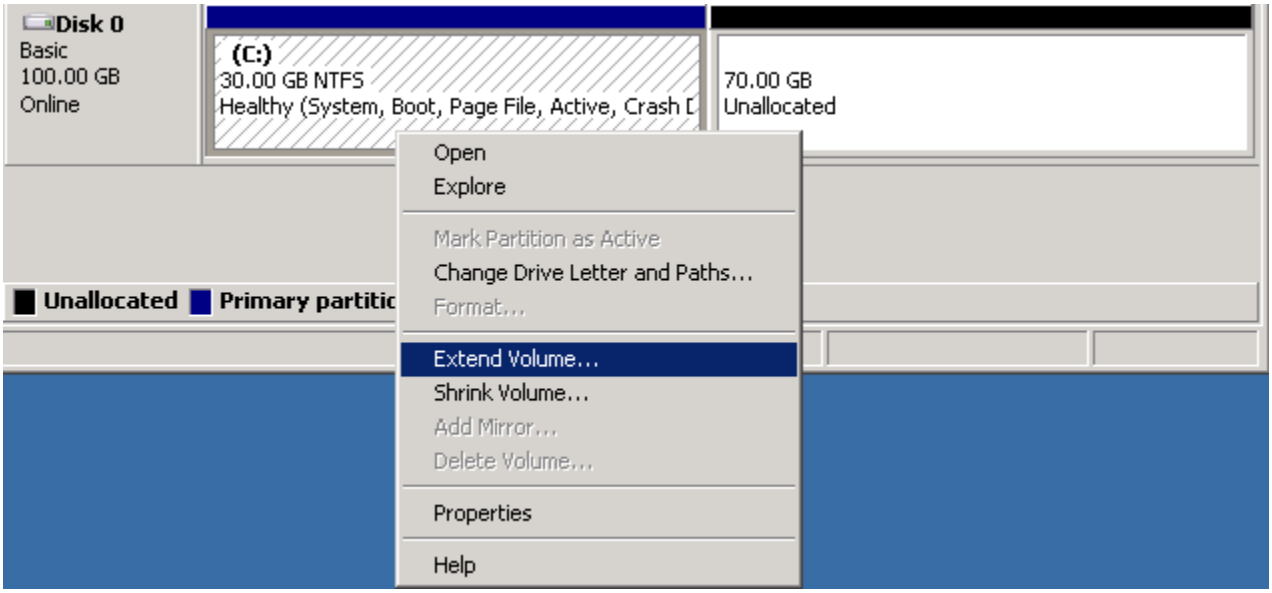
Untuk memperluas sistem file menggunakan Manajemen Disk

1. Sebelum memperluas sistem file yang berisi data berharga, praktik terbaiknya adalah membuat snapshot volume yang berisi data tersebut jika Anda perlu mengembalikan perubahan Anda. Untuk informasi selengkapnya, lihat Membuat snapshot Amazon EBS.
2. Masuk ke instans Windows menggunakan Remote Desktop.
3. Pada dialog Jalankan, masukkan diskmgmt.msc dan tekan Enter. Utilitas Manajemen Disk terbuka.

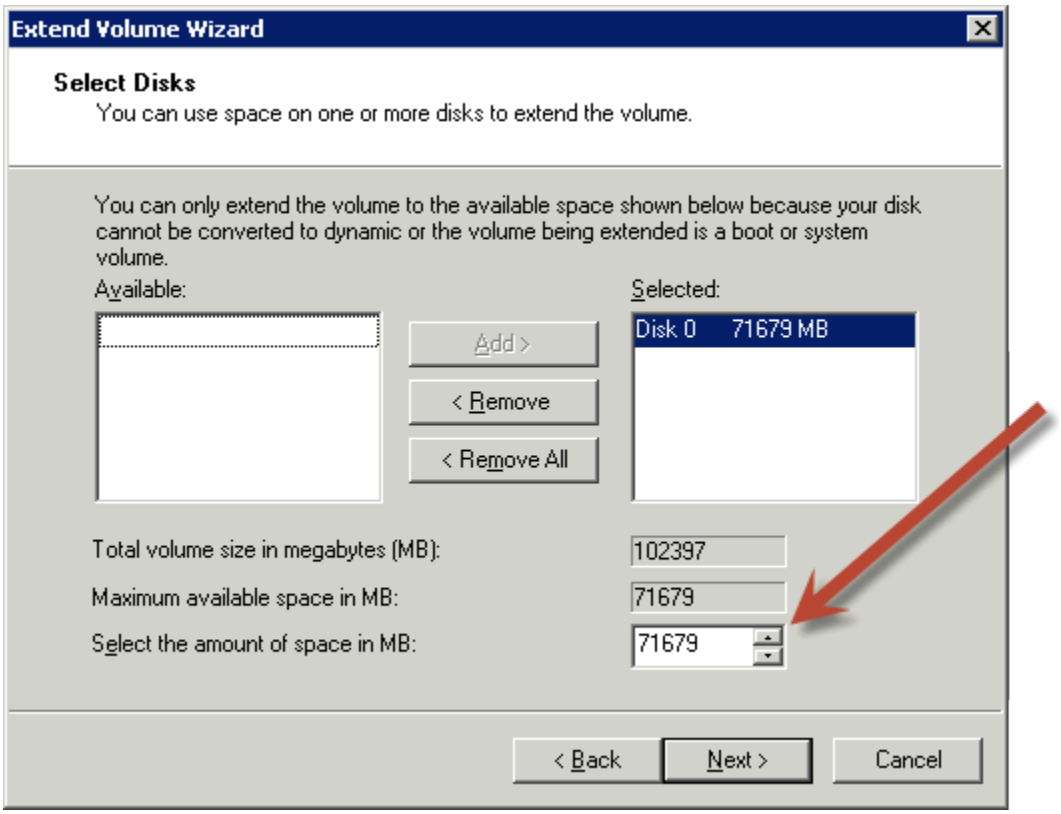


4. Di menu Manajemen Disk, pilih Tindakan, Pindai Ulang Disk.
5. Buka menu konteks (klik kanan) untuk drive yang diperluas dan pilih Perluas Volume.

> **Note**
>
> Perluas volume mungkin dinonaktifkan (berwarna abu-abu) jika:
>
> - Ruang yang tidak terisi tidak berdekatan dengan drive. Ruang yang tidak terisi harus berdekatan dengan sisi kanan drive yang ingin Anda perluas.
> - Volume menggunakan gaya partisi Master Boot Record (MBR) dan ukurannya sudah 2 TB. Volume yang menggunakan MBR tidak dapat melebihi 2 TB dalam ukuran.



6. Di wizard Perluas Volume, pilih Selanjutnya. Untuk Pilih jumlah ruang dalam MB, masukkan jumlah megabyte untuk memperluas volume. Umumnya, Anda menentukan ruang maksimum yang tersedia. Teks yang disorot di bawah Dipilih adalah jumlah ruang yang ditambahkan, bukan ukuran akhir yang akan dimiliki oleh volume. Selesaikan panduan.



7. Jika Anda menambah ukuran volume NVMe pada instans yang tidak memiliki driver NVMe AWS , Anda harus melakukan boot ulang instans agar Windows dapat melihat ukuran volume yang baru. Untuk informasi selengkapnya tentang menginstal driver AWS NVMe, lihat driver AWS NVMe untuk instance Windows.

PowerShell

Gunakan prosedur berikut untuk memperluas sistem file Windows menggunakan PowerShell.

Untuk memperluas sistem file menggunakan PowerShell

1. Sebelum memperluas sistem file yang berisi data berharga, praktik terbaiknya adalah membuat snapshot volume yang berisi data tersebut jika Anda perlu mengembalikan perubahan Anda. Untuk informasi selengkapnya, lihat Membuat snapshot Amazon EBS.
2. Masuk ke instans Windows menggunakan Remote Desktop.
3. Jalankan PowerShell sebagai administrator.
4. Jalankan `Get-Partition` perintah. PowerShell mengembalikan nomor partisi yang sesuai untuk setiap partisi, huruf drive, offset, ukuran, dan jenis. Perhatikan huruf drive dari partisi yang akan diperluas.
5. Jalankan perintah berikut untuk memindai ulang disk.

```
"rescan" | diskpart
```

6. Jalankan perintah berikut, menggunakan huruf drive yang Anda catat di langkah 4 sebagai pengganti**<drive-letter>**. PowerShell mengembalikan ukuran minimum dan maksimum partisi yang diizinkan, dalam byte.

```
Get-PartitionSupportedSize -DriveLetter <drive-letter>
```

7. Untuk memperpanjang partisi ke jumlah tertentu, jalankan perintah berikut, yang akan memasukkan ukuran baru volume di tempat **<size>**. Anda dapat memasukkan ukurannya `KB`, `MB`, dan `GB`; contohnya, `50GB`.

```
Resize-Partition -DriveLetter <drive-letter> -Size <size>
```

Untuk memperpanjang partisi ke ukuran maksimum yang tersedia, jalankan perintah berikut.

```
Resize-Partition -DriveLetter <drive-letter> -Size $(Get-PartitionSupportedSize
 -DriveLetter <drive-letter>).SizeMax
```

PowerShell Perintah berikut menunjukkan perintah lengkap dan aliran respons untuk memperluas sistem file ke ukuran tertentu.

```
PS C:\> Get-Partition


   DiskPath: \\?\scsi#disk&ven_nvme&prod_amazon_elastic_b#4&26a12046&0&000000#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}

PartitionNumber  DriveLetter Offset                                        Size Type
---------------  ----------- ------                                        ---- ----
1                C           1048576                                     30 GB IFS


   DiskPath: \\?\scsi#disk&ven_nvme&prod_amazon_elastic_b#4&3476342 3&0&000000#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}

PartitionNumber  DriveLetter Offset                                        Size Type
---------------  ----------- ------                                        ---- ----
1                D           1048576                                     8 MB IFS


PS C:\> "rescan" | diskpart

Microsoft DiskPart version 10.0.17763.1911

Copyright (C) Microsoft Corporation.
On computer:

DISKPART>
Please wait while DiskPart scans your configuration...

DiskPart has finished scanning your configuration.

DISKPART>
PS C:\> Get-PartitionSupportedSize -DriveLetter D

SizeMin      SizeMax
-------      -------
8388608 107372085248


PS C:\> Resize-Partition -DriveLetter D -Size 50GB
PS C:\> Get-Partition


   DiskPath: \\?\scsi#disk&ven_nvme&prod_amazon_elastic_b#4&26a12046&0&000000#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}

PartitionNumber  DriveLetter Offset                                        Size Type
---------------  ----------- ------                                        ---- ----
1                C           1048576                                     30 GB IFS


   DiskPath: \\?\scsi#disk&ven_nvme&prod_amazon_elastic_b#4&3476342 3&0&000000#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}

PartitionNumber  DriveLetter Offset                                        Size Type
---------------  ----------- ------                                        ---- ----
1                D           1048576                                     50 GB IFS
```

PowerShell Perintah berikut menunjukkan perintah lengkap dan aliran respons untuk memperluas sistem file ke ukuran maksimum yang tersedia.

```
PS C:\> Get-Partition


   DiskPath: \\?\scsi#disk&ven_nvme&prod_amazon_elastic_b#4&26a12046&0&000000#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}

PartitionNumber  DriveLetter Offset                                        Size Type
---------------  ----------- ------                                        ---- ----
1                C           1048576                                     30 GB IFS


   DiskPath: \\?\scsi#disk&ven_nvme&prod_amazon_elastic_b#4&3476423&0&000000#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}

PartitionNumber  DriveLetter Offset                                        Size Type
---------------  ----------- ------                                        ---- ----
1                D           1048576                                     50 GB IFS


PS C:\> "rescan" | diskpart

Microsoft DiskPart version 10.0.17763.1911

Copyright (C) Microsoft Corporation.
On computer:

DISKPART>
Please wait while DiskPart scans your configuration...

DiskPart has finished scanning your configuration.

DISKPART>
PS C:\> Get-PartitionSupportedSize -DriveLetter D

 SizeMin      SizeMax
 -------      -------
59047936 107372085248


PS C:\> Resize-Partition -DriveLetter D -Size $(Get-PartitionSupportedSize -DriveLetter D).SizeMax
PS C:\> Get-Partition


   DiskPath: \\?\scsi#disk&ven_nvme&prod_amazon_elastic_b#4&26a12046&0&000000#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}

PartitionNumber  DriveLetter Offset                                        Size Type
---------------  ----------- ------                                        ---- ----
1                C           1048576                                     30 GB IFS


   DiskPath: \\?\scsi#disk&ven_nvme&prod_amazon_elastic_b#4&3476423&0&000000#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}

PartitionNumber  DriveLetter Offset                                        Size Type
---------------  ----------- ------                                        ---- ----
1                D           1048576                                    100 GB IFS
```

Lepaskan volume Amazon EBS dari instans

Anda perlu melepaskan volume Amazon Elastic Block Store (Amazon EBS) dari sebuah instans sebelum Anda dapat melampirkannya ke instans yang berbeda atau menghapusnya. Menghapus volume tidak memengaruhi data pada volume.

Topik

- Pertimbangan
- Lepaskan dan lepaskan volume

- Pemecahan Masalah

Pertimbangan

- Anda dapat melepaskan volume Amazon EBS dari suatu instans secara eksplisit atau dengan mengakhiri instans tersebut. Namun, jika proses sedang berjalan, Anda harus melepas volume terlebih dahulu dari instans.
- Jika volume EBS adalah perangkat root suatu instans, Anda harus menghentikan instans tersebut sebelum Anda dapat melepaskan volume.
- Anda dapat melampirkan ulang volume yang Anda lepas (tanpa melepasnya), tetapi tidak akan mendapatkan titik pemasangan yang sama. Jika penulisan ke volume sedang berlangsung saat dilepas, data di volume mungkin tidak sinkron.
- Setelah Anda melepaskan volume, Anda masih dikenakan biaya untuk penyimpanan volume selama jumlah penyimpanan melebihi batas Tingkat AWS Gratis. Anda harus menghapus volume agar tidak dikenai biaya lebih lanjut. Untuk informasi selengkapnya, lihat Menghapus volume Amazon EBS.

Lepaskan dan lepaskan volume

Gunakan prosedur berikut untuk melepaskan volume dari suatu instans. Hal ini dapat berguna saat Anda perlu memasang volume ke instans yang berbeda atau saat Anda perlu menghapus volume.

Langkah-langkah

- Langkah 1: Melepaskan volume
- Langkah 2: Melepaskan volume dari instans
- Langkah 3: (Hanya instance Windows) Copot pemasangan lokasi perangkat offline

Langkah 1: Melepaskan volume

Instans Linux

Dari instans Linux Anda, gunakan perintah berikut untuk melepaskan perangkat `/dev/sdh`.

```
[ec2-user ~]$ sudo umount -d /dev/sdh
```

Instans Windows

Dari instans Windows Anda, lepaskan volume dengan cara berikut.

1. Mulai utilitas Manajemen Disk.
   - (Windows Server 2012 dan versi yang lebih tinggi) Pada bilah tugas, klik kanan logo Windows dan pilih Manajemen Disk.
   - Windows Server 2008) Pilih Mulai, Alat Administratif, Manajemen Komputer, Manajemen Disk.
2. Klik kanan disk (misalnya, klik kanan Disk 1) lalu pilih Offline. Tunggu status disk untuk diubah ke Offline sebelum membuka konsol Amazon EC2.

Langkah 2: Melepaskan volume dari instans

Untuk melepaskan volume dari instans, gunakan salah satu metode berikut:

Console

Untuk memisahkan volume EBS menggunakan konsol

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Volume.
3. Pilih volume yang akan dilepaskan dan pilih Tindakan, Lepaskan volume.
4. Ketika diminta untuk mengonfirmasi, pilih Lepaskan.

AWS CLI

Untuk melepaskan volume EBS dari sebuah instans menggunakan AWS CLI

Setelah melepaskan volume, gunakan perintah detach-volume.

Tools for Windows PowerShell

Untuk melepaskan volume EBS dari sebuah instans menggunakan Tools for Windows PowerShell

Setelah melepas volume, gunakan Dismount-EC2Volumeperintah.

Langkah 3: (Hanya instance Windows) Copot pemasangan lokasi perangkat offline

Ketika Anda melepaskan dan melepaskan lampiran volume dari suatu instans, Windows menandai lokasi perangkat sebagai offline. Lokasi perangkat tetap offline setelah boot ulang, dan berhenti serta memulai ulang instans. Ketika Anda memulai ulang instans, Windows mungkin memasang salah satu volume yang tersisa ke lokasi perangkat offline. Hal ini menyebabkan volume tidak tersedia di Windows. Untuk mencegah hal ini terjadi dan untuk memastikan bahwa semua volume dilampirkan ke lokasi perangkat online pada waktu Windows dimulai berikutnya, lakukan langkah-langkah berikut:

1. Pada instans, buka Manajer Perangkat.
2. Dalam Manajer Perangkat, pilih Lihat, Tampilkan perangkat tersembunyi.
3. Dalam daftar perangkat, perluas simpul Kontroler penyimpanan.

   Lokasi perangkat tempat volume yang dilepas lampirannya dipasang bernama `AWS NVMe Elastic Block Storage Adapter` dan akan tampak berwarna abu-abu.
4. Klik kanan setiap lokasi perangkat berwarna abu-abu bernama `AWS NVMe Elastic Block Storage Adapter`, pilih Hapus instalasi perangkat dan pilih Hapus instalasi

   > **Important**
   >
   > Jangan pilih kotak centang Hapus perangkat lunak driver untuk perangkat ini.

Pemecahan Masalah

Berikut ini adalah masalah umum yang dihadapi saat melepaskan lampiran volume, dan cara mengatasinya.

> **Note**
>
> Untuk menjaga kemungkinan kehilangan data, ambil snapshot volume Anda sebelum mencoba melepasnya. Pelepasan paksa dari volume yang macet dapat menyebabkan kerusakan pada sistem file atau data yang ada di dalamnya atau tidak dapat melampirkan volume baru menggunakan nama perangkat yang sama, kecuali jika instans di-boot ulang.

- Jika Anda mengalami masalah saat melepaskan volume melalui konsol Amazon EC2, Anda akan terbantu dengan menggunakan perintah CLI describe-volumes untuk mendiagnosis masalah. Untuk informasi selengkapnya, lihat describe-volume.
- Jika volume Anda tetap dalam status `detaching`, Anda dapat memaksa pelepasan dengan memilih Lepas Paksa. Gunakan opsi ini hanya sebagai upaya terakhir untuk memisahkan volume dari instans yang gagal, atau jika Anda melepaskan volume dengan tujuan menghapusnya. Instans yang ada tidak memiliki peluang untuk membersihkan cache sistem file atau metadata sistem file. Jika Anda menggunakan opsi ini, Anda harus melakukan prosedur pemeriksaan dan perbaikan sistem file.
- Jika Anda telah mencoba melepas paksa volume beberapa kali selama beberapa menit tetapi tetap berada di status `detaching`, Anda dapat mengirim permintaan bantuan ke AWS re:Post. Untuk membantu mempercepat resolusi, sertakan ID volume dan jelaskan langkah-langkah yang telah Anda ambil.
- Saat Anda mencoba melepaskan volume yang masih terpasang, volume dapat macet di status `busy` saat mencoba untuk melepaskannya. Output berikut dari describe-volumes menunjukkan contoh dari kondisi ini:

```
"Volumes": [
    {
        "AvailabilityZone": "us-west-2b",
        "Attachments": [
            {
                "AttachTime": "2016-07-21T23:44:52.000Z",
                "InstanceId": "i-fedc9876",
                "VolumeId": "vol-1234abcd",
                "State": "busy",
                "DeleteOnTermination": false,
                "Device": "/dev/sdf"
            }
        ...
    }
]
```

Saat Anda mengalami kondisi ini, pelepasan dapat ditunda tanpa batas waktu hingga Anda melepas volume, pelepasan paksa, boot ulang instans, atau ketiganya.

Menghapus volume Amazon EBS

Anda dapat menghapus volume Amazon EBS yang tidak lagi Anda butuhkan. Setelah penghapusan, datanya hilang dan volumenya tidak dapat dilampirkan ke instans apa pun. Jadi, sebelum dihapus Anda dapat menyimpan snapshot volume, yang dapat Anda gunakan untuk membuat ulang volume nantinya.

> **Note**
>
> Anda tidak dapat menghapus volume jika terlampir ke suatu instans. Untuk menghapus volume, Anda harus melepasnya terlebih dahulu. Untuk informasi selengkapnya, lihat Lepaskan volume Amazon EBS dari instans.
> Anda dapat memeriksa apakah volume diampirkan pada suatu instans. Di konsol, pada Volume Anda dapat melihat status volume Anda.
>
> - Jika suatu volume dilampirkan ke suatu instans, volume ada dalam status `in-use`.
> - Jika suatu volume dilepas dari suatu instans, volume ada dalam status `available`. Anda dapat menghapus volume ini.

Anda dapat menghapus volume EBS menggunakan salah satu metode berikut.

Console

Untuk menghapus volume EBS menggunakan konsol

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Volume.
3. Pilih volume yang akan dihapus dan pilih Tindakan, Hapus volume.

   > **Note**
   >
   > Jika Hapus Volume berwarna abu-abu, volume terlampir pada instans. Anda harus melepaskan volume dari instans sebelum dapat dihapus.

4. Di kotak dialog konfirmasi, pilih Hapus.

AWS CLI

Untuk menghapus volume EBS menggunakan AWS CLI

Gunakan perintah delete-volume.

Tools for Windows PowerShell

Untuk menghapus volume EBS menggunakan Alat untuk Windows PowerShell

Gunakan perintah Remove-EC2Volume.

Ganti volume Amazon EBS menggunakan snapshot sebelumnya

Snapshot Amazon EBS adalah alat bantu pencadangan pilihan di Amazon EC2 karena kecepatan, kenyamanan, dan biayanya. Saat membuat volume dari snapshot, Anda membuat ulang statusnya pada titik waktu tertentu dengan data yang disimpan hingga titik tertentu secara utuh. Dengan memasang volume yang dibuat dari snapshot ke suatu instans, Anda dapat menduplikasi data di seluruh Wilayah, membuat lingkungan pengujian, mengganti volume produksi yang rusak atau korup secara keseluruhan, atau mengambil file dan direktori spesifik dan mentransfernya ke volume lain yang terlampir. Untuk informasi selengkapnya, lihat Snapshot Amazon EBS.

Anda dapat menggunakan salah satu prosedur berikut untuk mengganti volume Amazon EBS dengan volume lain yang dibuat dari snapshot sebelumnya dari volume tersebut.

Console

Untuk mengganti volume menggunakan konsol

1. Buat volume dari snapshot dan tulis ID volume baru. Untuk informasi selengkapnya, lihat Membuat volume dari snapshot.

   > **Note**
   >
   > Pastikan Anda membuat volume di Zona Ketersediaan yang sama dengan instans. Volume hanya dapat dilampirkan pada instans di Zona Ketersediaan yang sama.

2. Pada halaman Instans, pilih instans untuk mengganti volume dan tuliskan ID instans.

   Dengan instans yang masih dipilih, pilih tab Penyimpanan. Di bagian Perangkat blok, cari volume yang akan diganti dan tuliskan nama perangkat untuk volume, misalnya `/dev/sda1`.

Pilih ID volume.

3. Pada layar Volume, pilih volume dan pilih Tindakan, Lepaskan volume, Lepaskan.
4. Pilih volume baru yang Anda buat pada langkah 1 dan pilih Tindakan, Pasang volume.

   Untuk Instans dan Nama perangkat, masukkan ID instans dan nama perangkat yang Anda tulis di Langkah 2, lalu pilih Pasang volume.
5. Sambungkan ke instans dan pasang volume. Untuk informasi selengkapnya, lihat Buat volume Amazon EBS tersedia untuk digunakan.

AWS CLI

Untuk mengganti volume menggunakan AWS CLI

1. Buat snapshot volume baru dari snapshot. Gunakan perintah create-volume. Untuk `--snapshot-id`, tentukan ID snapshot yang akan digunakan. Untuk `--availability-zone`, tentukan Zona Ketersediaan yang sama dengan instans. Konfigurasi parameter yang tersisa sesuai kebutuhan.

   > **Note**
   >
   > Pastikan Anda membuat volume di Zona Ketersediaan yang sama dengan instans. Volume hanya dapat dilampirkan pada instans di Zona Ketersediaan yang sama.

   ```
   $ aws ec2 create-volume \
   --volume-type volume_type \
   --size volume_size \
   --snapshot-id snapshot_id \
   --availability-zone az_id
   ```

   Pada output perintah, catat ID volume baru.
2. Dapatkan nama perangkat volume yang akan diganti. Gunakan perintah describe-instances. Untuk `--instance-ids`, tentukan ID instans tempat mengganti volume.

   ```
   $ aws ec2 describe-instances --instance-ids instance_id
   ```

Dalam `BlockDeviceMappings` di output perintah, catat `DeviceName` dan `VolumeId` untuk volume yang akan diganti.

3. Lepaskan volume yang akan diganti dari instans. Gunakan perintah detach-volume. Untuk `--volume-id`, tentukan ID volume yang akan dilepas.

```
$ aws ec2 detach-volume --volume-id volume_id
```

4. Lampirkan volume pengganti ke instans. Gunakan perintah attach-volume. Untuk `--volume-id`, tentukan ID volume pengganti. Untuk `--instance-id`, tentukan ID dari instans tempat melampirkan volume. Untuk `--device`, tentukan nama perangkat yang sama yang Anda catat sebelumnya.

```
$ aws ec2 attach-volume \
--volume-id volume_id \
--instance-id instance_id \
--device device_name
```

5. Sambungkan ke instans dan pasang volume. Untuk informasi selengkapnya, lihat Buat volume Amazon EBS tersedia untuk digunakan.

Pantau volume Amazon EBS Anda

AWS secara otomatis menyediakan data yang dapat Anda gunakan untuk memantau volume Amazon EBS Anda.

Daftar Isi

Untuk informasi pemantauan tambahan, lihat CloudWatch Metrik Amazon untuk Amazon EBS dan Amazon EventBridge untuk Amazon EBS.

Pemeriksaan status volume EBS

Pemeriksaan status volume memungkinkan Anda untuk lebih memahami, melacak, dan mengelola potensi ketidakkonsistenan data di volume Amazon EBS. Panduan ini dirancang untuk memberikan informasi yang Anda perlukan untuk menentukan apakah volume Amazon EBS terganggu, dan membantu Anda mengendalikan cara penanganan volume yang berpotensi tidak konsisten.

Pemeriksaan status volume adalah uji otomatis yang berjalan setiap 5 menit dan mengembalikan status lulus atau gagal. Jika semua pemeriksaan berhasil, status volume adalah `ok`. Jika pemeriksaan gagal, status volume adalah `impaired`. Jika statusnya `insufficient-data`, pemeriksaan mungkin masih berlangsung pada volume. Anda dapat melihat hasil pemeriksaan status volume untuk mengidentifikasi volume yang terganggu dan mengambil tindakan yang diperlukan.

Ketika Amazon EBS menentukan bahwa data volume berpotensi tidak konsisten, default-nya adalah data tersebut menonaktifkan I/O ke volume dari setiap instans EC2 yang terlampir, yang membantu mencegah kerusakan data. Setelah I/O dinonaktifkan, pemeriksaan status volume berikutnya gagal, dan status volume `impaired`. Selain itu, Anda akan melihat peristiwa yang memberi tahu Anda bahwa I/O dinonaktifkan, dan bahwa Anda dapat menyelesaikan status volume yang terganggu dengan mengaktifkan I/O ke volume. Kami menunggu sampai Anda mengaktifkan I/O untuk memberi Anda kesempatan untuk memutuskan apakah akan terus membiarkan instance Anda menggunakan volume, atau menjalankan pemeriksaan konsistensi menggunakan perintah, seperti fsck (instance Linux) atau chkdsk (instance Windows), sebelum melakukannya.

> **Note**
>
> Status volume didasarkan pada pemeriksaan status volume, dan tidak mencerminkan status volume. Oleh karena itu, status volume tidak menunjukkan volume dalam `error` menyatakan (misalnya, jika volume tidak dapat menerima I/O.) Untuk informasi tentang status volume, lihat Status volume.

Jika konsistensi volume tertentu tidak menjadi masalah, dan Anda lebih suka bahwa volume disediakan segera jika terganggu, Anda dapat mengganti perilaku default dengan mengonfigurasi volume untuk mengaktifkan I/O secara otomatis. Jika Anda mengaktifkan atribut volume IO Aktif Otomatis (`autoEnableIO` dalam API), pemeriksaan status volume terus berlanjut. Selain itu, Anda akan melihat sebuah peristiwa yang memberi tahu Anda bahwa volume ditentukan berpotensi tidak konsisten, tetapi I/O secara otomatis diaktifkan. Ini memungkinkan Anda memeriksa konsistensi volume atau menggantinya di lain waktu.

Pemeriksaan status performa I/O membandingkan performa volume aktual dengan performa yang diharapkan dari suatu volume. Hal ini akan memperingatkan Anda jika volume beperforma di bawah harapan. Pemeriksaan status ini hanya tersedia untuk SSD IOPS yang Tersedia (`io1` dan `io2`) dan volume SSD Tujuan Umum (`gp3`) yang terlampir pada suatu instans. Pemeriksaan status tidak valid untuk SSD Tujuan Umum (`gp2`), HDD Throughput yang Dioptimalkan (`st1`), HDD Cold (`sc1`), atau volume Magnetik (`standard`). Pemeriksaan status kinerja I/O dilakukan setiap menit sekali, dan CloudWatch mengumpulkan data ini setiap 5 menit. Mungkin diperlukan waktu hingga 5 menit dari saat Anda melampirkan volume `io1` atau `io2` ke instans untuk pemeriksaan status guna melaporkan status performa I/O.

> **Important**
>
> Saat menginisialisasi volume SSD IOPS yang Tersedia yang dipulihkan dari snapshot, performa volume dapat turun di bawah 50 persen dari tingkat yang diharapkan, yang menyebabkan volume menampilkan status `warning` dalam pemeriksaan status Performa I/O. Hal ini wajar, dan Anda dapat mengabaikan status `warning` pada volume SSD IOPS yang Tersedia saat Anda menginisialisasinya. Untuk informasi selengkapnya, lihat Inisialisasi volume Amazon EBS.

Tabel berikut mencantumkan status untuk volume Amazon EBS.

Status volume	Status yang diaktifkan I/O	Status performa I/O (hanya volume **`io1`**, **`io2`**, dan **`gp3`**)
`ok`	Diaktifkan (I/O Diaktifkan atau I/O Diaktifkan Otomatis)	Normal (Performa volume sesuai dengan yang diharapkan)
`warning`	Diaktifkan (I/O Diaktifkan atau I/O Diaktifkan Otomatis)	Terdegradasi (Performa volume di bawah ekspektasi) Sangat Menurun (Performa volume jauh di bawah harapan)
`impaired`	Diaktifkan (I/O Diaktifkan atau I/O Diaktifkan Otomatis)	Terhenti (Performa volume sangat terpengaruh)

Status volume	Status yang diaktifkan I/O	Status performa I/O (hanya volume **io1**, **io2**, dan **gp3**)
	Dinonaktifkan (Volume sedang offline dan menunggu pemulihan, atau menunggu pengguna mengaktifkan I/O)	Tidak Tersedia (Tidak dapat menentukan performa I/O karena I/O dinonaktifkan)
`insufficient-data`	Diaktifkan (I/O Diaktifkan atau I/O Diaktifkan Otomatis) Data Tidak Cukup	Data Tidak Cukup

Anda dapat melihat dan bekerja dengan pemeriksaan status menggunakan metode berikut.

Console

Untuk melihat pemeriksaan status

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Volume.

   Kolom Status Volume menampilkan status operasional setiap volume.
3. Untuk melihat detail status instans tertentu, pilih instans di kisi lalu pilih tab Pemeriksaan status.
4. Jika Anda memiliki volume dengan pemeriksaan status gagal (statusnya adalah `impaired`), lihat Bekerja dengan volume yang terganggu.

Atau, Anda dapat memilih Peristiwa di navigator guna melihat semua peristiwa untuk instans dan volume Anda. Untuk informasi selengkapnya, lihat Peristiwa volume EBS.

AWS CLI

Untuk melihat informasi status volume

Gunakan perintah describe-volume-status.

Untuk informasi selengkapnya tentang antarmuka baris perintah ini, lihat Mengakses Amazon EC2.

Tools for Windows PowerShell

Untuk melihat informasi status volume

Gunakan perintah Get-EC2 VolumeStatus.

Untuk informasi selengkapnya tentang antarmuka baris perintah ini, lihat Mengakses Amazon EC2.

Peristiwa volume EBS

Ketika Amazon EBS menentukan bahwa data volume berpotensi tidak konsisten, maka secara default menonaktifkan I/O ke volume dari semua instans EC2 terpasang. Hal ini menyebabkan pemeriksaan status volume gagal, dan membuat peristiwa status volume yang menunjukkan penyebab kegagalan.

Untuk mengaktifkan secara otomatis I/O pada volume dengan potensi ketidakkonsistenan data, ubah pengaturan atribut volume IO Aktif Otomatis (`autoEnableIO` di API). Untuk informasi selengkapnya tentang perubahan atribut ini, lihat Bekerja dengan volume yang terganggu.

Setiap peristiwa mencakup waktu mulai yang menunjukkan waktu terjadinya peristiwa, dan durasi yang menunjukkan berapa lama I/O untuk volume dinonaktifkan. Waktu selesai ditambahkan ke peristiwa saat I/O untuk volume diaktifkan.

Peristiwa status volume mencakup salah satu deskripsi berikut:

`Awaiting Action: Enable IO`

Data volume berpotensi tidak konsisten. I/O dinonaktifkan untuk volume hingga Anda mengaktifkannya secara eksplisit. Deskripsi peristiwa berubah menjadi IO Enabled setelah Anda secara eksplisit mengaktifkan I/O.

`IO Enabled`

Operasi I/O secara eksplisit diaktifkan untuk volume ini.

`IO Auto-Enabled`

Operasi I/O secara otomatis diaktifkan pada volume ini setelah peristiwa terjadi. Kami menyarankan Anda memeriksa inkonsistensi data sebelum melanjutkan penggunaan data.

`Normal`

Untuk volume `io1`, `io2`, dan `gp3` saja. Performa volume sesuai dengan yang diharapkan.

`Degraded`

Untuk volume `io1`, `io2`, dan `gp3` saja. Performa volume di bawah harapan.

`Severely Degraded`

Untuk volume `io1`, `io2`, dan `gp3` saja. Performa volume jauh di bawah harapan.

`Stalled`

Untuk volume `io1`, `io2`, dan `gp3` saja. Performa volume sangat terpengaruh.

Anda dapat melihat peristiwa untuk volume Anda menggunakan metode berikut.

Console

Untuk melihat peristiwa volume Anda

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Peristiwa. Semua instans dan volume yang memiliki peristiwa dicantumkan.
3. Anda dapat memfilter berdasarkan volume untuk melihat status volume saja. Anda juga dapat memfilter jenis status tertentu.
4. Pilih volume untuk menampilkan peristiwa spesifik.

AWS CLI

Untuk melihat peristiwa volume Anda

Gunakan perintah describe-volume-status.

Untuk informasi selengkapnya tentang antarmuka baris perintah ini, lihat Mengakses Amazon EC2.

Tools for Windows PowerShell

Untuk melihat peristiwa volume Anda

Gunakan perintah Get-EC2 VolumeStatus.

Untuk informasi selengkapnya tentang antarmuka baris perintah ini, lihat Mengakses Amazon EC2.

Jika Anda memiliki volume dengan I/O dinonaktifkan, lihat Bekerja dengan volume yang terganggu. Jika Anda memiliki volume dengan performa I/O berada di bawah normal, hal ini dapat menjadi kondisi sementara karena tindakan yang telah Anda ambil (misalnya, membuat snapshot volume selama penggunaan puncak, menjalankan volume pada instans yang tidak dapat mendukung bandwidth I/O yang diperlukan, mengakses data pada volume untuk pertama kali, dll.).

Bekerja dengan volume yang terganggu

Gunakan opsi berikut jika volume terganggu karena data volume berpotensi tidak konsisten.

Opsi

- Opsi 1: Melakukan pemeriksaan konsistensi pada volume yang terlampir pada instans
- Opsi 2: Melakukan pemeriksaan konsistensi pada volume menggunakan instans lain
- Opsi 3: Hapus volume jika Anda tidak lagi membutuhkannya

Opsi 1: Melakukan pemeriksaan konsistensi pada volume yang terlampir pada instans

Opsi paling sederhana adalah untuk mengaktifkan I/O, kemudian melakukan pemeriksaan konsistensi data pada volume sementara volume masih terlampir pada instans Amazon EC2.

Untuk melakukan pemeriksaan konsistensi pada volume yang terpasang

1. Hentikan aplikasi apa pun dari menggunakan volume.
2. Aktifkan I/O pada volume. Gunakan salah satu metode berikut.

   Console

   1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
   2. Di panel navigasi, pilih Peristiwa.
   3. Pilih volume yang memungkinkan operasi I/O.
   4. Pilih Tindakan, Aktifkan I/O.

   AWS CLI

   Untuk mengaktifkan I/O untuk volume dengan AWS CLI

   Gunakan perintah enable-volume-io.

Tools for Windows PowerShell

Untuk mengaktifkan I/O untuk volume dengan Alat untuk Windows PowerShell

Gunakan perintah Enable-EC2VolumeIO.

3. Periksa data di volume.

   a. Jalankan perintah fsck (instance Linux) atau chkdsk (instance Windows).

   b. (Opsional) Tinjau semua log aplikasi atau sistem yang tersedia untuk pesan kesalahan yang relevan.

   c. Jika volume telah terganggu selama lebih dari 20 menit, Anda dapat menghubungi AWS Support Center. Pilih Pemecahan Masalah, kemudian di kotak dialog Menyelesaikan Masalah Pemeriksaan Status, pilih Hubungi Dukungan untuk mengirimkan kasus dukungan.

Opsi 2: Melakukan pemeriksaan konsistensi pada volume menggunakan instans lain

Gunakan prosedur berikut untuk memeriksa volume di luar lingkungan produksi Anda.

> **Important**
>
> Prosedur ini dapat menyebabkan hilangnya I/O tulis yang ditangguhkan ketika volume I/O dinonaktifkan.

Untuk melakukan pemeriksaan konsistensi pada volume secara terpisah

1. Hentikan aplikasi apa pun dari menggunakan volume.
2. Lepaskan volume dari instans Untuk informasi selengkapnya, lihat Lepaskan volume Amazon EBS dari instans.
3. Aktifkan I/O pada volume. Gunakan salah satu metode berikut.

   Console

   1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
   2. Di panel navigasi, pilih Peristiwa.
   3. Pilih volume yang Anda lepas di langkah sebelumnya.
   4. Pilih Tindakan, Aktifkan I/O.

AWS CLI

Untuk mengaktifkan I/O untuk volume dengan AWS CLI

Gunakan perintah enable-volume-io.

Tools for Windows PowerShell

Untuk mengaktifkan I/O untuk volume dengan Alat untuk Windows PowerShell

Gunakan perintah Enable-EC2VolumeIO.

4. Lampirkan volume ke instans lainnya. Untuk informasi selengkapnya, lihat Meluncurkan instans Anda danLampirkan volume Amazon EBS ke instans.
5. Periksa data di volume.

   a. Jalankan perintah fsck (instance Linux) atau chkdsk (instance Windows).
   b. (Opsional) Tinjau semua log aplikasi atau sistem yang tersedia untuk pesan kesalahan yang relevan.
   c. Jika volume telah terganggu selama lebih dari 20 menit, Anda dapat menghubungi AWS Support Center. Pilih Pemecahan Masalah, lalu di kotak dialog pemecahan masalah, pilih Hubungi Dukungan untuk mengirimkan kasus dukungan.

Opsi 3: Hapus volume jika Anda tidak lagi membutuhkannya

Jika Anda ingin menghapus volume dari lingkungan Anda, cukup hapus volume. Untuk informasi tentang menghapus volume, lihat Menghapus volume Amazon EBS.

Jika Anda memiliki snapshot baru yang mencadangkan data pada volume, Anda dapat membuat volume baru dari snapshot. Untuk informasi selengkapnya, lihat Membuat volume dari snapshot.

Bekerja dengan atribut volume IO yang Diaktifkan Otomatis

Ketika Amazon EBS menentukan bahwa data volume berpotensi tidak konsisten, maka secara default menonaktifkan I/O ke volume dari semua instans EC2 terpasang Hal ini menyebabkan pemeriksaan status volume gagal, dan membuat peristiwa status volume yang menunjukkan penyebab kegagalan. Jika konsistensi volume tertentu tidak menjadi masalah, dan Anda lebih memilih agar volume tersebut tersedia segera jika terganggu, Anda dapat mengganti perilaku default dengan mengonfigurasi volume untuk mengaktifkan I/O secara otomatis. Jika Anda mengaktifkan

atribut volume IO Aktif Otomatis (`autoEnableIO` di API), I/O antara volume dan instans secara otomatis diaktifkan ulang dan pemeriksaan status volume akan terlewati. Selain itu, Anda akan melihat peristiwa yang memberi tahu Anda bahwa volume berada dalam status yang berpotensi tidak konsisten, tetapi I/O secara otomatis diaktifkan. Jika peristiwa ini terjadi, Anda harus memeriksa konsistensi volume dan menggantinya jika perlu. Untuk informasi selengkapnya, lihat Peristiwa volume EBS.

Anda dapat melihat dan memodifikasi atribut IO Aktif Otomatis volume menggunakan salah satu metode berikut.

Amazon EC2 console

Untuk melihat atribut IO Diaktifkan Otomatis dari volume

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Volume.
3. Pilih volume dan pilih tab Pemeriksaan status.

   Bidang I/O aktif otomatis menampilkan pengaturan saat ini (Diaktifkan atau Dinonaktifkan) untuk volume yang dipilih.

Untuk memodifikasi atribut IO yang Diaktifkan Otomatis dari volume

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Volume.
3. Pilih volume dan pilih Tindakan, Kelola I/O yang diaktifkan otomatis.
4. Untuk mengaktifkan I/O secara otomatis untuk volume yang terganggu, pilih kotak centang Aktifkan otomatis I/O untuk volume yang terganggu. Untuk menonaktifkan fitur, kosongkan kotak centang.
5. Pilih Perbarui.

AWS CLI

Untuk melihat atribut autoEnableIO dari volume

Gunakan perintah describe-volume-attribute.

Untuk mengubah atribut autoEnableIO dari volume

Gunakan perintah modify-volume-attribute.

Untuk informasi selengkapnya tentang antarmuka baris perintah ini, lihat Mengakses Amazon EC2

Tools for Windows PowerShell

Untuk melihat atribut autoEnableIO dari volume

Gunakan perintah Get-EC2 VolumeAttribute.

Untuk mengubah atribut autoEnableIO dari volume

Gunakan perintah Edit-EC2 VolumeAttribute.

Untuk informasi selengkapnya tentang antarmuka baris perintah ini, lihat Mengakses Amazon EC2

Pengujian kesalahan pada Amazon EBS

Gunakan AWS Fault Injection Service dan tindakan Jeda I/O untuk menghentikan sementara I/O antara volume Amazon EBS dan instance yang dilampirkan untuk menguji cara beban kerja Anda menangani interupsi I/O. Dengan AWS FIS, Anda dapat menggunakan eksperimen terkontrol untuk menguji arsitektur dan pemantauan, seperti CloudWatch alarm Amazon dan konfigurasi batas waktu OS, serta meningkatkan ketahanan terhadap kesalahan penyimpanan.

Untuk informasi selengkapnya AWS FIS, lihat Panduan AWS Fault Injection Service Pengguna.

Pertimbangan

Perhatikan pertimbangan berikut untuk menjeda volume I/O:

- Anda dapat menjeda I/O untuk semua jenis volume Amazon EBS yang dilampirkan ke instans yang dibangun di Sistem Nitro.
- Anda dapat menjeda I/O untuk volume root.
- Anda dapat menjeda I/O untuk volume Multi-Lampiran yang diaktifkan. Jika Anda menjeda I/O untuk volume yang mengaktifkan Multi-Lampiran, I/O dijeda antara volume dan semua instans yang dilampirkan.
- Untuk menguji konfigurasi batas waktu OS Anda, tetapkan durasi percobaan sama dengan atau lebih besar dari nilai yang ditentukan untuk `nvme_core.io_timeout`. Untuk informasi selengkapnya, lihat Waktu habis operasi I/O.

- Jika Anda mendorong I/O ke volume dengan I/O dijeda, hal berikut akan terjadi:
  - Transisi status volume ke `impaired` dalam 120 detik. Untuk informasi selengkapnya, lihat Pantau volume Amazon EBS Anda.
  - CloudWatch Metrik untuk panjang antrian (`VolumeQueueLength`) akan menjadi bukan nol. Alarm atau pemantauan apa pun harus memantau kedalaman antrean non-nol. Untuk informasi selengkapnya, lihat Metrik untuk volume Amazon EBS.
  - CloudWatch Metrik untuk `VolumeReadOps` atau `VolumeWriteOps` akan menjadi`0`, yang menunjukkan bahwa volume tidak lagi memproses I/O.

Batasan

Perhatikan pertimbangan berikut untuk menjeda volume I/O:

- Volume penyimpanan instans tidak didukung.
- Tipe instans berbasis Xen tidak didukung.
- Anda tidak dapat menjeda I/O untuk volume yang dibuat di Outpost di AWS Outposts, di AWS Wavelength Zona, atau di Zona Lokal.

Anda dapat melakukan eksperimen dasar dari konsol Amazon EC2, atau Anda dapat melakukan eksperimen lanjutan menggunakan konsol. AWS FIS Untuk informasi selengkapnya tentang melakukan eksperimen lanjutan menggunakan AWS FIS konsol, lihat Tutorial untuk AWS FIS di Panduan AWS Fault Injection Service Pengguna.

Untuk melakukan percobaan dasar menggunakan konsol Amazon EC2

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Volume.
3. Pilih volume untuk menjeda I/O dan pilih Tindakan, Injeksi kesalahan, Jeda volume I/O.
4. Untuk Durasi, masukkan durasi untuk menjeda I/O antara volume dan instans. Bidang di sebelah daftar dropdown Durasi menunjukkan durasi dalam format ISO 8601.
5. Di bagian Akses layanan, pilih peran layanan IAM untuk berasumsi AWS FIS untuk melakukan eksperimen. Anda dapat menggunakan peran default, atau peran yang sudah ada yang Anda buat. Untuk informasi selengkapnya, lihat Membuat peran IAM untuk AWS FIS eksperimen.
6. Pilih Jeda volume I/O. Saat diminta, masukkan `start` di bidang konfirmasi dan pilih Mulai percobaan.

7. Pantau kemajuan dan dampak percobaan Anda. Untuk informasi selengkapnya, lihat Memantau AWS FIS di Panduan Pengguna AWS FIS .

Snapshot Amazon EBS

Anda dapat mencadangkan data pada volume Amazon EBS Anda dengan membuat point-in-time salinan, yang dikenal sebagai snapshot Amazon EBS. Snapshot adalah cadangan tambahan, yang berarti bahwa kami hanya menyimpan blok pada perangkat yang telah berubah sejak snapshot terbaru Anda. Hal ini meminimalkan waktu yang diperlukan untuk membuat snapshot dan menghemat biaya penyimpanan dengan tidak menduplikasi data.

> **Important**
>
> AWS tidak secara otomatis mencadangkan data yang disimpan pada volume EBS Anda. Untuk ketahanan data dan pemulihan bencana, Anda bertanggung jawab untuk membuat snapshot EBS secara teratur, atau menyiapkan pembuatan snapshot otomatis dengan menggunakan Amazon Data Lifecycle Manager atau AWS Backup.

Snapshot EBS disimpan di Amazon S3, di bucket S3 yang tidak dapat Anda akses secara langsung. Anda dapat membuat dan mengelola snapshot Anda menggunakan konsol Amazon EC2 atau API Amazon EC2. Anda tidak dapat mengakses snapshot menggunakan konsol Amazon S3 atau API Amazon S3.

Setiap snapshot berisi semua informasi yang diperlukan untuk memulihkan data Anda (dari saat ketika snapshot diambil) ke volume EBS baru. Saat Anda membuat volume EBS berdasarkan snapshot, volume baru dimulai sebagai replika persis dari volume yang digunakan untuk membuat snapshot. Volume yang direplikasi memuat data di latar belakang sehingga Anda dapat segera menggunakannya. Jika Anda mengakses data yang belum dimuat, volume akan langsung mengunduh data yang diminta dari Amazon S3, lalu melanjutkan memuat sisa data volume di latar belakang. Untuk informasi selengkapnya, lihat Membuat snapshot Amazon EBS. Saat Anda menghapus snapshot, hanya data yang unik dari snapshot tersebut yang dihapus. Untuk informasi selengkapnya, lihat Hapus snapshot Amazon EBS.

Untuk informasi selengkapnya, lihat halaman produk Amazon EBS Snapshots.

Peristiwa snapshot

Anda dapat melacak status snapshot EBS Anda melalui CloudWatch Acara. Untuk informasi selengkapnya, lihat Peristiwa snapshot EBS.

Snapshot yang konsisten dengan aplikasi (hanya instance Windows)

Anda dapat menggunakan Run Command Systems Manager untuk mengambil snapshot yang konsisten aplikasi dari semua volume EBS yang terlampir ke instans Amazon EC2 Windows Anda. Proses snapshot menggunakan Volume Shadow Copy Service (VSS) Windows untuk mengambil cadangan tingkat citra aplikasi yang sadar VSS, termasuk data dari transaksi yang tertunda antara aplikasi ini dan disk. Anda tidak perlu mematikan instans atau memutusnya saat Anda mencadangkan semua volume yang terlampir. Untuk informasi selengkapnya, lihat Membuat Snapshot yang Konsisten Aplikasi VSS.

Snapshot multivolume

Snapshots dapat digunakan untuk membuat cadangan beban kerja sangat penting, seperti basis data besar atau sistem file yang mencakup banyak volume EBS. Snapshot multi-volume memungkinkan Anda mengambil snapshot yang akurat point-in-time, terkoordinasi dengan data, dan konsisten crash di beberapa volume EBS yang dilampirkan ke instans EC2. Anda tidak perlu lagi menghentikan instans atau mengoordinasikan antara volume untuk memastikannya crash-consistent, karena snapshot secara otomatis diambil di banyak volume EBS. Untuk informasi selengkapnya, lihat langkah-langkah untuk membuat snapshot EBS multivolume di Membuat snapshot Amazon EBS .

Harga snapshot

Biaya untuk snapshot Anda didasarkan pada jumlah data yang disimpan. Karena snapshot bersifat inkremental, menghapus snapshot mungkin tidak mengurangi biaya penyimpanan data Anda. Data yang direferensikan secara eksklusif oleh snapshot dihapus saat snapshot dihapus, tetapi data yang dirujuk oleh snapshot lain disimpan. Untuk informasi selengkapnya, lihat Snapshot dan volume Amazon Elastic Block Store di Panduan Pengguna AWS Billing .

Daftar Isi

- Recycle Bin untuk snapshot
- Snapshot lokal Amazon EBS di Outposts

Cara kerja snapshot

Snapshot pertama yang Anda buat dari volume selalu merupakan snapshot penuh. Snapshot ini mencakup semua blok data yang ditulis ke volume pada saat membuat snapshot. Snapshot berikutnya dengan volume yang sama adalah snapshot inkremental. Snapshot ini hanya menyertakan blok data yang diubah dan baru yang ditulis ke volume sejak snapshot terakhir dibuat

Ukuran snapshot penuh ditentukan oleh ukuran data yang dicadangkan, bukan ukuran volume sumber. Demikian pula, biaya penyimpanan yang terkait dengan snapshot penuh ditentukan oleh ukuran snapshot, bukan ukuran volume sumber. Misalnya, Anda membuat snapshot pertama dari volume `200 GiB` Amazon EBS yang hanya berisi `50 GiB` data. Hal ini menghasilkan snapshot lengkap yang berukuran `50 GiB`, dan Anda ditagih untuk penyimpanan snapshot `50 GiB`.

Demikian pula, ukuran dan biaya penyimpanan snapshot tambahan ditentukan oleh ukuran data apa pun yang ditulis ke volume sejak snapshot sebelumnya dibuat. Melanjutkan contoh ini, jika Anda membuat snapshot kedua volume `200 GiB` setelah mengubah `20 GiB` data dan menambahkan `10 GiB` data, snapshot inkremental berukuran `30 GiB`. Anda kemudian ditagih untuk penyimpanan snapshot `30 GiB` tambahan itu.

Untuk informasi selengkapnya tentang harga snapshot, lihat Harga Amazon EBS.

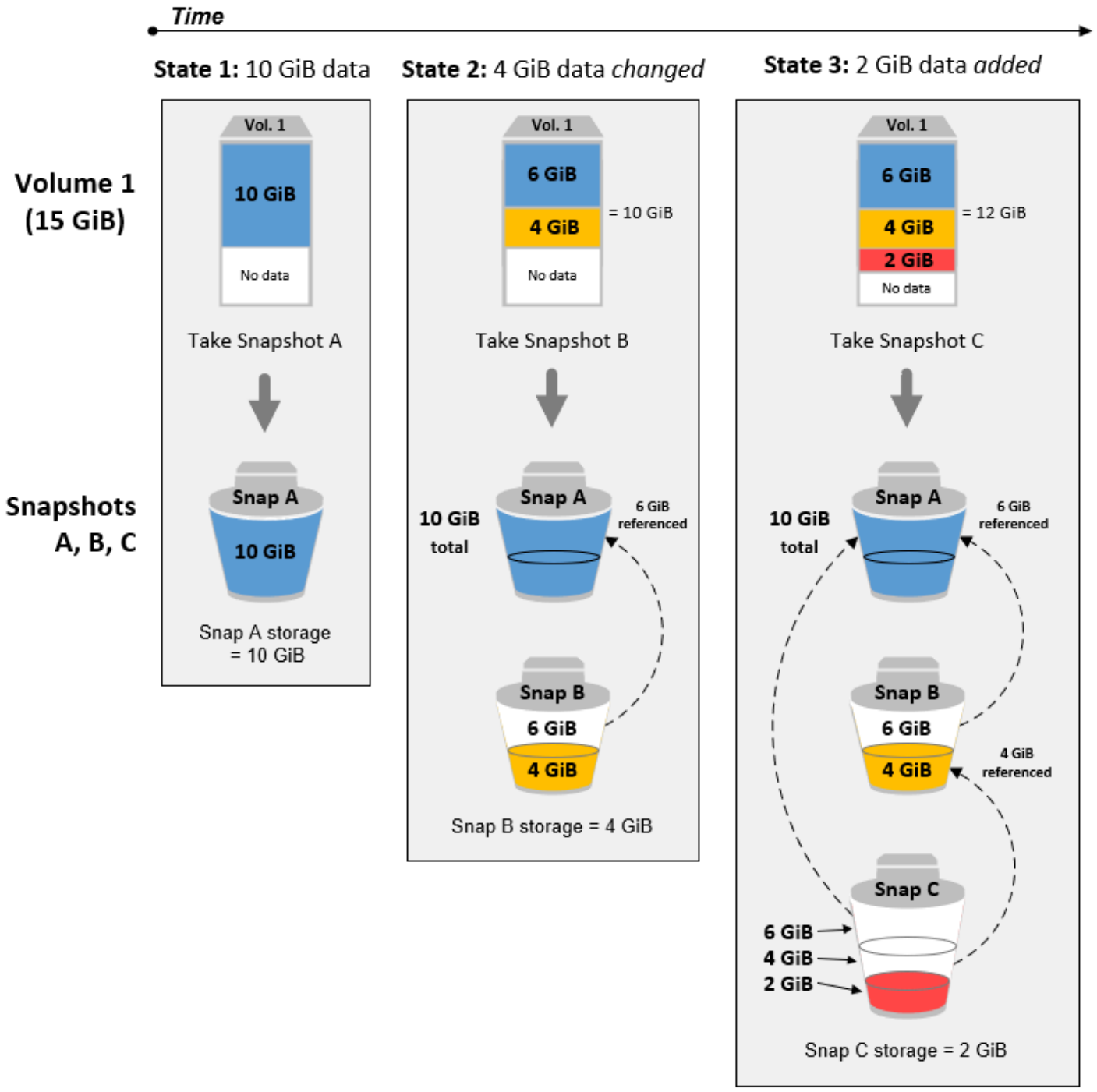
> **Important**
>
> Saat Anda mengarsipkan snapshot inkremental, snapshot dikonversi ke snapshot penuh yang mencakup semua blok yang ditulis ke volume pada saat snapshot dibuat. Kemudian, snapshot dipindahkan ke tingkat Arsip Snapshot Amazon EBS. Snapshot di tingkat arsip ditagih dengan tarif yang berbeda dari snapshot di tingkat standar. Untuk informasi selengkapnya, lihat Harga dan penagihan.

Bagian berikut menunjukkan cara snapshot EBS menangkap status volume pada satu titik waktu, dan cara snapshot berikutnya dari volume yang berubah membuat riwayat perubahan tersebut.

Banyak snapshot dengan volume yang sama

Diagram di bagian bawah ini menampilkan Volume 1, yang berukuran 15 GiB, pada tiga titik waktu. Snapshot diambil dari setiap tiga status volume ini. Diagram secara khusus menunjukkan hal berikut:

- Dalam Status 1, volume memiliki 10 GiB data. Snap A adalah snapshot pertama yang diambil dari volume. Snap A adalah snapshot lengkap dan seluruh 10 GiB data dicadangkan.
- Di Status 2, volume masih berisi 10 GiB data, tetapi hanya 4 GiB yang berubah setelah Snap A diambil. Snap B adalah snapshot inkremental. Snapshot perlu mencadangkan 4 GiB yang berubah saja. Data lain 6 GiB yang tidak berubah, yang sudah dicadangkan di Snap A, direferensikan oleh Snap B alih-alih dicadangkan lagi. Hal ini ditunjukkan dengan panah putus-putus.
- Di Status 3, 2 GiB data telah ditambahkan ke volume, untuk total 12 GiB, setelah Snap B diambil. Snap B adalah snapshot inkremental. Snapshot perlu mencadangkan hanya 2 GiB yang ditambahkan setelah Snap B diambil. Seperti yang ditunjukkan oleh panah putus-putus, Snap C juga mereferensikan 4 GiB data yang disimpan di Snap B, dan 6 GiB data yang disimpan di Snap A.
- Total penyimpanan yang diperlukan untuk tiga snapshot adalah 16 GiB total. Masing-masing menyumbang 10 GiB untuk Snap A, 4 GiB untuk Snap B, dan 2 GiB untuk Snap C.



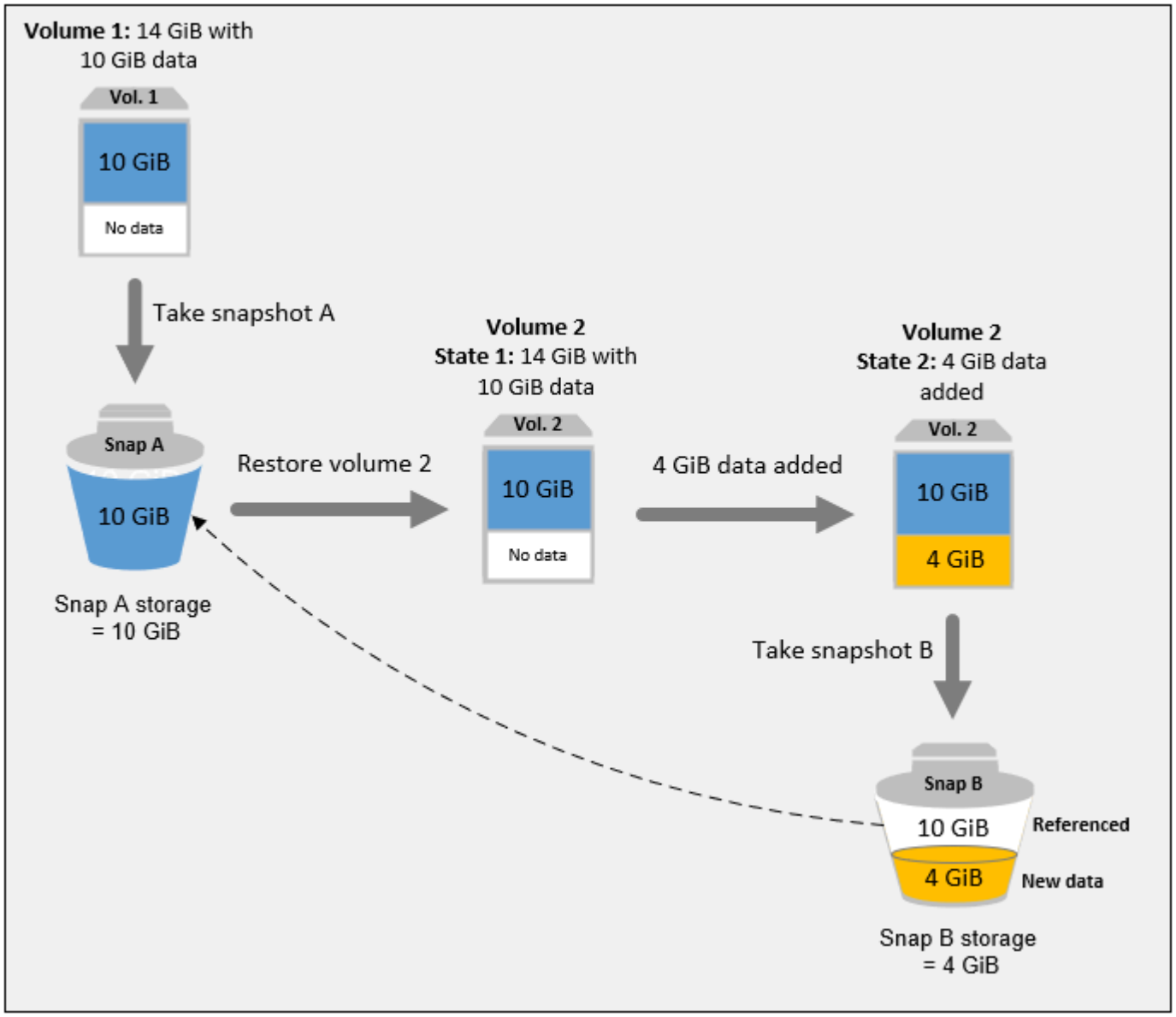
Snapshot inkremental dari volume berbeda

Diagram di bagian ini menunjukkan cara snapshot inkremental dapat diambil dari volume yang berbeda.

1. Vol 1, yang berukuran `14 GiB`, memiliki `10 GiB` data. Karena Snap A adalah snapshot pertama yang diambil dari volume, snapshot ini adalah snapshot penuh dan keseluruhan `10 GiB` data dicadangkan.
2. Vol 2 dibuat dari Snap A, sehingga ini adalah replika persis dari Vol 1 pada saat snapshot diambil.
3. Seiring waktu, `4 GiB` data ditambahkan ke Vol 2 dan ukuran total data adalah `14 GiB`.
4. Snap B diambil dari Vol 2. Untuk Snap B, hanya `4 GiB` data yang ditambahkan setelah volume dibuat dari Snap A dicadangkan. `10 GiB` data lain yang tidak berubah, yang sudah disimpan di Snap A, direferensikan oleh Snap B alih-alih dicadangkan lagi.

   Snap B adalah sebuah snapshot inkremental dari Snap A, meskipun dibuat dari volume yang berbeda.

> **Important**
>
> Diagram mengasumsikan bahwa Anda memiliki Vol 1 dan Snap A, dan bahwa Vol 2 dienkripsi dengan kunci KMS yang sama dengan Vol 1. Jika Vol 1 dimiliki oleh AWS akun lain dan akun itu mengambil Snap A dan membagikannya kepada Anda, maka Snap B akan menjadi snapshot lengkap. Atau, jika Vol 2 dienkripsi dengan kunci KMS yang berbeda dari Vol 1, Snap B akan menjadi snapshot penuh.



Untuk informasi selengkapnya tentang cara data dikelola saat Anda menghapus snapshot, lihat Hapus snapshot Amazon EBS.

Salin dan bagikan snapshot

Anda dapat membagikan snapshot di seluruh AWS akun dengan memodifikasi izin aksesnya. Anda dapat membuat salinan snapshot Anda sendiri serta snapshot yang telah dibagikan dengan Anda. Untuk informasi selengkapnya, lihat Membagikan snapshot Amazon EBS.

Sebuah snapshot dibatasi ke AWS Wilayah tempat ia dibuat. Setelah Anda membuat snapshot dari volume EBS, Anda dapat menggunakannya untuk membuat volume baru di Wilayah yang sama.

Untuk informasi selengkapnya, lihat Membuat volume dari snapshot. Anda juga dapat menyalin snapshot di seluruh Wilayah, sehingga dapat menggunakan beberapa Wilayah untuk ekspansi geografis, migrasi pusat data, dan pemulihan bencana. Anda dapat menyalin snapshot yang dapat diakses yang memiliki status `completed`. Untuk informasi selengkapnya, lihat Menyalin snapshot Amazon EBS.

Dukungan enkripsi untuk snapshot

Ringkasan EBS sepenuhnya mendukung enkripsi EBS.

- Snapshot volume terenkripsi secara otomatis dienkripsi.
- Volume yang dibuat dari snapshot terenkripsi dienkripsi secara otomatis.
- Volume yang Anda buat dari snapshot tidak terenkripsi yang Anda miliki atau memiliki akses ke dapat dienkripsi. on-the-fly
- Saat menyalin snapshot yang tidak terenkripsi milik Anda, Anda dapat mengenkripsinya selama proses penyalinan.
- Saat menyalin snapshot terenkripsi milik Anda atau yang Anda miliki aksesnya, Anda dapat mengenkripsi ulang dengan kunci yang berbeda selama proses penyalinan.
- Snapshot pertama yang Anda ambil dari volume terenkripsi yang dibuat dari snapshot tanpa enkripsi selalu merupakan snapshot penuh.
- Snapshot pertama yang Anda ambil dari volume terenkripsi ulang, yang memiliki CMK berbeda dibandingkan dengan snapshot sumber, selalu merupakan snapshot penuh.

Dokumentasi lengkap tentang kemungkinan skenario enkripsi snapshot tersedia di Membuat snapshot Amazon EBS dan dalam Menyalin snapshot Amazon EBS.

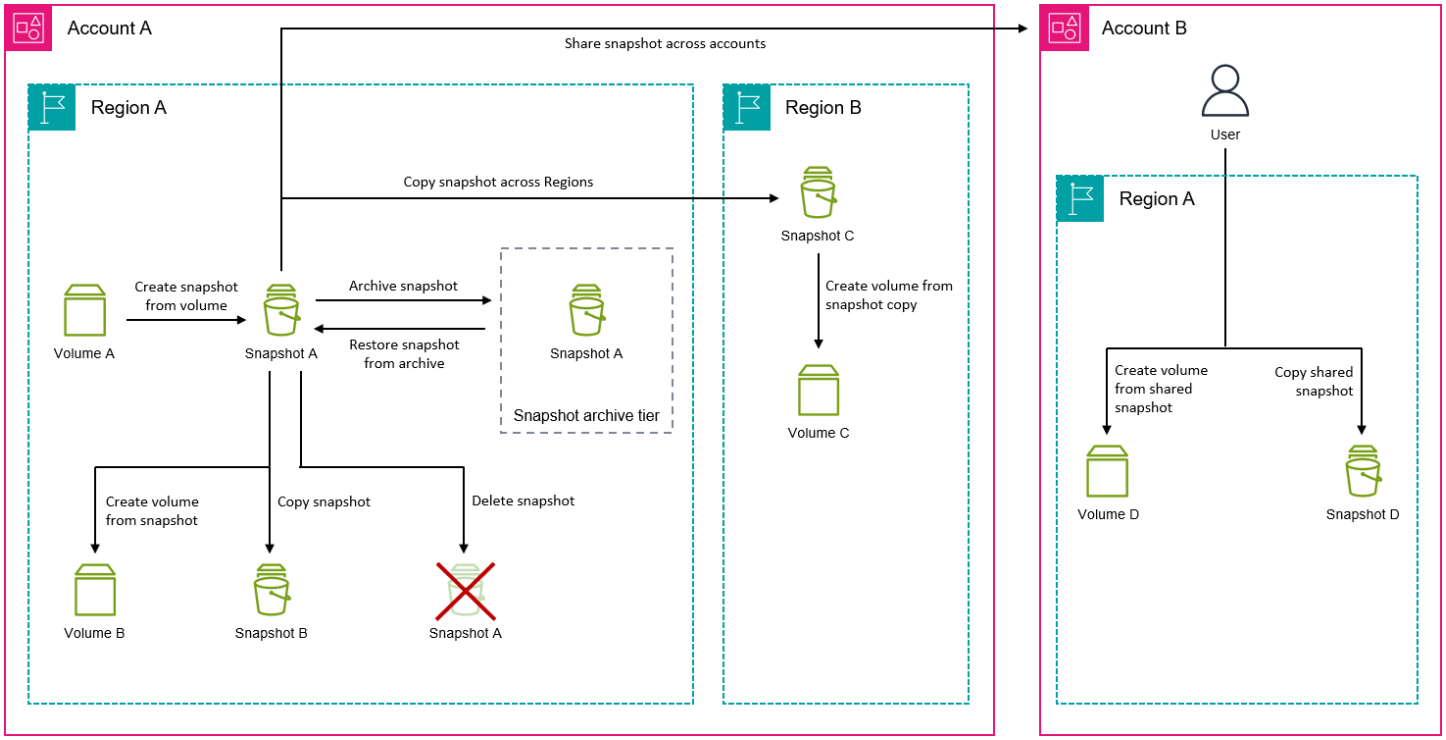
Untuk informasi selengkapnya, lihat Enkripsi EBS Amazon.

Siklus hidup snapshot Amazon EBS

Siklus hidup snapshot Amazon EBS dimulai dengan proses pembuatan. Anda membuat snapshot dari volume Amazon EBS. Anda dapat menggunakan snapshot untuk memulihkan volume Amazon EBS baru. Anda dapat membuat salinan snapshot baik di Wilayah yang sama, atau di Wilayah yang berbeda. Anda dapat berbagi foto dengan yang lain Akun AWS, baik secara publik maupun pribadi. Akun tersebut dapat memulihkan volume dari snapshot bersama, atau mereka dapat membuat

salinan snapshot bersama di akun mereka sendiri. Jika Anda tidak memerlukan akses langsung ke snapshot, Anda dapat mengarsipkannya untuk menghemat biaya penyimpanan.

Gambar berikut menunjukkan tindakan yang dapat Anda lakukan pada snapshot sebagai bagian dari siklus hidup snapshot.



Tugas

- Membuat snapshot Amazon EBS
- Melihat informasi snapshot Amazon EBS
- Menyalin snapshot Amazon EBS
- Membagikan snapshot Amazon EBS
- Mengarsipkan snapshot Amazon EBS
- Hapus snapshot Amazon EBS
- Otomatiskan siklus hidup snapshot

Membuat snapshot Amazon EBS

Untuk membuat snapshot yang konsisten dengan aplikasi pada instance Windows, lihat Membuat Snapshot Konsisten Aplikasi VSS.

Anda dapat membuat point-in-time snapshot dari volume EBS dan menggunakannya sebagai baseline untuk volume baru atau untuk cadangan data. Jika Anda membuat snapshot berkala dari suatu volume, snapshot tersebut bersifat inkremental—snapshot baru hanya menyimpan blok yang telah berubah sejak snapshot terakhir Anda.

Snapshot terjadi secara asinkron; point-in-time snapshot dibuat segera, tetapi status snapshot `pending` sampai snapshot selesai (ketika semua blok yang dimodifikasi telah ditransfer ke Amazon S3), yang dapat memakan waktu beberapa jam untuk snapshot awal yang besar atau snapshot berikutnya di mana banyak blok telah berubah. Saat prosesnya selesai, snapshot yang sedang berlangsung tidak terpengaruh oleh pembacaan dan penulisan terus-menerus ke volume.

Anda dapat mengambil snapshot dari volume terlampir yang sedang digunakan. Namun, snapshot hanya menangkap data yang telah ditulis ke volume Amazon EBS Anda pada saat perintah snapshot dikeluarkan. Ini mungkin mengecualikan data yang telah disimpan oleh aplikasi atau sistem operasi apa pun. Jika Anda dapat menjeda file apa pun yang ditulis ke volume dalam waktu yang cukup lama untuk mengambil snapshot, snapshot Anda akan lengkap. Namun, jika Anda tidak dapat menjeda semua tugas penulisan file ke volume, Anda harus melepas volume dari instans, mengeluarkan perintah snapshot, kemudian memasang ulang volume untuk memastikan snapshot yang konsisten dan lengkap. Anda dapat memasang ulang dan menggunakan volume Anda saat status snapshot adalah `pending`.

Untuk memudahkan manajemen snapshot, Anda dapat menandai snapshot Anda selama pembuatan atau menambahkan tanda sesudahnya. Misalnya, Anda dapat menerapkan tag yang menjelaskan volume asli dari mana snapshot dibuat, atau nama perangkat yang digunakan untuk melampirkan volume asli ke instance.

Enkripsi Snapshot

Snapshot yang diambil dari volume terenkripsi dienkripsi secara otomatis. Volume yang dibuat dari snapshot terenkripsi juga dienkripsi secara otomatis. Data dalam volume Anda yang dienkripsi dan snapshot apa pun yang terkait dilindungi baik saat diam maupun saat bergerak. Untuk informasi selengkapnya, lihat Enkripsi EBS Amazon.

Secara default, hanya Anda yang dapat membuat volume dari snapshot yang Anda miliki. Namun, Anda dapat membagikan snapshot Anda yang tidak terenkripsi dengan AWS akun tertentu, atau Anda dapat membagikannya dengan seluruh AWS komunitas dengan menjadikannya publik. Untuk informasi selengkapnya, lihat Membagikan snapshot Amazon EBS.

Anda dapat membagikan snapshot terenkripsi hanya dengan akun tertentu. AWS Agar orang lain dapat menggunakan snapshot bersama yang dienkripsi, Anda juga harus berbagi kunci CMK yang

digunakan untuk mengenkripsinya. Pengguna yang memiliki akses ke snapshot terenkripsi Anda harus membuat salinan pribadinya dan menggunakan salinan tersebut. Salinan snapshot bersama yang dienkripsi juga dapat dienkripsi ulang menggunakan kunci yang berbeda. Untuk informasi selengkapnya, lihat Membagikan snapshot Amazon EBS.

Snapshot multivolume

Anda dapat membuat snapshot multi-volume, yang merupakan point-in-time snapshot untuk semua, atau beberapa, volume yang dilampirkan ke sebuah instance.

Secara default, saat Anda membuat snapshot multivolume dari sebuah instans, Amazon EBS membuat snapshot dari semua volume (root dan data (non-root)) yang dilampirkan ke instans. Namun, Anda dapat memilih untuk membuat snapshot dari subset volume yang dilampirkan ke instans.

Anda dapat menandai snapshot multivolume Anda seperti jika Anda menginginkan satu snapshot volume. Kami menyarankan Anda menandai banyak snapshot volume untuk mengelolanya secara kolektif selama pemulihan, penyalinan, atau retensi. Anda juga dapat memilih untuk menyalin tanda secara otomatis dari volume sumber ke snapshot yang sesuai. Hal ini membantu Anda mengatur metadata snapshot, seperti kebijakan akses, informasi lampiran, dan alokasi biaya, untuk mencocokkan volume sumber.

Setelah snapshot dibuat, setiap snapshot diperlakukan sebagai snapshot individu. Anda dapat melakukan semua operasi snapshot, seperti memulihkan, menghapus, dan menyalin di seluruh Wilayah atau akun, sama seperti yag Anda lakukan dengan snapshot volume tunggal.

Snapshot multivolume yang crash-consistent biasanya dipulihkan sebagai satu set. Identifikasi snapshot yang berada dalam set crash-consistent dapat terbantu dengan menandai set Anda dengan ID instans, nama, atau detail relevan lainnya.

Setelah membuat snapshot Anda, mereka muncul di konsol EC2 Anda yang dibuat tepat. point-in-time

Jika salah satu snapshot untuk kumpulan snapshot multi-volume gagal, semua snapshot lainnya menampilkan status kesalahan dan `createSnapshots` CloudWatch peristiwa dengan hasil dikirim ke akun `failed` Anda. AWS Untuk informasi selengkapnya, lihat Membuat snapshot (membuatSnapshots).

Amazon Data Lifecycle Manager

Anda dapat membuat kebijakan siklus hidup snapshot untuk mengotomatisasi pembuatan dan retensi snapshot volume individual dan snapshot multivolume instans. Untuk informasi selengkapnya, lihat Amazon Data Lifecycle Manager.

Pertimbangan

Pertimbangan hal-hal berikut berlaku saat membuat snapshot:

- Saat Anda membuat snapshot untuk volume EBS yang berfungsi sebagai perangkat root, sebaiknya Anda menghentikan instans sebelum melakukan snapshot.
- Anda tidak dapat membuat snapshot dari instans yang mengaktifkan hibernasi, atau dari instans hibernasi. Jika Anda membuat snapshot atau AMI dari instans yang hibernasi atau mengaktifkan hibernasi, Anda mungkin tidak dapat terhubung ke instans baru yang diluncurkan dari AMI, atau dari AMI yang dibuat dari snapshot.
- Meskipun Anda dapat mengambil snapshot volume sementara snapshot volume tersebut sebelumnya ada dalam status `pending`, memiliki banyak snapshot volume `pending` dapat mengurangi performa volume hingga snapshot selesai.
- Ada batas satu snapshot `pending` untuk satu volume `st1` atau `sc1`, atau lima snapshot `pending` untuk satu volume dari tipe volume lainnya. Jika Anda menerima kesalahan `ConcurrentSnapshotLimitExceeded` saat mencoba membuat banyak snapshot secara bersamaan dari volume yang sama, tunggu satu atau beberapa snapshot `pending` selesai sebelum membuat snapshot lain dari volume tersebut.
- Ketika snapshot dibuat dari volume dengan kode AWS Marketplace produk, kode produk disebarkan ke snapshot.
- Saat membuat kumpulan snapshot multivolume dari instans, Anda dapat menentukan hingga 127 volume data (non-root) untuk dikecualikan. Jumlah maksimum volume Amazon EBS yang dapat dilampirkan ke instans bergantung pada tipe instans dan ukuran instans. Untuk informasi selengkapnya, lihat Batas volume instans.

Buat snapshot

Gunakan salah satu dari metode berikut ini untuk membuat snapshot dari volume yang ditentukan.

Console

Untuk membuat snapshot menggunakan konsol

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Snapshot, Buat snapshot.
3. Untuk Jenis sumber daya, pilih Volume.
4. Untuk ID Snapshot, pilih snapshot yang akan digunakan untuk membuat volume.

   Bidang Enkripsi menunjukkan status enkripsi volume yang dipilih. Jika volume yang dipilih dienkripsi, snapshot secara otomatis dienkripsi menggunakan kunci KMS yang sama. Jika volume yang dipilih tidak dienkripsi, snapshot tidak dienkripsi.
5. (Opsional) Untuk Deskripsi, masukkan deskripsi singkat untuk snapshot tersebut.
6. (Opsional) Untuk menetapkan tanda kustom ke snapshot, di bagian Tanda, pilih Tambahkan tanda, lalu masukkan pasangan nilai-kunci. Anda dapat menambahkan hingga 50 tanda.
7. Pilih Buat snapshot.

AWS CLI

Untuk membuat snapshot menggunakan AWS CLI

Menggunakan perintah create-snapshot.

Tools for Windows PowerShell

Untuk membuat snapshot menggunakan Alat untuk Windows PowerShell

Gunakan perintah New-EC2Snapshot.

Membuat snapshot multivolume

Saat membuat kumpulan snapshot multivolume dari sebuah instans, Anda dapat memilih apakah akan menyalin tanda dari volume sumber ke snapshot yang sesuai. Anda dapat menentukan apakah akan membuat snapshot volume root. Anda juga dapat menentukan apakah akan membuat snapshot dari semua volume data (non-root) yang dilampirkan ke instans, atau apakah akan membuat snapshot dari subset volume tersebut.

Pertimbangan-pertimbangan

- Snapshot multivolume mendukung hingga 128 volume Amazon EBS untuk setiap instans, yang mencakup volume root dan hingga 127 volume data (non-root). Jumlah maksimum volume Amazon EBS yang dapat dilampirkan ke instans bergantung pada tipe instans dan ukuran instans. Untuk informasi selengkapnya, lihat Batas volume instans.

Gunakan salah satu dari metode berikut ini untuk membuat snapshot dari volume suatu instans.

Console

Untuk membuat snapshot multivolume menggunakan konsol

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Snapshot, Buat snapshot.
3. Untuk Tipe sumber daya, pilih Instans.
4. (Opsional) Untuk Deskripsi, masukkan deskripsi singkat untuk snapshot tersebut. Deskripsi ini diterapkan ke semua snapshot.
5. (Opsional) Secara default, Amazon EBS membuat snapshot volume root instans. Jika Anda tidak ingin membuat snapshot dari volume root instans, pilih Kecualikan volume root.
6. (Opsional) Secara default, Amazon EBS membuat snapshot dari semua volume data (non-root) yang dilampirkan ke instans. Jika Anda ingin membuat snapshot dari subset volume data (non-root) yang dilampirkan ke instans, pilih Kecualikan volume data tertentu. Bagian Volume data terlampir mencantumkan semua volume data yang saat ini dilampirkan ke instans yang dipilih.

   Di bagian Volume data terlampir, pilih volume data yang tidak ingin Anda buat snapshot. Hanya volume yang tetap tidak dipilih yang akan disertakan dalam kumpulan snapshot multivolume. Anda dapat mengecualikan hingga 127 volume.
7. (Opsional) Untuk secara otomatis menyalin tanda dari volume sumber ke snapshot yang sesuai, untuk Salin tanda dari volume sumber, pilih Salin tanda. Tindakan ini akan mengatur metadata snapshot—seperti kebijakan akses, informasi lampiran, dan alokasi biaya—agar cocok dengan volume sumber.
8. (Opsional) Untuk menetapkan tanda kustom tambahan ke snapshot, di bagian Tanda, pilih Tambahkan tanda, lalu masukkan pasangan kunci-nilai. Anda dapat menambahkan hingga 50 tanda.
9. Pilih Buat snapshot.

Selama pembuatan snapshot, snapshot dikelola bersama. Jika salah satu snapshot dalam set volume gagal, snapshot lainnya dipindahkan ke status kesalahan untuk set volume. Anda dapat memantau kemajuan snapshot Anda menggunakan CloudWatchAcara. Setelah proses pembuatan snapshot selesai, buat CloudWatch peristiwa yang berisi status dan semua detail snapshot yang relevan untuk instance yang terpengaruh.

AWS CLI

Untuk membuat snapshot multi-volume menggunakan AWS CLI, gunakan perintah create-snapshots.

Jika Anda tidak ingin membuat snapshot dari volume root, untuk `--instance-specification ExcludeBootVolume`, tentukan`true`. Jika Anda tidak ingin membuat snapshot dari semua volume data (non-root) yang dilampirkan pada instans, untuk `--instance-specification ExcludeDataVolumes`, tentukan ID volume data yang tidak ingin Anda buat snapshot. Anda dapat menentukan hingga 127 volume data (non-root) untuk dikecualikan.

Tools for Windows PowerShell;

Untuk membuat snapshot multi-volume menggunakan Tools untuk Windows PowerShell, gunakan perintah. New-EC2SnapshotBatch

Jika Anda tidak ingin membuat snapshot dari volume root, untuk `-InstanceSpecification_ExcludeBootVolume`, tentukan`1`. Jika Anda tidak ingin membuat snapshot dari semua volume data (non-root) yang dilampirkan pada instans, untuk `-InstanceSpecification_ExcludeDataVolumes`, tentukan ID volume data yang tidak ingin Anda buat snapshot. Anda dapat menentukan hingga 127 volume data (non-root) untuk dikecualikan.

Jika semua snapshot berhasil diselesaikan, `createSnapshots` CloudWatch acara dengan hasil dikirim ke AWS akun Anda. `succeeded` Jika salah satu snapshot untuk kumpulan snapshot multi-volume gagal, semua snapshot lainnya menampilkan status kesalahan dan `createSnapshots` CloudWatch peristiwa dengan hasil dikirim ke akun `failed` Anda. AWS Untuk informasi selengkapnya, lihat Membuat snapshot (membuatSnapshots).

Cara menggunakan snapshot

Anda dapat menyalin snapshot, berbagi snapshot, dan membuat volume dari snapshot. Untuk informasi selengkapnya, lihat berikut ini:

- Menyalin snapshot Amazon EBS
- Membagikan snapshot Amazon EBS
- Membuat volume dari snapshot

Melihat informasi snapshot Amazon EBS

Anda dapat melihat informasi tentang grup keamanan Anda menggunakan salah satu metode berikut.

Console

Untuk melihat informasi snapshot menggunakan konsol

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Snapshot.
3. Untuk melihat snapshot yang Anda miliki, di sudut kiri atas layar, pilih Dimiliki oleh saya. Anda juga dapat memfilter daftar snapshot menggunakan tanda dan atribut snapshot lainnya. Di bidang Filter, pilih bidang atribut, lalu pilih atau masukkan nilai atribut. Misalnya, untuk hanya melihat snapshot terenkripsi, pilih Enkripsi, lalu masukkan `true`.
4. Untuk melihat informasi selengkapnya tentang snapshot tertentu, pilih ID di daftar.

AWS CLI

Untuk melihat informasi snapshot menggunakan AWS CLI

Gunakan perintah describe-snapshots.

Example Contoh 1: Filter berdasarkan tanda

Perintah berikut menjelaskan snapshot dengan tanda Stack=production.

```
aws ec2 describe-snapshots --filters Name=tag:Stack,Values=production
```

Example Contoh 2: Filter berdasarkan volume

Perintah berikut menjelaskan snapshot yang dibuat dari volume yang ditentukan.

```
aws ec2 describe-snapshots --filters Name=volume-id,Values=vol-049df61146c4d7901
```

Example Contoh 3: Memfilter berdasarkan usia snapshot

Dengan AWS CLI, Anda dapat menggunakan JMESPath untuk memfilter hasil menggunakan ekspresi. Misalnya, perintah berikut menampilkan ID dari semua snapshot yang dibuat oleh akun AWS Anda (direpresentasikan dengan *123456789012*) sebelum tanggal yang ditentukan (ditunjukkan dengan *31-03-2020*). Jika Anda tidak menentukan pemiliknya, hasilnya akan menyertakan semua snapshot publik.

```
aws ec2 describe-snapshots --filters Name=owner-id,Values=123456789012 --query
 "Snapshots[?(StartTime<='2020-03-31')].[SnapshotId]" --output text
```

Perintah berikut menampilkan ID dari semua snapshot yang dibuat dalam rentang tanggal tertentu.

```
aws ec2 describe-snapshots --filters Name=owner-id,Values=123456789012 --query
 "Snapshots[?(StartTime>='2019-01-01') && (StartTime<='2019-12-31')].[SnapshotId]"
 --output text
```

Tools for Windows PowerShell

Untuk melihat informasi snapshot menggunakan Alat untuk Windows PowerShell

Gunakan perintah Get-EC2Snapshot.

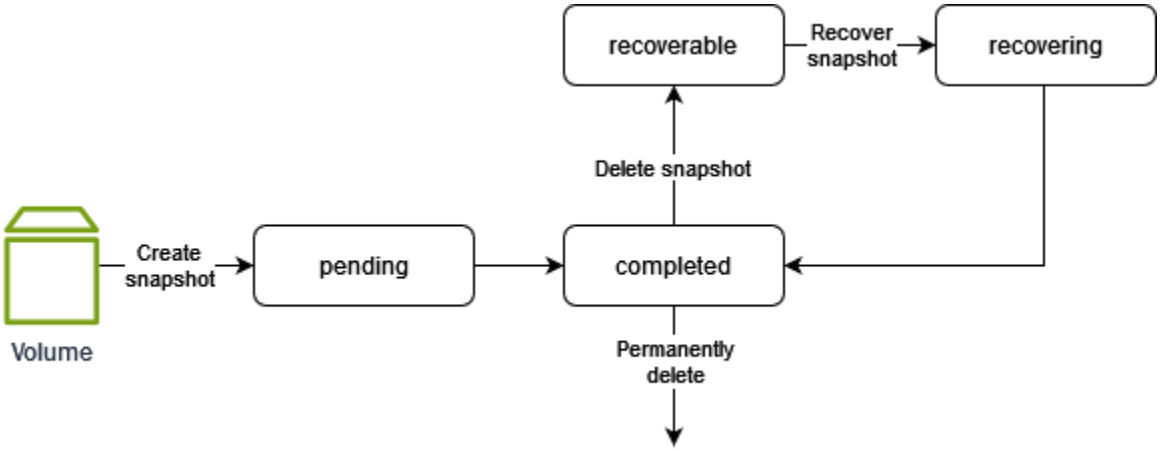
```
PS C:\> Get-EC2Snapshot -SnapshotId snapshot_id
```

Status snapshot

Transisi snapshot Amazon EBS melalui status yang berbeda dari saat dibuat hingga dihapus secara permanen.

Ilustrasi berikut menunjukkan transisi antara status snapshot. Saat Anda membuat snapshot, itu memasuki `pending` status. Setelah snapshot siap digunakan, ia memasuki `completed` status. Ketika Anda telah memutuskan bahwa Anda tidak lagi memerlukan snapshot, Anda dapat

menghapusnya. Jika Anda menghapus snapshot yang cocok dengan aturan retensi Recycle Bin, snapshot tersebut akan disimpan di Recycle Bin dan masuk ke status. `recoverable` Jika Anda memulihkan snapshot dari Recycle Bin, ia memasuki `recovering` status dan kemudian status. `completed` Jika tidak, itu dihapus secara permanen.



Tabel berikut merangkum status snapshot.

Status	Deskripsi
`pending`	Proses pembuatan snapshot masih berlangsung. Snapshot tidak dapat digunakan saat berada di `pending` negara bagian.
`completed`	Proses pembuatan snapshot telah selesai dan snapshot siap digunakan.
`recoverable`	Snapshot saat ini ada di Recycle Bin. Untuk menggunakan snapshot, Anda harus memulihkannya terlebih dahulu dari Recycle Bin.
`recovering`	Snapshot sedang dipulihkan dari Recycle Bin. Setelah snapshot dipulihkan, ia beralih ke `completed` status dan siap digunakan.
`error`	Proses pembuatan snapshot telah gagal. Snapshot tidak dapat digunakan jika berada di `error` negara bagian.

Menyalin snapshot Amazon EBS

Dengan Amazon EBS, Anda dapat membuat point-in-time snapshot volume, yang kami simpan untuk Anda di Amazon S3. Setelah Anda membuat snapshot dan selesai menyalin ke Amazon S3 (saat status snapshot), Anda dapat `completed` menyalinnya dari AWS satu Wilayah ke Wilayah lain, atau dalam Wilayah yang sama. Enkripsi sisi server Amazon S3 (256-bit AES) melindungi data bergerak snapshot selama operasi penyalinan. Salinan snapshot menerima ID yang berbeda dari ID snapshot asli.

Untuk menyalin snapshot multi-volume ke AWS Wilayah lain, ambil snapshot menggunakan tag yang Anda terapkan ke kumpulan snapshot multi-volume saat Anda membuatnya. Kemudian secara terpisah salin snapshot ke Wilayah lainnya.

Jika Anda ingin akun lain dapat menyalin snapshot Anda, Anda harus mengubah izin snapshot untuk mengizinkan akses ke akun itu atau membuat snapshot publik sehingga semua AWS akun dapat menyalinnya. Untuk informasi selengkapnya, lihat Membagikan snapshot Amazon EBS.

Untuk informasi tentang menyalin snapshot Amazon RDS, lihat Menyalin Snapshot DB dalam Panduan Pengguna Amazon RDS.

Kasus penggunaan

- Ekspansi geografis: Luncurkan aplikasi Anda di AWS Wilayah baru.
- Migrasi: Pindahkan aplikasi ke Wilayah baru, untuk memungkinkan ketersediaan yang lebih baik dan untuk meminimalkan biaya.
- Pemulihan bencana: Cadangkan data dan log Anda di berbagai lokasi geografis secara berkala. Jika terjadi bencana, Anda dapat memulihkan aplikasi Anda menggunakan point-in-time cadangan yang disimpan di Wilayah sekunder. Hal ini meminimalkan kehilangan data dan waktu pemulihan.
- Enkripsi: Mengenkripsi snapshot yang sebelumnya tidak dienkripsi, mengubah kunci yang dienkripsi untuk snapshot, atau membuat salinan yang Anda miliki untuk membuat volume darinya (untuk snapshot terenkripsi yang telah dibagikan dengan Anda).
- Persyaratan retensi data dan pengauditan: Salin snapshot EBS terenkripsi Anda dari satu akun AWS ke akun lainnya untuk menyimpan log data atau file lain untuk audit atau retensi data. Menggunakan akun yang berbeda membantu mencegah penghapusan snapshot yang tidak disengaja, dan melindungi Anda jika akun utama AWS Anda disusupi.

Daftar Isi

Prasyarat

- Anda dapat menyalin snapshot apa pun yang dapat diakses yang memiliki status `completed`, termasuk snapshot bersama dan snapshot yang telah Anda buat.
- Anda dapat menyalin AWS Marketplace, snapshot Impor/Ekspor VM, dan Storage Gateway, tetapi Anda harus memverifikasi bahwa snapshot didukung di Wilayah tujuan.
- Untuk menyalin snapshot terenkripsi, pengguna Anda harus memiliki izin berikut untuk menggunakan enkripsi Amazon EBS.
  - `kms:DescribeKey`
  - `kms:CreateGrant`
  - `kms:GenerateDataKey`
  - `kms:GenerateDataKeyWithoutPlaintext`
  - `kms:ReEncrypt`
  - `kms:Decrypt`
- Untuk menyalin snapshot terenkripsi yang dibagikan dari AWS akun lain, Anda harus memiliki izin untuk menggunakan kunci terkelola pelanggan yang digunakan untuk mengenkripsi snapshot. Untuk informasi selengkapnya, lihat Bagikan kunci KMS.

Pertimbangan

- Ada batas permintaan sejumlah `20` salinan snapshot bersamaan per Wilayah tujuan. Jika melebihi kuota ini, Anda menerima `ResourceLimitExceeded` kesalahan. Jika Anda menerima kesalahan ini, tunggu satu atau beberapa permintaan salinan selesai sebelum membuat permintaan salinan snapshot baru.
- Tanda yang ditentukan pengguna tidak disalin dari snapshot sumber ke snapshot baru. Anda dapat menambahkan tanda yang ditentukan pengguna selama atau setelah operasi penyalinan.

- Snapshot yang dibuat oleh operasi penyalinan snapshot memiliki ID volume arbitrer, seperti `vol-ffff` atau `vol-ffffffff`. ID volume yang tidak boleh digunakan untuk tujuan apa pun.
- Izin tingkat sumber daya yang ditentukan untuk operasi penyalinan berlaku hanya untuk snapshot baru. Anda tidak dapat menentukan izin tingkat sumber daya untuk snapshot sumber. Sebagai contoh, lihat Contoh: Menyalin snapshot.

Harga

- Untuk informasi harga tentang menyalin snapshot di seluruh AWS Wilayah dan akun, lihat Harga Amazon EBS.
- Jika Anda menyalin snapshot dan mengenkripsinya ke kunci KMS baru, salinan lengkap (tidak inkremental) dibuat. Hal ini menyebabkan biaya penyimpanan tambahan.
- Jika Anda menyalin snapshot ke Wilayah baru, salinan lengkap (non-inkremental) akan dibuat. Hal ini menyebabkan biaya penyimpanan tambahan. Salinan berikutnya dari snapshot yang sama bersifat inkremental.
- Jika Anda menggunakan transfer data eksternal atau lintas wilayah, biaya transfer data EC2 tambahan akan berlaku. Dan jika Anda menghapus snapshot apa pun setelah inisiasi, Anda tetap dikenai biaya untuk data yang telah ditransfer.

Menyalin snapshot inkremental

Jika salinan snapshot bersifat inkremental ditentukan oleh salinan snapshot yang baru saja diselesaikan. Saat Anda menyalin snapshot di seluruh Wilayah atau akun, salinan tersebut adalah salinan inkremental jika syarat berikut terpenuhi:

- Snapshot disalin ke Wilayah atau akun tujuan sebelumnya.
- Salinan snapshot terbaru masih ada di Wilayah atau akun tujuan.
- Salinan snapshot terbaru belum diarsipkan.
- Semua salinan snapshot di Wilayah atau akun tujuan tidak dienkripsi atau dienkripsi menggunakan kunci KMS yang sama.

Jika salinan snapshot terbaru dihapus, salinan berikutnya adalah salinan penuh, bukan salinan inkremental. Jika salinan masih tertunda saat Anda memulai salinan lain, salinan kedua dimulai hanya setelah salinan pertama selesai.

Operasi penyalinan snapshot dalam akun dan Wilayah yang sama menggunakan kunci KMS yang sama menghasilkan salinan tambahan.

Penyalinan snapshot bertahap mengurangi waktu yang diperlukan untuk menyalin snapshot dan menghemat biaya transfer data dan penyimpanan dengan tidak menggandakan data.

Kami sarankan Anda menandai snapshot dengan volume dan waktu pembuatan sehingga Anda dapat terus melacak salinan snapshot terbaru dari volume di Wilayah atau akun tujuan.

Untuk melihat apakah salinan snapshot Anda bersifat inkremental, periksa peristiwa CopySnapshot. CloudWatch

Enkripsi dan penyalinan snapshot

Saat menyalin snapshot, Anda dapat mengenkripsi salinan atau menentukan kunci KMS yang berbeda dari yang asli, dan snapshot salinan yang dihasilkan menggunakan kunci KMS baru. Namun, mengubah status enkripsi snapshot selama operasi penyalinan dapat menghasilkan salinan penuh (bukan inkremental), yang dapat menimbulkan biaya transfer dan penyimpanan data yang lebih besar. Untuk informasi selengkapnya, lihat Menyalin snapshot inkremental.

Untuk menyalin snapshot terenkripsi yang dibagikan dari AWS akun lain, Anda harus memiliki izin untuk menggunakan snapshot dan kunci terkelola pelanggan (CMK) yang digunakan untuk mengenkripsi snapshot. Saat menggunakan snapshot terenkripsi yang dibagikan dengan Anda, kami sarankan agar Anda mengenkripsi ulang snapshot dengan menyalinnya menggunakan kunci KMS yang Anda miliki. Langkah ini melindungi Anda jika kunci KMS asli diretas, atau jika pemilik mencabutnya, yang dapat menyebabkan Anda kehilangan akses ke volume terenkripsi apa pun yang Anda buat menggunakan snapshot. Untuk informasi selengkapnya, lihat Membagikan snapshot Amazon EBS.

Anda menerapkan enkripsi ke salinan snapshot EBS dengan menetapkan parameter `Encrypted` ke `true`. (Parameter `Encrypted` bersifat opsional jika enkripsi secara default diaktifkan).

Atau, Anda dapat menggunakan `KmsKeyId` untuk menentukan kunci kustom yang digunakan untuk mengenkripsi salinan snapshot. (Parameter `Encrypted` juga harus diatur ke `true`, bahkan jika enkripsi secara default diaktifkan.) Jika `KmsKeyId` tidak ditentukan, kunci yang digunakan untuk enkripsi tergantung pada kondisi enkripsi snapshot sumber dan kepemilikannya.

Tabel berikut menjelaskan hasil enkripsi untuk setiap kombinasi pengaturan saat menyalin snapshot yang Anda miliki dan snapshot yang dibagikan kepada Anda.

Enkripsi secara default	Adalah **`Encrypted`** parameter ditetapkan?	Status enkripsi snapshot sumber	Default (tidak ditentukan kunci KMS)	Kustom (kunci KMS ditentukan)
Nonaktif	Tidak	Tidak terenkripsi	Tidak terenkripsi	T/A
		Dienkripsi	Dienkripsi oleh Kunci yang dikelola AWS	
	Ya	Tidak terenkripsi	Dienkripsi dengan kunci KMS default	Dienkripsi dengan kunci KMS yang ditentukan**
		Dienkripsi	Dienkripsi dengan kunci KMS default	
Aktif	Tidak	Tidak terenkripsi	Dienkripsi dengan kunci KMS default	N/A
		Dienkripsi	Dienkripsi dengan kunci KMS default	
	Ya	Tidak terenkripsi	Dienkripsi dengan kunci KMS default	Dienkripsi dengan kunci KMS yang ditentukan**
		Dienkripsi	Dienkripsi dengan kunci KMS default	

** Ini adalah kunci KMS yang ditentukan dalam tindakan penyalinan snapshot. Kunci KMS ini digunakan sebagai pengganti kunci KMS default untuk akun dan Wilayah.

Menyalin snapshot

Untuk menyalin snapshot, gunakan salah satu metode berikut.

Console

Untuk menyalin snapshot menggunakan konsol

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Snapshot.
3. Pilih snapshot yang akan dihapus, lalu pilih Tindakan, Salin snapshot.
4. (Opsional) Untuk Deskripsi, masukkan deskripsi singkat untuk salinan snapshot tersebut.

   Secara default, deskripsi mencakup informasi tentang snapshot sumber sehingga Anda dapat mengidentifikasi salinan dari versi asli. Anda dapat mengubah deskripsi ini sesuai kebutuhan.
5. Untuk Wilayah Tujuan, pilih Wilayah tempat membuat salinan snapshot.
6. Tentukan status enkripsi untuk salinan snapshot.

   Jika snapshot yang dipilih dienkripsi, atau jika akun Anda diaktifkan untuk enkripsi secara default, salinan snapshot secara otomatis dienkripsi dan Anda tidak dapat mengubah status enkripsinya.

   Jika snapshot sumber tidak dienkripsi dan akun Anda tidak diaktifkan untuk enkripsi secara default, enkripsi bersifat opsional. Untuk mengenkripsi salinan snapshot, untuk Enkripsi, pilih Enkripsi snapshot ini. Kemudian, untuk Kunci KMS, pilih kunci KMS yang akan digunakan untuk mengenkripsi snapshot di Wilayah tujuan.
7. Pilih Salin Snapshot.

AWS CLI

Untuk menyalin snapshot menggunakan AWS CLI

Gunakan perintah copy-snapshot.

Tools for Windows PowerShell

Untuk menyalin snapshot menggunakan Alat untuk Windows PowerShell

Gunakan perintah Copy-EC2Snapshot.

Untuk memeriksa kegagalan

Jika Anda mencoba menyalin snapshot terenkripsi tanpa izin untuk menggunakan kunci enkripsi, operasi akan gagal secara diam-diam. Status kesalahan tidak ditampilkan di konsol hingga Anda menyegarkan halaman. Anda juga dapat memeriksa status snapshot dari baris perintah, seperti dalam contoh berikut.

```
aws ec2 describe-snapshots --snapshot-id snap-0123abcd
```

Jika salinan gagal karena izin kunci yang tidak mencukupi, Anda melihat pesan berikut: "StateMessage“: “ID kunci yang diberikan tidak dapat diakses”.

Saat menyalin snapshot terenkripsi, Anda harus memiliki izin `DescribeKey` pada CMK default. Secara tegas menolak izin ini akan menyebabkan kegagalan penyalinan. Untuk informasi tentang mengelola kunci CMK, lihat Autentikasi dan kontrol akses untuk AWS KMS.

Membagikan snapshot Amazon EBS

Anda dapat memodifikasi izin dari snapshot jika Anda ingin berbagi dengan akun AWS lainnya. Anda dapat berbagi snapshot secara publik dengan semua AWS akun lain, atau Anda dapat membagikannya secara pribadi dengan AWS akun individual yang Anda tentukan. Pengguna yang Anda beri wewenang dapat menggunakan snapshot yang Anda bagikan untuk membuat volume EBS mereka sendiri, sementara snapshot asli Anda tetap tidak terpengaruh.

> **Important**
>
> Saat Anda berbagi snapshot, Anda memberikan akses ke semua data di snapshot kepada orang lain. Bagikan snapshot hanya kepada individu yang Anda percayai dengan semua data snapshot Anda.

Untuk mencegah berbagi snapshot secara publik, Anda dapat mengaktifkan blokir akses publik untuk snapshot. Untuk informasi selengkapnya, lihat Memblokir akses publik ke AMI Anda.

Topik

- Sebelum Anda berbagi snapshot
- Membagikan snapshot
- Bagikan kunci KMS

- Melihat snapshot yang dibagikan dengan Anda
- Menggunakan snapshot yang dibagikan dengan Anda
- Menentukan penggunaan snapshot yang Anda bagikan

Sebelum Anda berbagi snapshot

Hal-hal berikut berlaku saat berbagi snapshot:

- Jika pemblokiran akses publik untuk snapshot diaktifkan untuk Wilayah, upaya guna membagikan snapshot secara publik akan diblokir. Snapshot masih dapat dibagikan secara privat.
- Snapshot dibatasi untuk Wilayah tempatnya dibuat. Untuk berbagi snapshot dengan Wilayah lain, salin snapshot ke Wilayah tersebut, lalu bagikan salinannya. Untuk informasi selengkapnya, lihat Menyalin snapshot Amazon EBS.
- Anda tidak dapat berbagi snapshot yang dienkripsi dengan Kunci yang dikelola AWS default. Anda hanya dapat berbagi snapshot yang dienkripsi dengan kunci yang dikelola pelanggan. Untuk informasi selengkapnya, lihat Membuat Kunci di Panduan Developer AWS Key Management Service .
- Anda hanya dapat berbagi snapshot yang tidak terenkripsi secara publik.
- Saat Anda berbagi snapshot terenkripsi, Anda juga harus berbagi kunci yang dikelola pelanggan yang digunakan untuk mengenkripsi snapshot. Untuk informasi selengkapnya, lihat Bagikan kunci KMS.

Membagikan snapshot

Anda dapat berbagi snapshot menggunakan salah satu metode yang dijelaskan di bagian ini.

Console

Untuk berbagi snapshot

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Snapshot.
3. Pilih snapshot yang akan dibagikan, lalu pilih Tindakan, Ubah izin.
4. Tentukan izin snapshot. Pengaturan saat ini menunjukkan izin berbagi snapshot saat ini.
    - Untuk membagikan snapshot secara publik dengan semua AWS akun, pilih Publik.

- Untuk membagikan snapshot secara pribadi dengan AWS akun tertentu, pilih Pribadi. Kemudian, di bagian Berbagi akun, pilih Tambah akun, dan masukkan 12 digit ID akun (tanpa tanda hubung) yang akan dibagikan.

5. Pilih Simpan perubahan.

AWS CLI

Izin untuk snapshot ditentukan menggunakan atribut `createVolumePermission` snapshot. Untuk membuat snapshot publik, atur grup ke `all`. Untuk berbagi snapshot dengan AWS akun tertentu, atur pengguna ke ID AWS akun.

Untuk berbagi snapshot secara publik

Gunakan perintah modify-snapshot-attribute.

Untuk `--attribute`, tentukan `createVolumePermission`. Untuk `--operation-type`, tentukan `add`. Untuk `--group-names`, tentukan `all`.

```
$  aws ec2 modify-snapshot-attribute --snapshot-id 1234567890abcdef0 --attribute
 createVolumePermission --operation-type add --group-names all
```

Untuk berbagi snapshot secara privat

Gunakan perintah modify-snapshot-attribute.

Untuk `--attribute`, tentukan `createVolumePermission`. Untuk `--operation-type`, tentukan `add`. Untuk`--user-ids`, tentukan ID 12 digit AWS akun yang dapat digunakan untuk berbagi snapshot.

```
$  aws ec2 modify-snapshot-attribute --snapshot-id 1234567890abcdef0 --attribute
 createVolumePermission --operation-type add --user-ids 123456789012
```

Tools for Windows PowerShell

Izin untuk snapshot ditentukan menggunakan atribut `createVolumePermission` snapshot. Untuk membuat snapshot publik, atur grup ke `all`. Untuk berbagi snapshot dengan AWS akun tertentu, atur pengguna ke ID AWS akun.

Untuk berbagi snapshot secara publik

Gunakan perintah Edit-EC2SnapshotAttribute.

Untuk `-Attribute`, tentukan `CreateVolumePermission`. Untuk `-OperationType`, tentukan `Add`. Untuk `-GroupName`, tentukan `all`.

```
PS C:\>  Edit-EC2SnapshotAttribute -SnapshotId 1234567890abcdef0 -Attribute
 CreateVolumePermission -OperationType Add -GroupName all
```

Untuk berbagi snapshot secara privat

Gunakan perintah Edit-EC2SnapshotAttribute.

Untuk `-Attribute`, tentukan `CreateVolumePermission`. Untuk `-OperationType`, tentukan `Add`. Untuk`UserId`, tentukan ID 12 digit AWS akun yang dapat digunakan untuk berbagi snapshot.

```
PS C:\>  Edit-EC2SnapshotAttribute -SnapshotId 1234567890abcdef0 -Attribute
 CreateVolumePermission -OperationType Add -UserId 123456789012
```

Bagikan kunci KMS

Saat Anda berbagi snapshot terenkripsi, Anda juga harus berbagi kunci yang dikelola pelanggan yang digunakan untuk mengenkripsi snapshot. Anda dapat menerapkan izin lintas akun ke kunci yang dikelola pelanggan baik saat dibuat atau di lain waktu.

Pengguna kunci yang dikelola pelanggan bersama Anda yang mengakses snapshot terenkripsi harus diberikan izin untuk melakukan tindakan berikut pada kunci tersebut:

- `kms:DescribeKey`
- `kms:CreateGrant`
- `kms:GenerateDataKey`
- `kms:GenerateDataKeyWithoutPlaintext`
- `kms:ReEncrypt`
- `kms:Decrypt`

> **Tip**
>
> Untuk mengikuti prinsip hak akses paling rendah, jangan biarkan akses penuh ke `kms:CreateGrant`. Sebagai gantinya, gunakan tombol `kms:GrantIsForAWSResource`

kondisi untuk memungkinkan pengguna membuat hibah pada kunci KMS hanya ketika hibah dibuat atas nama pengguna oleh layanan. AWS

Untuk informasi selengkapnya tentang mengontrol akses ke kunci yang dikelola pelanggan, lihat Menggunakan kebijakan kunci di AWS KMS di Panduan Developer AWS Key Management Service .

Untuk berbagi kunci terkelola pelanggan menggunakan AWS KMS konsol

1. Buka AWS KMS konsol di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Pilih Kunci yang dikelola pelanggan di panel navigasi.
4. Di kolom Alias, pilih alias (tautan teks) dari kunci yang dikelola pelanggan yang Anda gunakan untuk mengenkripsi snapshot. Detail kunci terbuka di halaman baru.
5. Di bagian Kebijakan kunci, Anda melihat tampilan kebijakan atau tampilan default. Tampilan kebijakan menampilkan dokumen kebijakan kunci. Tampilan default menampilkan bagian untuk Administrator kunci, Penghapusan kunci, Penggunaan Kunci, dan Akun AWS lainnya. Tampilan default ditampilkan jika Anda membuat kebijakan di konsol dan belum menyesuaikannya. Jika tampilan default tidak tersedia, Anda perlu mengedit kebijakan secara manual dalam tampilan kebijakan. Untuk informasi selengkapnya, lihat Melihat Kebijakan Kunci (Konsol) dalam AWS Key Management Service Panduan Developer.

   Gunakan tampilan kebijakan atau tampilan default, bergantung pada tampilan mana yang dapat Anda akses, untuk menambahkan satu atau beberapa ID AWS akun ke kebijakan, sebagai berikut:

   - (Tampilan kebijakan) Pilih Edit. Tambahkan satu atau beberapa ID AWS akun ke pernyataan berikut: `"Allow use of the key"` dan`"Allow attachment of persistent resources"`. Pilih Simpan perubahan. Dalam contoh berikut, ID AWS akun `444455556666` ditambahkan ke kebijakan.

     ```
     {
       "Sid": "Allow use of the key",
       "Effect": "Allow",
       "Principal": {"AWS": [
         "arn:aws:iam::111122223333:user/KeyUser",
         "arn:aws:iam::444455556666:root"
       ]},
       "Action": [
     ```

```
      "kms:Encrypt",
      "kms:Decrypt",
      "kms:ReEncrypt*",
      "kms:GenerateDataKey*",
      "kms:DescribeKey"
    ],
    "Resource": "*"
  },
  {
    "Sid": "Allow attachment of persistent resources",
    "Effect": "Allow",
    "Principal": {"AWS": [
      "arn:aws:iam::111122223333:user/KeyUser",
      "arn:aws:iam::444455556666:root"
    ]},
    "Action": [
      "kms:CreateGrant",
      "kms:ListGrants",
      "kms:RevokeGrant"
    ],
    "Resource": "*",
    "Condition": {"Bool": {"kms:GrantIsForAWSResource": true}}
  }
```

- (Tampilan default) Gulir ke bawah ke AWS akun lain. Pilih Tambahkan AWS akun lain dan masukkan ID AWS akun seperti yang diminta. Untuk menambahkan akun lain, pilih Tambahkan AWS akun lain dan masukkan ID AWS akun. Setelah Anda menambahkan semua akun AWS , pilih Simpan perubahan.

Melihat snapshot yang dibagikan dengan Anda

Anda dapat melihat snapshot yang dibagikan dengan Anda menggunakan salah satu metode berikut.

Console

Untuk melihat snapshot yang dibagikan menggunakan konsol

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Snapshot.
3. Filter snapshot yang tercantum. Di sudut kiri atas layar, pilih salah satu opsi berikut:

- Snapshot privat — Untuk melihat hanya snapshot yang dibagikan dengan Anda secara privat.
- Snapshot publik — Untuk melihat hanya snapshot yang dibagikan dengan Anda secara publik.

AWS CLI

Untuk melihat izin snapshot menggunakan baris perintah

Gunakan perintah describe-snapshot-attribute.

Tools for Windows PowerShell

Untuk melihat izin snapshot menggunakan baris perintah

Gunakan perintah Get-EC2SnapshotAttribute.

Menggunakan snapshot yang dibagikan dengan Anda

Untuk menggunakan snapshot yang dibagikan yang tidak dienkripsi

Cari snapshot yang dibagikan berdasarkan ID atau deskripsi. Untuk informasi selengkapnya, lihat Melihat snapshot yang dibagikan dengan Anda. Anda dapat menggunakan snapshot ini sebagaimana dengan snapshot lain yang Anda miliki di akun Anda. Misalnya, Anda dapat membuat volume dari snapshot atau menyalinnya ke Wilayah yang berbeda.

Untuk menggunakan snapshot yang dibagikan yang terenkripsi

Cari snapshot yang dibagikan berdasarkan ID atau deskripsi. Untuk informasi selengkapnya, lihat Melihat snapshot yang dibagikan dengan Anda. Buat salinan snapshot yang dibagikan di akun Anda, dan enkripsi salinannya dengan kunci KMS yang Anda miliki. Anda kemudian dapat menggunakan salinan untuk membuat volume atau Anda dapat menyalinnya ke Wilayah yang berbeda.

Menentukan penggunaan snapshot yang Anda bagikan

Anda dapat menggunakan AWS CloudTrail untuk memantau apakah snapshot yang telah Anda bagikan dengan orang lain disalin atau digunakan untuk membuat volume. Peristiwa berikut masuk CloudTrail:

- SharedSnapshotCopyInitiated— Snapshot bersama sedang disalin.

- SharedSnapshotVolumeCreated— Snapshot bersama sedang digunakan untuk membuat volume.

Untuk informasi selengkapnya tentang penggunaan CloudTrail, lihat Log panggilan Amazon EC2 dan Amazon EBS API dengan. AWS CloudTrail

Mengarsipkan snapshot Amazon EBS

Arsip Snapshot Amazon EBS adalah tingkat penyimpanan baru yang dapat Anda gunakan untuk penyimpanan berbiaya rendah dan jangka panjang dari snapshot yang jarang diakses yang tidak membutuhkan pengambilan yang sering dan cepat.

Secara default, saat Anda membuat snapshot, snapshot disimpan di tingkat Standar Snapshot Amazon EBS (tingkat standar). Snapshot yang disimpan di tingkat standar bersifat inkremental. Hal ini berarti bahwa hanya blok pada volume yang berubah setelah snapshot terbaru Anda disimpan.

Saat Anda mengarsipkan snapshot, snapshot inkremental akan dikonversi ke snapshot penuh, dan akan dipindahkan dari tingkat standar ke tingkat Arsip Snapshot Amazon EBS (tingkat arsip). Snapshot lengkap mencakup semua blok yang ditulis ke volume pada saat snapshot dibuat.

Saat Anda perlu mengakses snapshot yang diarsipkan, Anda dapat memulihkannya dari tingkat arsip ke tingkat standar, lalu menggunakannya dengan cara yang sama seperti Anda menggunakan snapshot lain di akun Anda.

Arsip Snapshot Amazon EBS menawarkan biaya penyimpanan snapshot hingga 75 persen lebih rendah untuk snapshot yang Anda rencanakan untuk disimpan selama 90 hari atau lebih lama dan yang jarang perlu Anda akses.

Beberapa kasus penggunaan khas meliputi:

- Mengarsipkan satu-satunya snapshot volume, seperti snapshot end-of-project
- Mengarsipkan snapshot point-in-time inkremental lengkap untuk alasan kepatuhan.
- Mengarsipkan snapshot inkremental bulanan, triwulanan, atau tahunan.

Topik

- Pertimbangan dan batasan
- Harga dan penagihan
- Kuota
- Panduan dan praktik terbaik untuk mengarsipkan snapshot

- Izin IAM yang diperlukan
- Bekerja dengan pengarsipan snapshot
- Memantau pengarsipan snapshot

Pertimbangan dan batasan

Pertimbangan

- Periode arsip minimum adalah 90 hari. Jika Anda menghapus atau memulihkan snapshot yang diarsipkan secara permanen sebelum periode arsip minimum 90 hari, Anda akan ditagih untuk sisa hari di tingkat arsip, dibulatkan ke jam terdekat. Untuk informasi selengkapnya, lihat Harga dan penagihan.
- Diperlukan waktu hingga 72 jam untuk memulihkan snapshot yang diarsipkan dari tingkat arsip ke tingkat standar, tergantung pada ukuran snapshot.
- Snapshot yang diarsipkan selalu merupakan snapshot penuh. Snapshot lengkap berisi semua blok yang ditulis ke volume pada saat snapshot dibuat. Snapshot lengkap kemungkinan akan lebih besar dari snapshot inkremental dari tempat snapshot tersebut dibuat. Namun, jika Anda hanya memiliki satu snapshot inkremental volume pada tingkat standar, ukuran snapshot penuh di tingkat arsip akan berukuran sama dengan snapshot di tingkat standar. Ini karena snapshot pertama yang diambil dari sebuah volume selalu merupakan snapshot penuh.
- Pengarsipan direkomendasikan untuk snapshot bulanan, triwulanan, atau tahunan. Mengarsipkan snapshot inkremental harian dari satu volume dapat menyebabkan biaya yang lebih tinggi jika dibandingkan dengan menyimpannya di tingkat standar.
- Saat snapshot diarsipkan, data snapshot yang direferensikan oleh snapshot lain dalam garis keturunan snapshot dipertahankan di tingkat standar. Biaya data dan penyimpanan yang terkait dengan data yang direferensikan yang disimpan pada tingkat standar dialokasikan ke snapshot berikutnya dalam garis keturunan. Ini memastikan bahwa snapshot berikutnya dalam garis keturunan tidak terpengaruh oleh arsip.
- Jika Anda menghapus snapshot yang diarsipkan yang cocok dengan aturan retensi Keranjang Sampah, snapshot yang diarsipkan akan disimpan di Keranjang Sampah untuk periode retensi yang ditentukan dalam aturan retensi. Untuk menggunakan snapshot, Anda harus terlebih dahulu memulihkannya dari Keranjang Sampah, lalu mengembalikannya dari tingkat arsip. Untuk informasi selengkapnya, lihat Recycle Bin danHarga dan penagihan.
- Anda tidak dapat menggunakan snapshot yang diarsipkan dalam pemetaan perangkat blok atau untuk membuat volume Amazon EBS.

- Anda dapat mengarsipkan snapshot yang dibuat AWS Backup menggunakan Konsol AWS Backup, API, atau alat baris perintah. Untuk informasi selengkapnya, lihat Membuat rencana cadangan di Panduan Developer AWS Backup .

Batasan

- Anda dapat mengarsipkan snapshot yang ada dalam status `completed` saja.
- Anda hanya dapat mengarsipkan snapshot yang Anda miliki di akun Anda. Untuk mengarsipkan snapshot yang dibagikan dengan Anda, pertama-tama salin snapshot ke akun Anda dan kemudian arsipkan salinan snapshot.
- Sebelum dapat menggunakan snapshot yang diarsipkan, Anda harus terlebih dahulu memulihkannya ke tingkat standar. Memulihkan ke tingkat standar diperlukan untuk membuat volume dari snapshot melalui operasi API `CreateVolume` dan `RunInstances` serta untuk berbagi atau menyalin snapshot. Untuk informasi selengkapnya, lihat Memulihkan snapshot yang diarsipkan.
- Anda dapat mengarsipkan snapshot yang dikaitkan dengan satu atau beberapa AMI hanya jika semua AMI terkait dinonaktifkan. Untuk informasi selengkapnya, lihat Menonaktifkan AMI.
- Anda tidak dapat mengaktifkan AMI yang dinonaktifkan jika snapshot terkait dipulihkan sementara. Semua snapshot terkait harus dipulihkan secara permanen sebelum Anda dapat mengaktifkan AMI.
- Anda tidak dapat membatalkan arsip snapshot atau proses pemulihan snapshot setelah dimulai.
- Anda dapat mengunci snapshot yang diarsipkan. Jika Anda mengarsipkan snapshot yang telah Anda bagikan dengan akun lain, akun yang digunakan untuk berbagi snapshot kehilangan akses setelah snapshot diarsipkan.
- Anda dapat menyalin snapshot yang diarsipkan. Jika Anda perlu menyalin snapshot yang diarsipkan, Anda harus memulihkannya terlebih dahulu.
- Anda tidak dapat mengaktifkan pemulihan snapshot cepat untuk snapshot yang diarsipkan. Pemulihan snapshot cepat dinonaktifkan secara otomatis saat snapshot diarsipkan. Jika Anda perlu menggunakan pemulihan snapshot cepat, Anda harus mengaktifkannya secara manual setelah memulihkan snapshot.

Harga dan penagihan

Snapshot yang diarsipkan dikenai biaya dengan tarif 0,0125 USD per GB-bulan. Misalnya, jika Anda mengarsipkan snapshot sebesar 100 GiB, Anda dikenai biaya sebesar 1,25 USD (100 GiB * 0,0125 USD) per bulan.

Pemulihan snapshot dikenai biaya dengan tarif 0,03 USD per GB data yang dipulihkan. Misalnya, jika Anda mengembalikan snapshot 100 GiB dari tingkat arsip, Anda akan dikenai biaya satu kali sebesar 3 USD (100 GiB * 0,03 USD).

Setelah snapshot dipulihkan ke tingkat standar, snapshot dikenai biaya tarif standar untuk snapshot sebesar 0,05 USD per GB-bulan.

Untuk informasi selengkapnya, lihat harga Amazon EBS.

Penagihan untuk periode arsip minimum

Periode arsip minimum adalah 90 hari. Jika Anda menghapus atau memulihkan snapshot yang diarsipkan secara permanen sebelum periode arsip minimum 90 hari, Anda akan ditagih biaya prorata yang sama dengan biaya penyimpanan tingkat arsip untuk sisa harinya, dibulatkan ke jam terdekat. Misalnya, jika Anda menghapus atau memulihkan snapshot yang diarsipkan secara permanen setelah 40 hari, Anda akan dikenai biaya selama 50 hari tersisa dari periode arsip minimum.

> **Note**
>
> Memulihkan snapshot yang diarsipkan sementara sebelum periode arsip minimum 90 hari tidak dikenai biaya ini.

Pemulihan sementara

Saat Anda memulihkan snapshot sementara, snapshot dipulihkan dari tingkat arsip ke tingkat standar, dan salinan snapshot tetap berada di tingkat arsip. Anda dikenai biaya untuk snapshot di tingkat standar dan salinan snapshot di tingkat arsip selama periode pemulihan sementara. Ketika snapshot yang dipulihkan sementara dihapus dari tingkat standar, Anda tidak lagi ditandaih untuk itu, dan Anda ditandaih untuk snapshot di tingkat arsip saja.

Pemulihan permanen

Saat Anda memulihkan snapshot sementara, snapshot dipulihkan dari tingkat arsip ke tingkat standar, dan salinan snapshot tetap berada di tingkat arsip. Anda ditandaih untuk snapshot di tingkat standar saja.

Menghapus snapshot

Jika Anda menghapus snapshot saat sedang diarsipkan, Anda akan ditandaih untuk data snapshot yang telah dipindahkan ke tingkat arsip. Data ini tunduk pada periode arsip minimum 90 hari dan ditandaih sesuai pada saat penghapusan. Misalnya, jika Anda mengarsipkan snapshot 100 GiB, dan Anda menghapus snapshot setelah hanya 40 GiB diarsipkan, Anda ditandaih $1,50 untuk periode arsip minimum 90 hari untuk 40 GiB yang telah diarsipkan ($0,0125 per GB-bulan* 40 GB * (90 hari* 24 jam)/(24 jam/hari* 30 hari bulan).

Jika Anda menghapus snapshot saat sedang dipulihkan dari tingkat arsip, Anda akan dikenai biaya untuk pemulihan snapshot untuk ukuran penuh snapshot (ukuran snapshot * $0,03). Misalnya, jika Anda mengembalikan snapshot 100 GiB dari tingkat arsip, dan Anda menghapus snapshot kapan saja sebelum pemulihan snapshot selesai, Anda ditandaih $3 (ukuran snapshot 100 GiB* $0,03).

Keranjang Sampah

Snapshot yang diarsipkan dikenai biaya dengan tarif untuk snapshot yang diarsipkan saat berada di Keranjang Sampah. Snapshot yang diarsipkan yang ada di Keranjang Sampah tunduk pada periode arsip minimum 90 hari dan dikenai biaya sesuai jika dihapus oleh Keranjang Sampah sebelum periode arsip minimum. Dengan kata lain, jika aturan retensi menghapus snapshot yang diarsipkan dari Keranjang Sampah sebelum periode minimum 90 hari, Anda akan dikenai biaya untuk sisa hari.

Jika Anda menghapus snapshot yang cocok dengan aturan retensi saat snapshot sedang diarsipkan, snapshot yang diarsipkan akan disimpan di Keranjang Sampah untuk periode retensi yang ditentukan dalam aturan retensi. Itu dikenai biaya dengan tarif untuk snapshot yang diarsipkan.

Jika Anda menghapus snapshot yang cocok dengan aturan retensi saat snapshot dipulihkan, snapshot yang dipulihkan akan disimpan di Keranjang Sampah selama sisa periode retensi, dan dikenai biaya pada tingkat snapshot standar. Untuk menggunakan snapshot yang dipulihkan, Anda harus memulihkannya terlebih dahulu dari Keranjang Sampah.

Untuk informasi selengkapnya, lihat Recycle Bin.

Pelacakan biaya

Snapshot yang diarsipkan muncul di AWS Cost and Usage Report dengan ID sumber daya yang sama dan Nama Sumber Daya Amazon (ARN). Untuk informasi selengkapnya, silakan lihat Panduan Pengguna AWS Cost and Usage Report.

Anda dapat menggunakan tipe penggunaan berikut untuk mengidentifikasi biaya terkait:

- `SnapshotArchiveStorage` — biaya untuk penyimpanan data bulanan
- `SnapshotArchiveRetrieval` — biaya satu kali untuk pemulihan snapshot
- `SnapshotArchiveEarlyDelete` — biaya untuk menghapus atau memulihkan snapshot secara permanen sebelum periode arsip minimum (90 hari)

Kuota

Bagian ini menjelaskan kuota default untuk snapshot yang diarsipkan dan sedang berlangsung.

Kuota	Kuota default			
Snapshot yang diarsipkan per volume	25			
Arsip snapshot dalam proses bersamaan per akun	25			
Arsip snapshot dalam proses bersamaan	5			

Kuota	Kuota default			
per akun				

Jika Anda membutuhkan lebih dari batas default, lengkapi formulir AWS Support Center Create case untuk meminta peningkatan batas.

Panduan dan praktik terbaik untuk mengarsipkan snapshot

Bagian ini memberikan beberapa pedoman dan praktik terbaik untuk mengarsipkan snapshot.

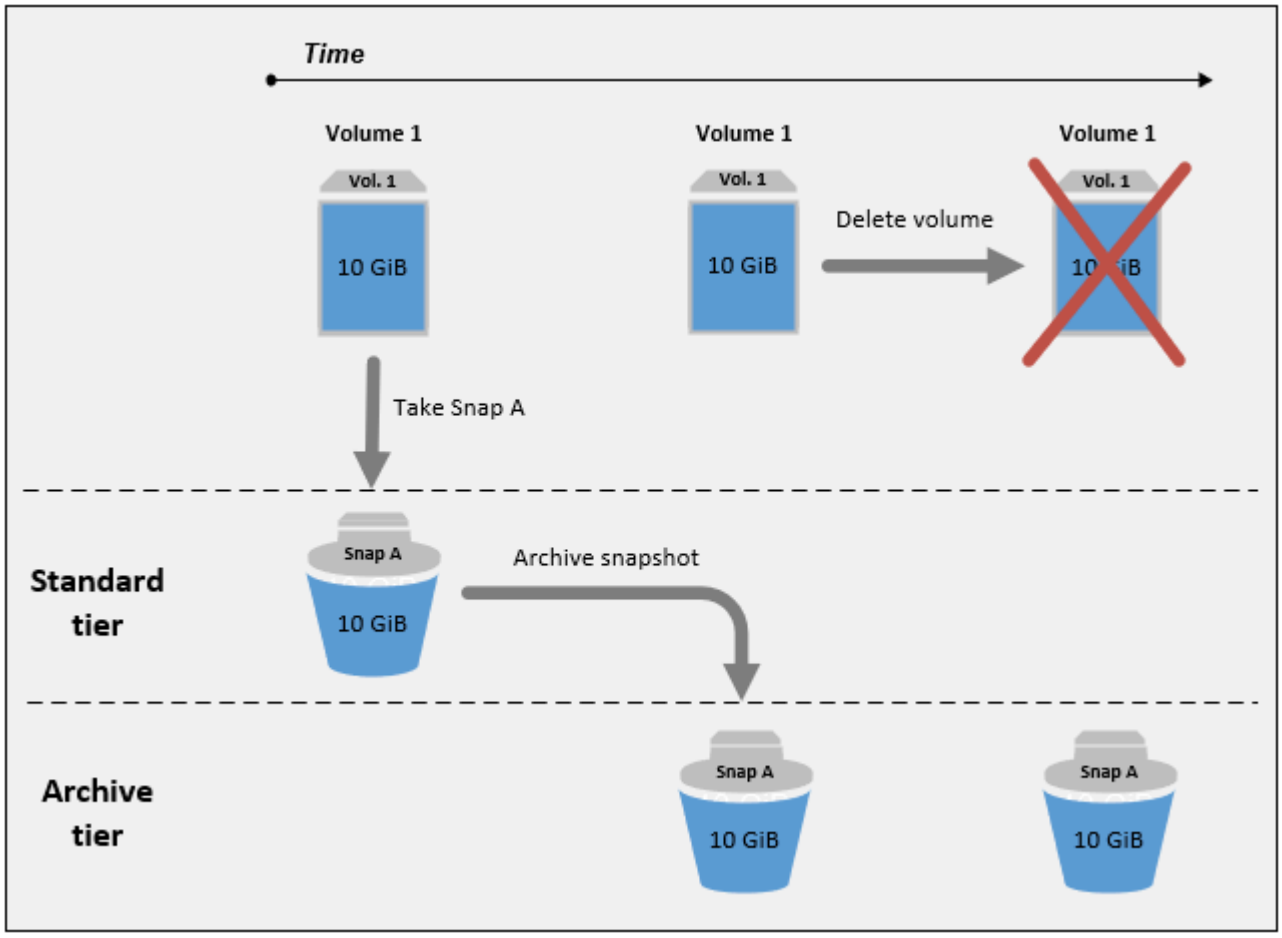
Topik

- Mengarsipkan satu-satunya snapshot volume
- Mengarsipkan snapshot tambahan dari satu volume
- Mengarsipkan snapshot lengkap untuk alasan kepatuhan
- Menentukan pengurangan biaya penyimpanan tingkat standar

Mengarsipkan satu-satunya snapshot volume

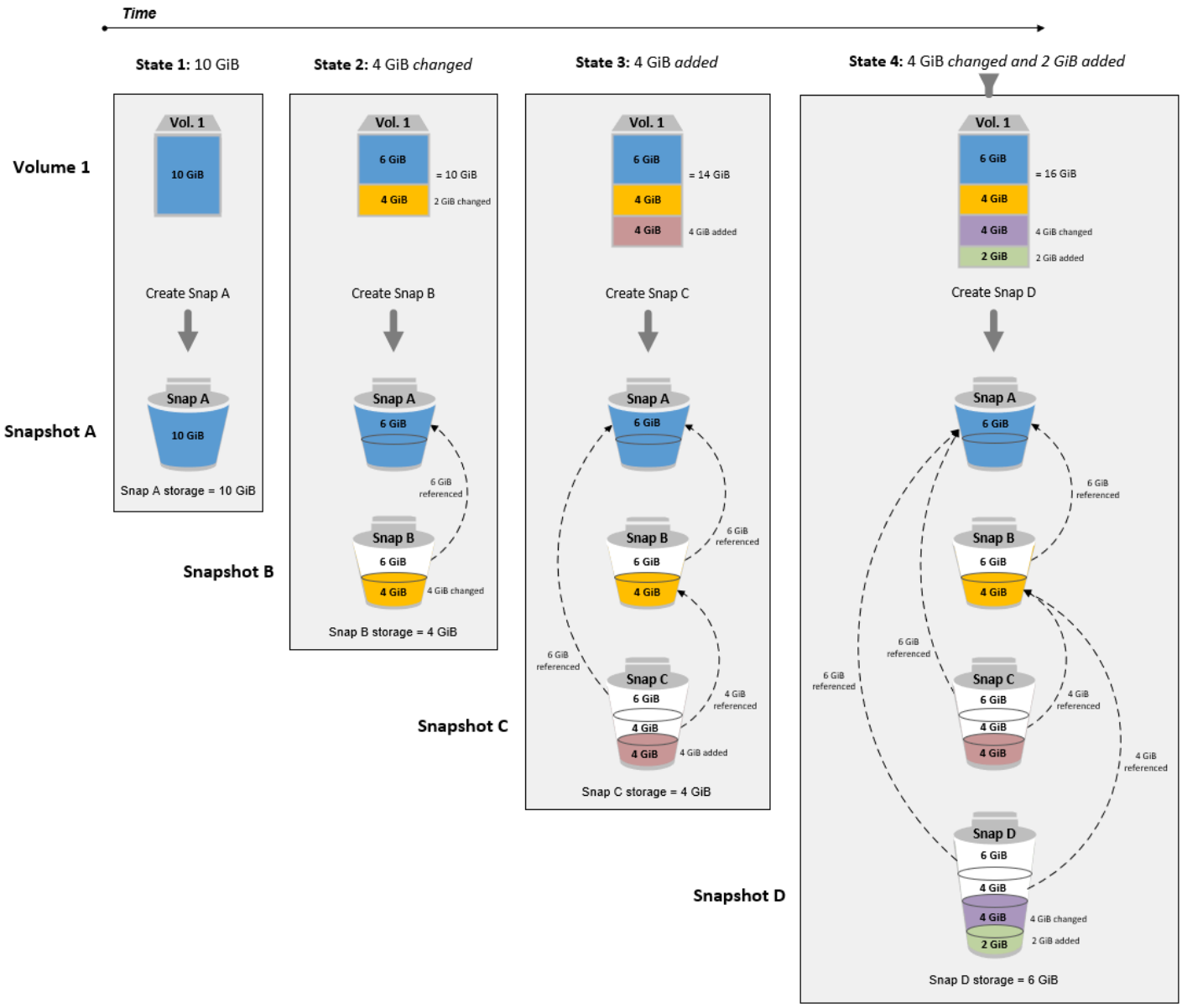
Bila Anda hanya memiliki satu snapshot volume, snapshot selalu berukuran sama dengan blok yang ditulis ke volume pada saat snapshot dibuat. Saat Anda mengarsipkan snapshot seperti itu, snapshot di tingkat standar dikonversi ke snapshot penuh berukuran setara dan dipindahkan dari tingkat standar ke tingkat arsip.

Mengarsipkan snapshot ini dapat membantu Anda menghemat dengan biaya penyimpanan yang lebih rendah. Jika Anda tidak lagi membutuhkan volume sumber, Anda dapat menghapus volume untuk penghematan biaya penyimpanan lebih lanjut.



Mengarsipkan snapshot tambahan dari satu volume

Saat Anda mengarsipkan snapshot inkremental, snapshot akan dikonversi ke snapshot penuh dan akan dipindahkan ke tingkat Arsip. Misalnya, pada gambar berikut, jika Anda mengarsipkan Snap B, snapshot dikonversi ke snapshot penuh yang berukuran 10 GiB dan dipindahkan ke tingkat arsip. Demikian pula, jika Anda mengarsipkan Snap C, ukuran snapshot lengkap di tingkat arsip adalah 14 GiB.



Jika Anda mengarsipkan snapshot untuk mengurangi biaya penyimpanan di tingkat standar, Anda tidak boleh mengarsipkan snapshot pertama dalam satu set snapshot tambahan. Snapshot ini direferensikan oleh snapshot berikutnya dalam garis keturunan snapshot. Dalam kebanyakan kasus, pengarsipan snapshot ini tidak akan mengurangi biaya penyimpanan.

> **Note**
>
> Anda tidak boleh mengarsipkan snapshot terakhir dalam satu set snapshot tambahan. Snapshot terakhir adalah snapshot terbaru yang diambil dari sebuah volume. Anda akan

memerlukan snapshot ini di tingkat standar jika Anda ingin membuat volume darinya jika terjadi korupsi atau kehilangan volume.

Jika Anda mengarsipkan snapshot yang berisi data yang direferensikan oleh snapshot berikutnya di garis keturunan, biaya penyimpanan dan penyimpanan data yang terkait dengan data yang direferensikan dialokasikan ke snapshot berikutnya dalam garis keturunan. Dalam kasus ini, pengarsipan snapshot tidak akan mengurangi biaya penyimpanan atau penyimpanan data. Misalnya, pada gambar sebelumnya, jika Anda mengarsipkan Snap B, 4 GiB datanya diatribusikan ke Snap C. Dalam hal ini, biaya penyimpanan Anda secara keseluruhan akan meningkat karena Anda dikenai biaya penyimpanan untuk versi lengkap Snap B di tingkat arsip, dan biaya penyimpanan Anda untuk tingkat standar tetap tidak berubah.

Jika Anda mengarsipkan Snap C, penyimpanan tingkat standar Anda akan berkurang sebesar 4 GiB karena data tidak direferensikan oleh snapshot lain nanti di garis keturunan. Dan penyimpanan tingkat arsip Anda akan meningkat sebesar 14 GiB karena snapshot dikonversi ke snapshot penuh.

Mengarsipkan snapshot lengkap untuk alasan kepatuhan

Anda mungkin perlu membuat cadangan volume penuh setiap bulan, triwulanan, atau tahunan untuk alasan kepatuhan. Untuk pencadangan ini, Anda mungkin memerlukan snapshot mandiri tanpa referensi mundur atau meneruskan ke snapshot lain di garis keturunan snapshot. Snapshot yang diarsipkan dengan EBS Snapshots Archive adalah snapshot lengkap, dan tidak memiliki referensi ke snapshot lain dalam garis keturunan. Selain itu, Anda mungkin perlu mempertahankan snapshot ini untuk alasan kepatuhan selama beberapa tahun. EBS Snapshots Archive membuatnya hemat biaya untuk mengarsipkan snapshot lengkap ini untuk retensi jangka panjang.

Menentukan pengurangan biaya penyimpanan tingkat standar

Jika Anda ingin mengarsipkan snapshot tambahan untuk mengurangi biaya penyimpanan Anda, Anda harus mempertimbangkan ukuran snapshot penuh di tingkat arsip dan pengurangan penyimpanan di tingkat standar. Bagian ini menjelaskan cara melakukannya.

Important

Respons API adalah data yang akurat pada point-in-time saat API dipanggil. Respons API dapat berbeda karena data yang terkait dengan snapshot berubah sebagai akibat dari perubahan garis keturunan snapshot.

Untuk menentukan pengurangan biaya penyimpanan dan penyimpanan di tingkat standar, gunakan langkah-langkah berikut.

1. Periksa ukuran snapshot lengkap. Untuk menentukan ukuran penuh snapshot, gunakan list-snapshot-blocksperintah. Untuk `--snapshot-id`, tentukan ID snapshot yang ingin Anda arsipkan.

```
$  aws ebs list-snapshot-blocks --snapshot-id snapshot_id
```

Ini mengembalikan informasi tentang semua blok dalam snapshot yang ditentukan. Blok terakhir `BlockIndex` yang dikembalikan oleh perintah menunjukkan jumlah blok dalam snapshot. Jumlah blok dikalikan dengan 512 KiB, yang merupakan ukuran blok snapshot, memberi Anda perkiraan yang dekat dengan ukuran snapshot penuh di tingkat arsip (blok * 512 KiB = ukuran snapshot penuh).

Misalnya, perintah berikut mencantumkan blok untuk snapshot `snap-01234567890abcdef`.

```
$  aws ebs list-snapshot-blocks --snapshot-id snap-01234567890abcdef
```

Berikut ini akan menunjukkan output perintah, dengan beberapa blok dihilangkan. Output berikut menunjukkan bahwa snapshot mencakup sekitar 16.383 blok data. Ini mendekati ukuran snapshot penuh sekitar 8 GiB (16.383 * 512 KiB = 7,99 GiB).

```
 {
    "VolumeSize": 8,
    "Blocks": [
        {
            "BlockToken": "ABgBAeShfa5RwG+RiWUg2pwmnCU/
YMnV7fGMxLbCWfEBEUmmuqac5RmoyVat",
            "BlockIndex": 0
        },
        {
            "BlockToken": "ABgBATdTONyThPUAbQhbUQXsn5TGoY/
J17GfE83j9WN7siupavOTw9E1KpFh",
            "BlockIndex": 1
        },
        {
            "BlockToken": "EBEUmmuqXsn5TGoY/QwmnCU/YMnV74eKE2TSsn5TGoY/
E83j9WQhbUQXsn5T",
            "BlockIndex": 4
```

```
        },
        .....
        {
            "BlockToken": "yThPUAbQhb5V8xpwmnCU/
YMnV74eKE2TSFY1sKP/4r05y47WETdTONyThPUA",
            "BlockIndex": 12890
        },
        {
            "BlockToken":
 "ABgBASHKD5V8xEbaRKdxdkZZS4eKE2TSFYlMG1sKP/4r05y47WEHqKaNPcLs",
            "BlockIndex": 12906
        },
        {
            "BlockToken": "ABgBARROGMUJo6P9X3CFHQGZNQ7av9B6vZtTTqV89QqC
+SkO0HWMlwkGXjnA",
            "BlockIndex": 16383
        }
    ],
    "VolumeSize": 8,
    "ExpiryTime": 1637677800.845,
    "BlockSize": 524288
}
```

2. Temukan volume sumber dari mana snapshot yang ingin Anda arsipkan dibuat. Gunakan perintah describe-snapshots. Untuk `--snapshot-id`, tentukan ID snapshot yang ingin Anda arsipkan. Parameter respons `VolumeId` menunjukkan ID volume sumber.

```
$ aws ec2 describe-snapshots --snapshot-id snapshot_id
```

Misalnya, perintah berikut memberikan informasi tentang snapshot `snap-09c9114207084f0d9`.

```
$ aws ec2 describe-snapshots --snapshot-id snap-09c9114207084f0d9
```

Berikut ini adalah output perintah, yang menunjukkan bahwa snapshot `snap-09c9114207084f0d9` dibuat dari volume `vol-0f3e2c292c52b85c3`.

```
{
    "Snapshots": [
        {
            "Description": "",
```

```
            "Tags": [],
            "Encrypted": false,
            VolumeId": "vol-0f3e2c292c52b85c3",
            "State": "completed",
            "VolumeSize": 8,
            "StartTime": "2021-11-16T08:29:49.840Z",
            "Progress": "100%",
            "OwnerId": "123456789012",
            "SnapshotId": "snap-09c9114207084f0d9"
        }
    ]
}
```

3. Temukan semua snapshot yang dibuat dari volume sumber. Gunakan perintah describe-snapshots. Tentukan filter `volume-id`, dan untuk nilai filter, tentukan ID volume dari langkah sebelumnya.

```
$ aws ec2 describe-snapshots --filters "Name=volume-id, Values=volume_id"
```

Misalnya, perintah berikut mengembalikan semua snapshot yang dibuat dari volume `vol-0f3e2c292c52b85c3`.

```
$ aws ec2 describe-snapshots --filters "Name=volume-id,
 Values=vol-0f3e2c292c52b85c3"
```

Berikut ini adalah output perintah, yang menunjukkan bahwa tiga snapshot dibuat dari volume `vol-0f3e2c292c52b85c3`.

```
{
    "Snapshots": [
        {
            "Description": "",
            "Tags": [],
            "Encrypted": false,
            "VolumeId": "vol-0f3e2c292c52b85c3",
            "State": "completed",
            "VolumeSize": 8,
            "StartTime": "2021-11-14T08:57:39.300Z",
            "Progress": "100%",
            "OwnerId": "123456789012",
            "SnapshotId": "snap-08ca60083f86816b0"
```

```
        },
        {
            "Description": "",
            "Tags": [],
            "Encrypted": false,
            "VolumeId": "vol-0f3e2c292c52b85c3",
            "State": "completed",
            "VolumeSize": 8,
            "StartTime": "2021-11-15T08:29:49.840Z",
            "Progress": "100%",
            "OwnerId": "123456789012",
            "SnapshotId": "snap-09c9114207084f0d9"
        },
        {
            "Description": "01",
            "Tags": [],
            "Encrypted": false,
            "VolumeId": "vol-0f3e2c292c52b85c3",
            "State": "completed",
            "VolumeSize": 8,
            "StartTime": "2021-11-16T07:50:08.042Z",
            "Progress": "100%",
            "OwnerId": "123456789012",
            "SnapshotId": "snap-024f49fe8dd853fa8"
        }
    ]
}
```

4. Menggunakan output dari perintah sebelumnya, urutkan snapshot berdasarkan waktu pembuatannya, dari yang paling awal hingga yang terbaru. Parameter `StartTime` respons untuk setiap snapshot menunjukkan waktu pembuatannya, dalam format waktu UTC.

   Misalnya, snapshot yang dikembalikan pada langkah sebelumnya yang diatur oleh waktu pembuatan, dari awal ke yang terbaru, adalah sebagai berikut:

   1. `snap-08ca60083f86816b0` (paling awal — dibuat sebelum snapshot yang ingin Anda arsipkan)
   2. `snap-09c9114207084f0d9` (snapshot untuk diarsipkan)
   3. `snap-024f49fe8dd853fa8` (terbaru — dibuat setelah snapshot yang ingin Anda arsipkan)

5. Identifikasi snapshot yang dibuat segera sebelum dan sesudah snapshot yang ingin Anda arsipkan. Dalam hal ini, Anda harus mengarsipkan snapshot `snap-09c9114207084f0d9`,

yang merupakan snapshot inkrementa; kedua yang dibuat dalam rangkaian tiga snapshot. Snapshot `snap-08ca60083f86816b0` dibuat segera sebelumnya, dan snapshot `snap-024f49fe8dd853fa8` dibuat segera setelahnya.

6. Temukan data yang tidak direferensikan dalam snapshot yang ingin Anda arsipkan. Pertama, temukan blok yang berbeda antara snapshot yang dibuat segera sebelum snapshot yang ingin Anda arsipkan, dan snapshot yang ingin Anda arsipkan. Gunakan perintah list-changed-blocks. Untuk `--first-snapshot-id`, tentukan ID snapshot yang dibuat segera sebelum snapshot yang ingin Anda arsipkan. Untuk `--second-snapshot-id`, tentukan ID snapshot yang ingin Anda arsipkan.

```
$ aws ebs list-changed-blocks --first-snapshot-id snapshot_created_before --second-snapshot-id snapshot_to_archive
```

Misalnya, perintah berikut menunjukkan indeks blok untuk blok yang berbeda antara snapshot `snap-08ca60083f86816b0` (snapshot yang dibuat sebelum snapshot yang ingin Anda arsipkan), dan snapshot `snap-09c9114207084f0d9` (snapshot yang ingin Anda arsipkan).

```
$ aws ebs list-changed-blocks --first-snapshot-id snap-08ca60083f86816b0 --second-snapshot-id snap-09c9114207084f0d9
```

Berikut ini akan menunjukkan output perintah, dengan beberapa blok dihilangkan.

```
{
    "BlockSize": 524288,
    "ChangedBlocks": [
        {
            "FirstBlockToken": "ABgBAX6y
+WH6Rm9y5zq1VyeTCmEzGmTT0jNZG1cDirFq1rOVeFbWXsH3W4z/",
            "SecondBlockToken": "ABgBASyx0bHHBnTERu
+9USLxYK/81UT0dbHIUFqUjQUkwTwK5qkjP8NSGyNB",
            "BlockIndex": 4
        },
        {
            "FirstBlockToken": "ABgBAcfL
+EfmQmlNgstqrFnYgsAxR4SDSO4LkNLYOOChGBWcfJnpn90E9XX1",
            "SecondBlockToken": "ABgBAdX0mtX6aBAt3EBy
+8jFCESMpig7csKjbO2Ocd08m2iNJV2Ue+cRwUqF",
            "BlockIndex": 5
        },
        {
```

```
            "FirstBlockToken": "ABgBAVBaFJmbP/eRHGh7vnJlAwyiyNUi3MKZmEMxs2wC3AmM/
fc6yCOAMb65",
            "SecondBlockToken":
 "ABgBAdewWkHKTcrhZmsfM7GbaHyXD1Ctcn2nppz4wYItZRmAo1M72fpXU0Yv",
            "BlockIndex": 13
        },
        {
            "FirstBlockToken": "ABgBAQGxwuf6z095L6DpRoVRVnOqPxmx9r7Wf6O+i
+ltZ0dwPpGN39ijztLn",
            "SecondBlockToken": "ABgBAUdlitCVI7c6hGsT4ckkKCw6bMRclnV
+bKjViu/9UESTcW7CD9w4J2td",
            "BlockIndex": 14
        },
        {
            "FirstBlockToken":
 "ABgBAZBfEv4EHS1aSXTXxSE3mBZG6CNeIkwxpljzmgSHICGlFmZCyJXzE4r3",
            "SecondBlockToken":
 "ABgBAVWR7QuQQB0AP2TtmNkgS4Aec5KAQVCldnpc91zBiNmSfW9ouIlbeXWy",
            "BlockIndex": 15
        },
        .....
        {
            "SecondBlockToken": "ABgBAeHwXPL+z3DBLjDhwjdAM9+CPGV5VO5Q3rEEA
+ku50P498hjnTAgMhLG",
            "BlockIndex": 13171
        },
        {
            "SecondBlockToken":
 "ABgBAbZcPiVtLx6U3Fb4lAjRdrkJMwW5M2tiCgIp6ZZpcZ8AwXxkjVUUHADq",
            "BlockIndex": 13172
        },
        {
            "SecondBlockToken": "ABgBAVmEd/pQ9VW9hWiOujOAKcauOnUFCO
+eZ5ASVdWLXWWC04ijfoDTpTVZ",
            "BlockIndex": 13173
        },
        {
            "SecondBlockToken": "ABgBAT/jeN7w
+8ALuNdaiwXmsSfM6tOvMoLBLJ14LKvavw4IiB1d0iykWe6b",
            "BlockIndex": 13174
        },
        {
            "SecondBlockToken": "ABgBAXtGvUhTjjUqkwKXfXzyR2GpQei/
+pJSG/19ESwvt7Hd8GHaUqVs6Zf3",
```

```
            "BlockIndex": 13175
        }
    ],
    "ExpiryTime": 1637648751.813,
    "VolumeSize": 8
}
```

Selanjutnya, gunakan perintah yang sama untuk menemukan blok yang berbeda antara snapshot yang ingin Anda arsipkan dan snapshot yang dibuat segera setelahnya. Untuk `--first-snapshot-id`, tentukan ID snapshot yang ingin Anda arsipkan. Untuk `--second-snapshot-id`, tentukan ID snapshot yang dibuat segera sebelum snapshot yang ingin Anda arsipkan.

```
$ aws ebs list-changed-blocks --first-snapshot-id snapshot_to_archive --second-
snapshot-id snapshot_created_after
```

Misalnya, perintah berikut menunjukkan indeks blok untuk blok yang berbeda antara snapshot `snap-09c9114207084f0d9` (snapshot yang dibuat sebelum snapshot yang ingin Anda arsipkan), dan snapshot `snap-024f49fe8dd853fa8` (snapshot yang ingin Anda arsipkan).

```
$ aws ebs list-changed-blocks --first-snapshot-id snap-09c9114207084f0d9 --second-
snapshot-id snap-024f49fe8dd853fa8
```

Berikut ini akan menunjukkan output perintah, dengan beberapa blok dihilangkan.

```
{
    "BlockSize": 524288,
    "ChangedBlocks": [
        {
            "FirstBlockToken": "ABgBAVax0bHHBnTERu
+9USLxYK/81UT0dbSnkDk0gqwRFSFGWA7HYbkkAy5Y",
            "SecondBlockToken":
 "ABgBASEvi9x8Om7Htp37cKG2NT9XUzEbLHpGcayelomSoHpGy8LGyvG0yYfK",
            "BlockIndex": 4
        },
        {
            "FirstBlockToken": "ABgBAeL0mtX6aBAt3EBy+8jFCESMpig7csfMrI4ufnQJT3XBm/
pwJZ1n2Uec",
```

```
            "SecondBlockToken": "ABgBAXmUTg6rAI
+v0LvekshbxCVpJjWILvxgC0AG0GQBEUNRVHkNABBwXLkO",
            "BlockIndex": 5
        },
        {
            "FirstBlockToken":
 "ABgBATKwWkHKTcrhZmsfM7GbaHyXD1CtcnjIZv9YzisYsQTMHfTfh4AhS0s2",
            "SecondBlockToken": "ABgBAcmiPFovWgXQio
+VBrxOqGy4PKZ9SAAHaZ2HQBM9fQQU0+EXxQjVGv37",
            "BlockIndex": 13
        },
        {
            "FirstBlockToken":
 "ABgBAbRlitCVI7c6hGsT4ckkKCw6bMRclnARrMt1hUbIhFnfz8kmUaZOP2ZE",
            "SecondBlockToken": "ABgBAXe935n544+rxhJ0INB8q7pAeoPZkkD27vkspE/
qKyvOwpozYII6UNCT",
            "BlockIndex": 14
        },
        {
            "FirstBlockToken": "ABgBAd+yxCO26I
+1Nm2KmuKfrhjCkuaP6LXuol3opCNk6+XRGcct4suBHje1",
            "SecondBlockToken": "ABgBAcPpnXz821NtTvWBPTz8uUFXnS8jXubvghEjZulIjHgc
+7saWys77shb",
            "BlockIndex": 18
        },
        .....
        {
            "SecondBlockToken": "ABgBATni4sDE5rS8/a9pqV03lU/lKCW
+CTxFl3cQ5p2f2h1njpuUiGbqKGUa",
            "BlockIndex": 13190
        },
        {
            "SecondBlockToken": "ABgBARbXo7zFhu7IEQ/9VMYFCTCtCuQ
+iSlWVpBIshmeyeS5FD/M0i64U+a9",
            "BlockIndex": 13191
        },
        {
            "SecondBlockToken": "ABgBAZ8DhMk+rROXa4dZlNK45rMYnVIGGSyTeiMli/sp/
JXUVZKJ9sMKIsGF",
            "BlockIndex": 13192
        },
        {
            "SecondBlockToken":
 "ABgBATh6MBVE904l6sqOC27s1nVntFUpDwiMcRWGyJHy8sIgGL5yuYXHAVty",
```

```
            "BlockIndex": 13193
        },
        {
            "SecondBlockToken":
 "ABgBARuZykaFBWpCWrJPXaPCneQMbyVgnITJqj4c1kJWPIj5Gn61OQyy+giN",
            "BlockIndex": 13194
        }
    ],
    "ExpiryTime": 1637692677.286,
    "VolumeSize": 8
}
```

7. Bandingkan output yang dikembalikan oleh kedua perintah pada langkah sebelumnya. Jika indeks blok yang sama muncul di kedua output perintah, ini menunjukkan bahwa blok berisi data yang tidak direferensikan.

   Misalnya, output perintah pada langkah sebelumnya menunjukkan bahwa blok 4, 5, 13, dan 14 unik untuk snapshot `snap-09c9114207084f0d9` dan bahwa mereka tidak direferensikan oleh snapshot lain dalam garis keturunan snapshot.

   Untuk menentukan pengurangan penyimpanan tingkat standar, kalikan jumlah blok yang muncul di kedua output perintah dengan 512 KiB, yang merupakan ukuran blok snapshot.

   Misalnya, jika 9.950 indeks blok muncul di kedua output perintah, ini menunjukkan bahwa Anda akan mengurangi penyimpanan tingkat standar sekitar 4,85 GiB (9.950 blok * 512 KiB = 4,85 GiB).

8. Tentukan biaya penyimpanan untuk menyimpan blok yang tidak direferensikan di tingkat standar selama 90 hari. Bandingkan nilai ini dengan biaya penyimpanan snapshot penuh, dijelaskan dari langkah 1, di tingkat arsip. Anda dapat menentukan penghematan biaya dengan membandingkan nilainya, dengan asumsi bahwa Anda tidak memulihkan snapshot penuh dari tingkat arsip selama periode minimum 90 hari. Untuk informasi selengkapnya, lihat Harga dan penagihan.

Izin IAM yang diperlukan

Secara default, pengguna tidak memiliki izin untuk menggunakan pengarsipan snapshot. Untuk mengizinkan pengguna bekerja dengan sumber daya ini, Anda harus membuat kebijakan IAM yang memberikan izin menggunakan sumber daya dan tindakan API tertentu. Untuk informasi selengkapnya, lihat Membuat kebijakan IAM dalam Panduan Pengguna IAM.

Untuk menggunakan pengarsipan snapshot, pengguna memerlukan izin berikut.

- `ec2:DescribeSnapshotTierStatus`
- `ec2:ModifySnapshotTier`
- `ec2:RestoreSnapshotTier`

Pengguna konsol mungkin memerlukan izin tambahan seperti `ec2:DescribeSnapshots`.

Untuk mengarsipkan dan memulihkan snapshot terenkripsi, AWS KMS izin tambahan berikut diperlukan.

- `kms:CreateGrant`
- `kms:Decrypt`
- `kms:DescribeKey`

Berikut ini adalah contoh kebijakan IAM yang memberikan izin kepada pengguna IAM untuk mengarsipkan, memulihkan, serta melihat snapshot terenkripsi dan tidak terenkripsi. Ini termasuk izin `ec2:DescribeSnapshots` untuk pengguna konsol. Jika beberapa izin tidak diperlukan, Anda dapat menghapusnya dari kebijakan.

> **Tip**
>
> Untuk mengikuti prinsip hak akses paling rendah, jangan biarkan akses penuh ke `kms:CreateGrant`. Sebagai gantinya, gunakan tombol `kms:GrantIsForAWSResource` kondisi untuk memungkinkan pengguna membuat hibah pada kunci KMS hanya ketika hibah dibuat atas nama pengguna oleh AWS layanan, seperti yang ditunjukkan pada contoh berikut.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "ec2:DescribeSnapshotTierStatus",
            "ec2:ModifySnapshotTier",
            "ec2:RestoreSnapshotTier",
```

```
            "ec2:DescribeSnapshots",
            "kms:CreateGrant",
            "kms:Decrypt",
            "kms:DescribeKey"
        ],
        "Resource": "*",
        "Condition": {
                "Bool": {
                    "kms:GrantIsForAWSResource": true
                }
            }
    }]
}
```

Untuk memberikan akses, tambahkan izin ke pengguna, grup, atau peran Anda:

- Pengguna dan grup di AWS IAM Identity Center:

  Buat rangkaian izin. Ikuti petunjuk dalam Buat set izin dalam Panduan Pengguna AWS IAM Identity Center .
- Pengguna yang dikelola di IAM melalui penyedia identitas:

  Buat peran untuk federasi identitas. Ikuti petunjuk dalam Membuat peran untuk penyedia identitas pihak ketiga (federasi) di Panduan Pengguna IAM.
- Pengguna IAM:
  - Buat peran yang dapat diambil pengguna Anda. Ikuti petunjuk dalam Membuat peran untuk pengguna IAM dalam Panduan Pengguna IAM.
  - (Tidak disarankan) Pasang kebijakan langsung ke pengguna atau tambahkan pengguna ke grup pengguna. Ikuti petunjuk dalam Menambahkan izin ke pengguna (konsol) dalam Panduan Pengguna IAM.

Bekerja dengan pengarsipan snapshot

Topik

- Mengarsipkan snapshot AMI
- Memulihkan snapshot yang diarsipkan
- Modifikasi periode pemulihan atau jenis pemulihan untuk snapshot yang dipulihkan sementara
- Lihat snapshot yang diarsipkan

Mengarsipkan snapshot AMI

Anda dapat mengarsipkan snapshot apa pun yang ada dalam status `completed` dan yang Anda miliki di akun Anda. Anda tidak dapat mengarsipkan snapshot yang ada di status `pending` atau `error`, atau snapshot yang dibagikan dengan Anda. Untuk informasi selengkapnya, lihat Pertimbangan dan batasan.

Jika snapshot dikaitkan dengan satu atau lebih AMI, Anda harus menonaktifkan AMI terkait terlebih dahulu sebelum Anda dapat mengarsipkan snapshot. Untuk informasi selengkapnya, lihat Menonaktifkan AMI.

Snapshot yang diarsipkan mempertahankan ID snapshot, status enkripsi, izin AWS Identity and Access Management (IAM), informasi pemilik, dan tag sumber daya. Namun, pemulihan snapshot cepat dan berbagi snapshot dinonaktifkan secara otomatis setelah snapshot diarsipkan.

Anda dapat terus menggunakan snapshot saat arsip sedang dalam proses. Segera setelah status tingkat snapshot mencapai status `archival-complete`, Anda tidak dapat lagi menggunakan snapshot.

Anda dapat mengarsipkan snapshot menggunakan salah satu metode berikut.

Console

Untuk mengarsipkan snapshot

Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.

1. Di panel navigasi, pilih Snapshot.
2. Dalam daftar snapshot, pilih snapshot yang akan diarsipkan, kemudian pilih Tindakan, Arsipkan snapshot.
3. Untuk mengonfirmasi, pilih Pulihkan snapshot.

AWS CLI

Untuk mengarsipkan snapshot

Gunakan modify-snapshot-tier AWS CLI perintah. Untuk `--snapshot-id`, tentukan ID snapshot yang ingin Anda arsipkan. Untuk `--storage-tier`, tentukan `archive`.

```
$ aws ec2 modify-snapshot-tier \
```

```
--snapshot-id snapshot_id \
--storage-tier archive
```

Misalnya, perintah berikut mengarsipkan snapshot `snap-01234567890abcedf`.

```
$ aws ec2 modify-snapshot-tier \
--snapshot-id snap-01234567890abcedf \
--storage-tier archive
```

Berikut adalah output perintahnya. Parameter respons `TieringStartTime` menunjukkan tanggal dan waktu proses arsip dimulai, dalam format waktu UTC (YYY-MM-DDTHH:MM:SSZ).

```
{
    "SnapshotId": "snap-01234567890abcedf",
    "TieringStartTime": "2021-09-15T16:44:37.574Z"
}
```

Memulihkan snapshot yang diarsipkan

Sebelum dapat menggunakan snapshot yang diarsipkan, Anda harus terlebih dahulu memulihkannya ke tingkat standar. Snapshot yang dipulihkan memiliki ID snapshot, status enkripsi, izin IAM, informasi pemilik, dan tanda sumber daya yang sama yang dimilikinya sebelum diarsipkan. Anda dapat menggunakan AMI yang dipulihkan dengan cara yang sama seperti Anda menggunakan AMI lainnya di akun Anda. Snapshot yang dipulihkan selalu merupakan snapshot penuh.

Saat memulihkan snapshot, Anda dapat memilih untuk memulihkannya secara permanen atau sementara.

Jika Anda memulihkan snapshot secara permanen, snapshot dipindahkan dari tingkat arsip ke tingkat standar secara permanen. Snapshot tetap dipulihkan dan siap digunakan sampai Anda mengarsipkan ulang secara manual atau Anda menghapusnya secara manual. Saat Anda memulihkan snapshot secara permanen, snapshot dihapus dari tingkat arsip.

Jika Anda memulihkan snapshot sementara, snapshot disalin dari tingkat arsip ke tingkat standar untuk periode pemulihan yang Anda tentukan. Snapshot tetap dipulihkan dan siap digunakan hanya untuk periode pemulihan. Selama periode pemulihan, salinan snapshot tetap berada di tingkat arsip. Setelah periode berakhir, snapshot secara otomatis dihapus dari tingkat standar. Anda dapat menambah atau mengurangi periode pemulihan atau mengubah tipe pemulihan menjadi permanen

kapan saja selama periode pemulihan. Untuk informasi selengkapnya, lihat Modifikasi periode pemulihan atau jenis pemulihan untuk snapshot yang dipulihkan sementara.

Jika Anda memulihkan snapshot yang terkait dengan AMI yang dinonaktifkan, dan Anda bermaksud menggunakan AMI itu, Anda harus terlebih dahulu memulihkan semua snapshot terkait secara permanen dan kemudian mengaktifkan kembali AMI yang dinonaktifkan sebelum Anda dapat menggunakannya. Anda tidak dapat mengaktifkan AMI jika snapshot terkait dipulihkan sementara. Anda dapat menggunakan perintah berikut untuk menemukan semua snapshot yang terkait dengan AMI.

```
$ C:\> aws ec2 describe-images --image-id ami_id \
  --query Images[*].BlockDeviceMappings[*].Ebs[].SnapshotId[]
```

Anda dapat memulihkan snapshot yang diarsipkan menggunakan salah satu metode berikut.

Console

Untuk memulihkan snapshot dari arsip

Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.

1. Di panel navigasi, pilih Snapshot.
2. Dalam daftar snapshot, pilih snapshot yang akan diarsipkan, kemudian pilih Tindakan, Pulihkan snapshot dari arsip.
3. Tentukan jenis pemulihan yang akan dilakukan. Untuk Jenis pemulihan, lakukan salah satu langkah berikut:
   - Untuk memulihkan snapshot secara permanen, pilih Permanen.
   - Untuk memulihkan snapshot secara sementara, pilih Sementara, kemudian untuk Periode pemulihan sementara, masukkan jumlah hari untuk mengembalikan snapshot.
4. Untuk mengonfirmasi, pilih Pulihkan snapshot.

AWS CLI

Untuk memulihkan snapshot yang diarsipkan secara permanen

Gunakan restore-snapshot-tier AWS CLI perintah. Untuk `--snapshot-id`, tentukan ID snapshot yang akan dipulihkan, dan sertakan opsi `--permanent-restore`.

```
$ aws ec2 restore-snapshot-tier \
--snapshot-id snapshot_id \
--permanent-restore
```

Misalnya, perintah berikut memulihkan snapshot `snap-01234567890abcedf` secara permanen.

```
$ aws ec2 restore-snapshot-tier \
--snapshot-id snap-01234567890abcedf \
--permanent-restore
```

Berikut adalah output perintahnya.

```
{
    "SnapshotId": "snap-01234567890abcedf",
    "IsPermanentRestore": true
}
```

Untuk memulihkan snapshot yang diarsipkan sementara

Gunakan restore-snapshot-tier AWS CLI perintah. Abaikan `--permanent-restore` opsi. Untuk `--snapshot-id`, tentukan ID snapshot yang akan dipulihkan, dan untuk `--temporary-restore-days`, tentukan jumlah hari untuk memulihkan snapshot.

`--temporary-restore-days` harus ditentukan dalam beberapa hari. Rentang yang diizinkan adalah 1 - 180. Jika Anda tidak menentukan nilai, secara otomatis nilainya adalah 1 hari.

```
$ aws ec2 restore-snapshot-tier \
--snapshot-id snapshot_id \
--temporary-restore-days number_of_days
```

Misalnya, perintah berikut memulihkan snapshot `snap-01234567890abcedf` sementara untuk periode pemulihan 5 hari.

```
$ aws ec2 restore-snapshot-tier \
--snapshot-id snap-01234567890abcedf \
--temporary-restore-days 5
```

Berikut adalah output perintahnya.

```
{
```

```
    "SnapshotId": "snap-01234567890abcedf",
    "RestoreDuration": 5,
    "IsPermanentRestore": false
}
```

Modifikasi periode pemulihan atau jenis pemulihan untuk snapshot yang dipulihkan sementara

Saat Anda memulihkan snapshot sementara, Anda harus menentukan jumlah hari di mana snapshot akan tetap dipulihkan di akun Anda. Setelah periode kedaluwasa, snapshot secara otomatis dihapus dari tingkat standar.

Anda dapat mengubah periode pemulihan untuk snapshot yang dipulihkan sementara kapan saja.

Anda dapat memilih untuk menambah atau mengurangi periode pemulihan, atau Anda dapat mengubah jenis pemulihan dari sementara menjadi permanen.

Jika Anda mengubah periode pemulihan, periode pemulihan baru berlaku sejak tanggal saat ini. Misalnya, jika Anda menentukan periode pemulihan baru 5 hari, snapshot akan tetap dipulihkan selama lima hari dari tanggal saat ini.

> **Note**
>
> Anda dapat mengakhiri pemulihan sementara lebih awal dengan mengatur periode pemulihan menjadi 1 hari.

Jika Anda mengubah jenis pemulihan dari sementara ke permanen, salinan snapshot dihapus dari tingkat arsip, dan snapshot tetap tersedia di akun Anda sampai Anda mengarsipkan ulang atau menghapusnya secara manual.

Anda dapat memodifikasi periode pemulihan untuk snapshot menggunakan salah satu metode berikut.

Console

Untuk memodifikasi periode pemulihan atau jenis pemulihan

Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.

1. Di panel navigasi, pilih Snapshot.

2. Dalam daftar snapshot, pilih snapshot yang sebelumnya Anda pulihkan sementara, kemudian pilih Tindakan, Pulihkan snapshot dari arsip.
3. Untuk Jenis pemulihan, lakukan salah satu langkah berikut:
   - Untuk mengubah jenis pemulihan dari sementara ke permanen, pilih Permanen.
   - Untuk menambah atau mengurangi periode pemulihan, tetap pilih Sementara, dan kemudian untuk Periode pemulihan sementara, masukkan periode pemulihan baru dalam beberapa hari.
4. Untuk mengonfirmasi, pilih Pulihkan snapshot.

AWS CLI

Untuk memodifikasi periode pemulihan atau jenis pemulihan

Gunakan restore-snapshot-tier AWS CLI perintah. Untuk `--snapshot-id`, tentukan ID snapshot yang sebelumnya Anda pulihkan sementara. Untuk mengubah jenis pemulihan dari sementara ke permanen, tentukan `--permanent-restore` dan hilangkan `--temporary-restore-days`. Untuk menambah atau mengurangi periode pemulihan, hilangkan `--permanent-restore` dan untuk `--temporary-restore-days`, tentukan periode pemulihan baru dalam hitungan hari.

Contoh: Menambah atau mengurangi periode pemulihan

Perintah berikut mengubah periode pemulihan untuk snapshot `snap-01234567890abcedf` menjadi `10` hari.

```
$ aws ec2 restore-snapshot-tier \
--snapshot-id snap-01234567890abcedf
--temporary-restore-days 10
```

Berikut adalah output perintahnya.

```
{
    "SnapshotId": "snap-01234567890abcedf",
    "RestoreDuration": 10,
    "IsPermanentRestore": false
}
```

Contoh: Ubah jenis pemulihan menjadi permanen

Perintah berikut mengubah tipe pemulihan untuk snapshot `snap-01234567890abcedf` dari sementara menjadi permanen.

```
$ aws ec2 restore-snapshot-tier \
--snapshot-id snap-01234567890abcedf
--permanent-restore
```

Berikut adalah output perintahnya.

```
{
    "SnapshotId": "snap-01234567890abcedf",
    "IsPermanentRestore": true
}
```

Lihat snapshot yang diarsipkan

Anda dapat melihat informasi tingkat penyimpanan untuk snapshot menggunakan salah satu metode berikut.

Console

Untuk melihat informasi tingkat penyimpanan untuk snapshot

Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.

1. Di panel navigasi, pilih Snapshot.
2. Dalam daftar snapshot, pilih snapshot dan pilih tab Tingkat penyimpanan.

   Memberikan informasi berikut ini:

   - Perubahan tingkat terakhir dimulai pada — Tanggal dan waktu ketika arsip atau pemulihan terakhir dimulai.
   - Kemajuan perubahan tingkat - Kemajuan arsip terakhir atau tindakan pemulihan, sebagai persentase.
   - Tingkat penyimpanan — Tingkat penyimpanan untuk snapshot. Selalu `archive` untuk snapshot yang diarsipkan, dan `standard` untuk snapshot yang disimpan di tingkat standar, termasuk snapshot yang dipulihkan sementara.
   - Status tingkatan — Status arsip terakhir atau tindakan pemulihan.

- Arsip selesai pada — Tanggal dan waktu ketika arsip selesai.
- Pemulihan sementara kedaluwarsa pada — Tanggal dan waktu ketika snapshot yang dipulihkan sementara akan kedaluwarsa.

AWS CLI

Untuk melihat informasi pengarsipan tentang snapshot yang diarsipkan

Gunakan describe-snapshot-tier-status AWS CLI perintah. Tentukan filter `snapshot-id`, dan untuk nilai filter, tentukan ID snapshot. Atau, untuk melihat semua snapshot yang diarsipkan, hilangkan filter.

```
$ aws ec2 describe-snapshot-tier-status --filters "Name=snapshot-id,
 Values=snapshot_id"
```

Output mencakup parameter respons berikut:

- `Status` — Status snapshot. Selalu `completed` untuk snapshot yang diarsipkan. Hanya snapshot yang ada dalam status `completed` yang dapat diarsipkan.
- `LastTieringStartTime` — Tanggal dan waktu proses pengarsipan dimulai, dalam format waktu UTC (YYYY-MM-DDTHH:MM:SSZ).
- `LastTieringOperationState` — Status proses arsip saat ini. Status yang mungkin termasuk: `archival-in-progress` `archival-completed` `archival-failed` | `permanent-restore-in-progress` | `permanent-restore-completed` | | `permanent-restore-failed` | `temporary-restore-in-progress` | `temporary-restore-completed` | `temporary-restore-failed`
- `LastTieringProgress` — Kemajuan proses arsip snapshot, dalam persentase.
- `StorageTier` — Tingkat penyimpanan untuk snapshot. Selalu `archive` untuk snapshot yang diarsipkan, dan `standard` untuk snapshot yang disimpan di tingkat standar, termasuk snapshot yang dipulihkan sementara.
- `ArchivalCompleteTime` — Tanggal dan waktu proses pengarsipan selesai, dalam format waktu UTC (YYYY-MM-DDTHH:MM:SSZ).

Contoh

Perintah berikut menampilkan informasi tentang snapshot `snap-01234567890abcedf`.

```
$ aws ec2 describe-snapshot-tier-status --filters "Name=snapshot-id,
 Values=snap-01234567890abcedf"
```

Berikut adalah output perintahnya.

```
{
    "SnapshotTierStatuses": [
        {
            "Status": "completed",
            "ArchivalCompleteTime": "2021-09-15T17:33:16.147Z",
            "LastTieringProgress": 100,
            "Tags": [],
            "VolumeId": "vol-01234567890abcedf",
            "LastTieringOperationState": "archival-completed",
            "StorageTier": "archive",
            "OwnerId": "123456789012",
            "SnapshotId": "snap-01234567890abcedf",
            "LastTieringStartTime": "2021-09-15T16:44:37.574Z"
        }
    ]
}
```

Untuk melihat snapshot yang diarsipkan dan tingkat standar

Gunakan perintah AWS CLI describe-snapshots. Untuk `--snapshot-ids`, tentukan ID tampilan snapshot.

```
$ aws ec2 describe-snapshots --snapshot-ids snapshot_id
```

Misalnya, perintah berikut memberikan informasi tentang snapshot `snap-01234567890abcedf`.

```
$ aws ec2 describe-snapshots --snapshot-ids snap-01234567890abcedf
```

Berikut adalah output perintahnya. Parameter respons `StorageTier` menunjukkan apakah snapshot saat ini diarsipkan. `archive` menunjukkan bahwa snapshot saat ini diarsipkan dan disimpan di tingkat arsip, serta `standard` menunjukkan bahwa snapshot saat ini tidak diarsipkan dan disimpan di tingkat standar.

Dalam contoh output berikut, hanya `Snap A` yang diarsipkan. `Snap B` dan `Snap C` tidak diarsipkan.

Selain itu, parameter respons `RestoreExpiryTime` dikembalikan hanya untuk snapshot yang dipulihkan sementara dari arsip. Hal ini menunjukkan waktu snapshot yang dipulihkan sementara akan dihapus secara otomatis dari tingkat standar. Parameter respons tersebut tidak dikembalikan untuk snapshot yang dipulihkan secara permanen.

Dalam contoh output berikut, `Snap C` dipulihkan sementara, dan akan secara otomatis dihapus dari tingkat standar pada 2021-09-19T21:00:00.000Z (19 September 2021, pukul 21:00 UTC).


```
{
    "Snapshots": [
        {
            "Description": "Snap A",
            "Encrypted": false,
            "VolumeId": "vol-01234567890aaaaaa",
            "State": "completed",
            "VolumeSize": 8,
            "StartTime": "2021-09-07T21:00:00.000Z",
            "Progress": "100%",
            "OwnerId": "123456789012",
            "SnapshotId": "snap-01234567890aaaaaa",
            "StorageTier": "archive",
            "Tags": []
        },
        {
            "Description": "Snap B",
            "Encrypted": false,
            "VolumeId": "vol-09876543210bbbbbb",
            "State": "completed",
            "VolumeSize": 10,
            "StartTime": "2021-09-14T21:00:00.000Z",
            "Progress": "100%",
            "OwnerId": "123456789012",
            "SnapshotId": "snap-09876543210bbbbbb",
            "StorageTier": "standard",
            "RestoreExpiryTime": "2019-09-19T21:00:00.000Z",
            "Tags": []
        },
        {
            "Description": "Snap C",
            "Encrypted": false,
            "VolumeId": "vol-054321543210cccccc",
            "State": "completed",
            "VolumeSize": 12,
```

```
            "StartTime": "2021-08-01T21:00:00.000Z",
            "Progress": "100%",
            "OwnerId": "123456789012",
            "SnapshotId": "snap-054321543210cccccc",
            "StorageTier": "standard",
            "Tags": []
        }
    ]
}
```

Untuk hanya melihat snapshot yang disimpan di tingkat arsip atau tingkat standar

Gunakan perintah deskripsi-snapshot AWS CLI . Sertakan opsi `--filter`, untuk nama filter, tentukan `storage-tier`, dan untuk nilai filter tentukan antara `archive` atau `standard`.

```
$ aws ec2 describe-snapshots --filters "Name=storage-tier,Values=archive|standard"
```

Misalnya, perintah berikut menampilkan snapshot yang diarsipkan saja.

```
$ aws ec2 describe-snapshots --filters "Name=storage-tier,Values=archive"
```

Memantau pengarsipan snapshot

Amazon EBS memancarkan peristiwa yang terkait dengan tindakan pengarsipan snapshot. Anda dapat menggunakan AWS Lambda dan CloudWatch Acara Amazon untuk menangani pemberitahuan acara secara terprogram. Peristiwa dipancarkan atas dasar upaya terbaik. Untuk informasi selengkapnya, lihat Panduan Pengguna CloudWatch Acara Amazon.

Peristiwa berikut ini tersedia:

- `archiveSnapshot` — Dipancarkan ketika tindakan pengarsipan snapshot berhasil atau gagal.

  Berikut ini adalah contoh peristiwa yang dipancarkan saat tindakan arsip snapshot berhasil.

```
{
   "version": "0",
   "id": "01234567-0123-0123-0123-012345678901",
   "detail-type": "EBS Snapshot Notification",
   "source": "aws.ec2",
   "account": "123456789012",
```

```
   "time": "2021-05-25T13:12:22Z",
   "region": "us-east-1",
   "resources": [
     "arn:aws:ec2:us-east-1::snapshot/snap-01234567890abcdef"
   ],
   "detail": {
     "event": "archiveSnapshot",
     "result": "succeeded",
     "cause": "",
     "request-id": "123456789",
     "snapshot_id": "arn:aws:ec2:us-east-1::snapshot/snap-01234567890abcdef",
     "startTime": "2021-05-25T13:12:22Z",
     "endTime": "2021-05-45T15:30:00Z",
     "recycleBinExitTime": "2021-10-45T15:30:00Z"
  }
```

Berikut ini adalah contoh peristiwa yang dipancarkan saat tindakan arsip snapshot gagal.

```
{
   "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "EBS Snapshot Notification",
  "source": "aws.ec2",
  "account": "123456789012",
  "time": "2021-05-25T13:12:22Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2:us-east-1::snapshot/snap-01234567890abcdef"
  ],
  "detail": {
    "event": "archiveSnapshot",
    "result": "failed",
    "cause": "Source snapshot ID is not valid",
    "request-id": "1234567890",
    "snapshot_id": "arn:aws:ec2:us-east-1::snapshot/snap-01234567890abcdef",
    "startTime": "2021-05-25T13:12:22Z",
    "endTime": "2021-05-45T15:30:00Z",
    "recycleBinExitTime": "2021-10-45T15:30:00Z"
  }
}
```

- `permanentRestoreSnapshot` — Dipancarkan ketika tindakan pemulihan permanen berhasil atau gagal.

Berikut ini adalah contoh peristiwa yang dipancarkan saat tindakan pemulihan permanen berhasil.

```
{
   "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "EBS Snapshot Notification",
  "source": "aws.ec2",
  "account": "123456789012",
  "time": "2021-05-25T13:12:22Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2:us-east-1::snapshot/snap-01234567890abcdef"
  ],
  "detail": {
    "event": "permanentRestoreSnapshot",
    "result": "succeeded",
    "cause": "",
    "request-id": "1234567890",
    "snapshot_id": "arn:aws:ec2:us-east-1::snapshot/snap-01234567890abcdef",
    "startTime": "2021-05-25T13:12:22Z",
    "endTime": "2021-10-45T15:30:00Z"
  }
}
```

Berikut ini adalah contoh peristiwa yang dipancarkan saat tindakan pemulihan permanen gagal.

```
{
   "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "EBS Snapshot Notification",
  "source": "aws.ec2",
  "account": "123456789012",
  "time": "2021-05-25T13:12:22Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2:us-east-1::snapshot/snap-01234567890abcdef"
  ],
  "detail": {
    "event": "permanentRestoreSnapshot",
    "result": "failed",
    "cause": "Source snapshot ID is not valid",
    "request-id": "1234567890",
```

```
    "snapshot_id": "arn:aws:ec2:us-east-1::snapshot/snap-01234567890abcdef",
    "startTime": "2021-05-25T13:12:22Z",
    "endTime": "2021-05-45T15:30:00Z",
    "recycleBinExitTime": "2021-10-45T15:30:00Z"
  }
}
```

- `temporaryRestoreSnapshot` — Dipancarkan ketika tindakan pemulihan sementara berhasil atau gagal.

  Berikut ini adalah contoh peristiwa yang dipancarkan saat tindakan pemulihan sementara berhasil.

```
{
   "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "EBS Snapshot Notification",
  "source": "aws.ec2",
  "account": "123456789012",
  "time": "2021-05-25T13:12:22Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2:us-east-1::snapshot/snap-01234567890abcdef"
  ],
  "detail": {
    "event": "temporaryRestoreSnapshot",
    "result": "succeeded",
    "cause": "",
    "request-id": "1234567890",
    "snapshot_id": "arn:aws:ec2:us-us-east-1::snapshot/snap-01234567890abcdef",
    "startTime": "2021-05-25T13:12:22Z",
    "endTime": "2021-05-45T15:30:00Z",
    "restoreExpiryTime": "2021-06-45T15:30:00Z",
    "recycleBinExitTime": "2021-10-45T15:30:00Z"
  }
}
```

  Berikut ini adalah contoh peristiwa yang dipancarkan saat tindakan pemulihan sementara gagal.

```
{
   "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "EBS Snapshot Notification",
  "source": "aws.ec2",
```

```
  "account": "123456789012",
  "time": "2021-05-25T13:12:22Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2:us-east-1::snapshot/snap-01234567890abcdef"
  ],
  "detail": {
    "event": "temporaryRestoreSnapshot",
    "result": "failed",
    "cause": "Source snapshot ID is not valid",
    "request-id": "1234567890",
    "snapshot_id": "arn:aws:ec2:us-east-1::snapshot/snap-01234567890abcdef",
    "startTime": "2021-05-25T13:12:22Z",
    "endTime": "2021-05-45T15:30:00Z",
    "recycleBinExitTime": "2021-10-45T15:30:00Z"
  }
}
```

- `restoreExpiry` — Dipancarkan saat periode pemulihan untuk snapshot yang dipulihkan sementara kedaluwarsa.

  Berikut sebuah contoh.

```
{
   "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "EBS Snapshot Notification",
  "source": "aws.ec2",
  "account": "123456789012",
  "time": "2021-05-25T13:12:22Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2:us-east-1::snapshot/snap-01234567890abcdef"
  ],
  "detail": {
    "event": "restoryExpiry",
    "result": "succeeded",
    "cause": "",
    "request-id": "1234567890",
    "snapshot_id": "arn:aws:ec2:us-east-1::snapshot/snap-01234567890abcdef",
    "startTime": "2021-05-25T13:12:22Z",
    "endTime": "2021-05-45T15:30:00Z",
    "recycleBinExitTime": "2021-10-45T15:30:00Z"
```

```
    }
}
```

Hapus snapshot Amazon EBS

Setelah Anda tidak lagi memerlukan snapshot Amazon EBS untuk suatu volume, Anda dapat menghapusnya. Menghapus snapshot tidak berpengaruh pada volume. Menghapus volume tidak berdampak pada snapshot yang dibuat.

Penghapusan snapshot inkremental

Jika Anda membuat snapshot berkala untuk volume, snapshotnya bersifat inkremental. Hal ini berarti bahwa hanya blok pada perangkat yang berubah setelah snapshot terbaru Anda disimpan di snapshot baru. Meskipun snapshot disimpan secara bertahap, proses penghapusan snapshot dirancang agar Anda hanya mempertahankan snapshot terbaru untuk membuat volume.

Jika data terdapat pada volume yang disimpan dalam snapshot atau serangkaian snapshot sebelumnya, dan data lalu dihapus dari volume tersebut di lain waktu, data tersebut masih dianggap sebagai data unik dari snapshot sebelumnya. Data unik hanya dihapus dari urutan snapshot jika semua snapshot yang mengacu pada data unik tersebut dihapus.

Saat Anda menghapus snapshot, hanya data yang dirujuk secara eksklusif oleh snapshot tersebut yang dihapus. Data unik hanya dihapus jika semua snapshot yang mereferensikannya dihapus. Menghapus snapshot sebelumnya dari suatu volume tidak akan memengaruhi kemampuan Anda untuk membuat volume dari snapshot yang lebih baru dari volume tersebut.

Menghapus snapshot mungkin tidak akan mengurangi biaya penyimpanan data organisasi Anda. Snapshot lain mungkin merujuk pada data snapshot, dan data yang direferensikan selalu dipertahankan. Jika Anda menghapus snapshot yang berisi data yang digunakan berikutnya, biaya yang terkait dengan data yang direferensikan dialokasikan ke snapshot berikutnya. Untuk informasi selengkapnya tentang cara snapshot menyimpan data, lihat Cara kerja snapshot dan contoh berikut.

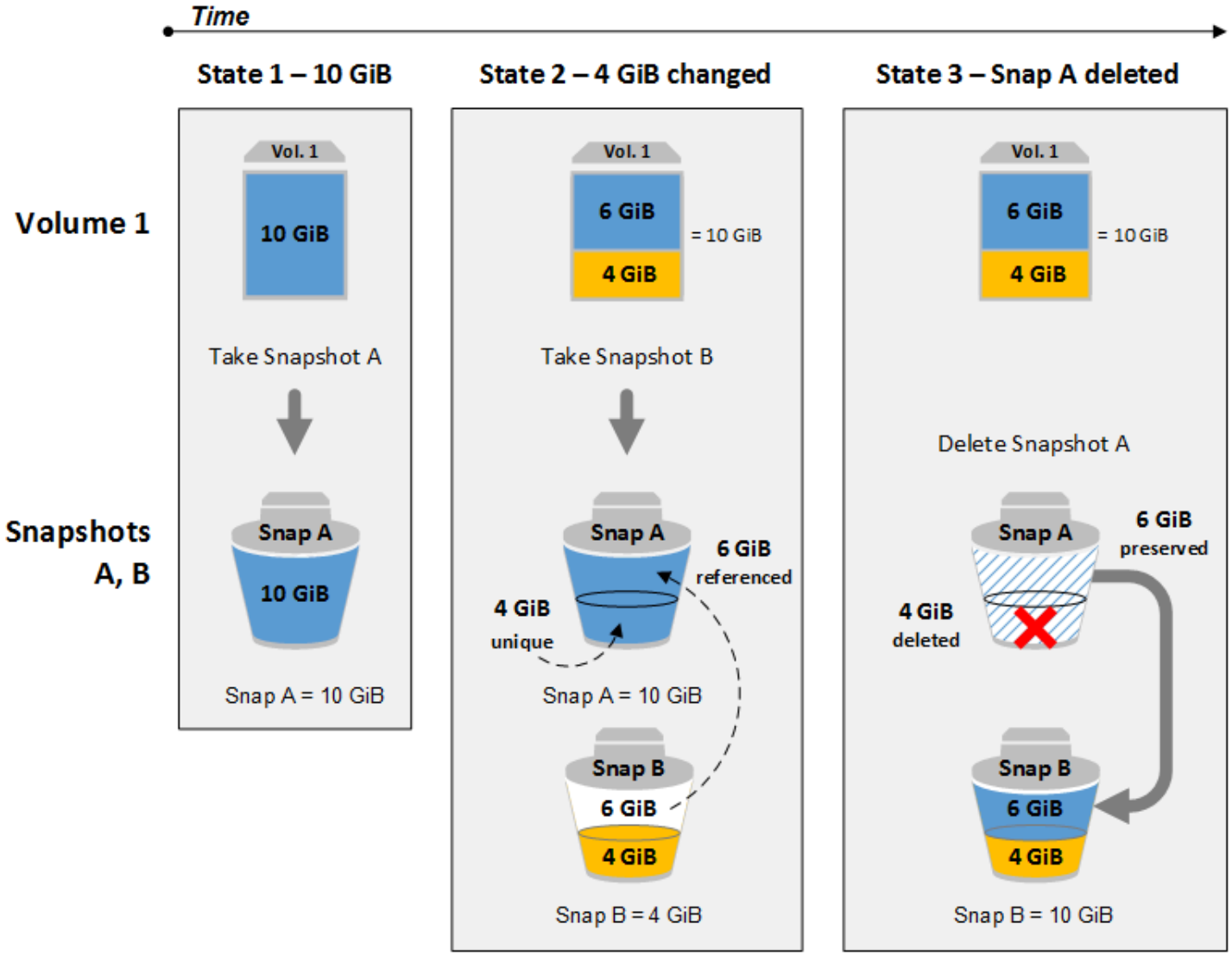
Dalam diagram berikut, Volume 1 ditampilkan pada tiga titik waktu. Snapshot telah menangkap masing-masing dari dua status pertama, dan di status ketiga, snapshot telah dihapus.

- Dalam Status 1, volume memiliki 10 data yang sama. Karena Snap A adalah snapshot pertama yang diambil dari volume, secara keseluruhan 10 GiB data harus disalin.
- Pada Status 2, volume masih berisi 10 data, tetapi 4 telah berubah. Snap B hanya perlu menyalin dan menyimpan 4 GiB yang berubah setelah Snap A diambil. 6 GiB data lainnya tidak berubah,

yang sudah disalin dan disimpan dalam Snap A, direferensikan oleh Snap B alih-alih disalin (kembali). Hal ini ditunjukkan dengan panah putus-putus.

- Pada status 3, volume tidak berubah sejak Status 2, tetapi Snapshot A telah dihapus. 6 GiB data yang disimpan dalam Snapshot A yang dirujuk oleh Snapshot B sekarang telah dipindahkan ke Snapshot B, seperti yang ditunjukkan oleh panah berat. Sebagai akibatnya, Anda masih dikenai biaya untuk menyimpan 10 GiB data; 6 GiB data yang tidak berubah yang disimpan dari Snap A, dan 4 GiB data yang diubah dari Snap B.

Menghapus snapshot dengan beberapa data yang direferensikan oleh snapshot lain



Pertimbangan

Pertimbangan berikut berlaku untuk menghapus snapshot:

- Anda tidak dapat menghapus snapshot dari perangkat root volume EBS yang digunakan oleh AMI terdaftar. Pertimbangan ini berlaku bahkan jika AMI yang terdaftar diusangkan atau dinonaktifkan. Anda harus membatalkan pendaftaran AMI terlebih dahulu sebelum Anda dapat menghapus snapshot. Untuk informasi selengkapnya, lihat membatalkan pendaftaran AMI Anda.
- Anda tidak dapat menghapus snapshot yang dikelola oleh AWS Backup layanan menggunakan Amazon EC2. Sebagai gantinya, gunakan AWS Backup untuk menghapus titik pemulihan yang sesuai di brankas cadangan. Untuk informasi selengkapnya, lihat Menghapus cadangan dalam Panduan Developer AWS Backup .
- Anda dapat membuat,mempertahankan, dan menghapus snapshot secara manual, atau Anda dapat menggunakan Amazon Data Lifecycle Manager untuk mengelola snapshot Anda. Untuk informasi selengkapnya, lihat Amazon Data Lifecycle Manager.
- Meskipun Anda dapat menghapus snapshot yang masih dalam proses, snapshot harus selesai sebelum penghapusan diterapkan. Ini mungkin memerlukan waktu lama. Jika Anda juga berada di batas snapshot yang sama, dan Anda mencoba mengambil snapshot tambahan, Anda akan menemui kesalahan `ConcurrentSnapshotLimitExceeded`. Untuk informasi selengkapnya, lihat Kuota Layanan untuk Amazon EBS di Referensi Umum Amazon Web Services.
- Jika Anda menghapus snapshot yang cocok dengan aturan retensi Recycle Bin, snapshot akan disimpan di Recycle Bin alih-alih segera dihapus. Untuk informasi selengkapnya, lihat Recycle Bin.
- Anda tidak dapat menghapus snapshot yang terkait dengan AMI yang didukung EBS yang dinonaktifkan. Untuk informasi selengkapnya, lihat Menonaktifkan AMI.
- Anda tidak dapat menghapus snapshot yang dibagikan dengan Anda.
- Jika Anda menghapus snapshot bersama yang Anda miliki, semua akun yang dengannya snapshot dibagikan kehilangan akses ke sana.

Menghapus snapshot

Untuk menghapus snapshot, gunakan salah satu metode berikut.

Console

Untuk menghapus snapshot menggunakan konsol

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Snapshot.
3. Pilih snapshot yang akan dihapus, lalu pilih Tindakan, Hapus snapshot.

4. Pilih Hapus.

AWS CLI

Untuk menghapus snapshot menggunakan AWS CLI

Gunakan perintah delete-snapshot.

Tools for Windows PowerShell

Untuk menghapus snapshot menggunakan Alat untuk Windows PowerShell

Gunakan perintah Remove-EC2Snapshot.

> **Tip pemecahan masalah**
>
> Jika Anda mendapatkan `Failed to delete snapshot` kesalahan yang menunjukkan bahwa snapshot saat ini sedang digunakan oleh AMI, Anda harus membatalkan pendaftaran AMI terkait sebelum Anda dapat menghapus snapshot. Anda tidak dapat menghapus snapshot yang terkait dengan AMI.
> Jika Anda menggunakan konsol dan AMI terkait dinonaktifkan, Anda harus memilih filter Gambar dinonaktifkan di layar AMI untuk melihat AMI yang dinonaktifkan.

Menghapus snapshot multivolume

Untuk menghapus snapshot multivolume, ambil semua snapshot untuk snapshot multivolume Anda menggunakan tanda yang Anda terapkan ke set saat Anda membuat snapshot. Lalu, hapus snapshot secara terpisah.

Anda tidak akan dicegah menghapus snapshot individual dalam set snapshot multi-volume. Jika Anda menghapus snapshot saat berada di `pending state`, hanya snapshot tersebut yang dihapus. Snapshot lain dalam set snapshot multivolume masih berhasil diselesaikan.

Otomatiskan siklus hidup snapshot

Anda dapat menggunakan Amazon Data Lifecycle Manager untuk mengotomatiskan pembuatan, retensi, dan penghapusan snapshot yang Anda gunakan untuk mencadangkan volume Amazon EBS Anda.

Untuk informasi selengkapnya, lihat Amazon Data Lifecycle Manager.

Pemulihan snapshot cepat Amazon EBS

Pemulihan snapshot cepat (FSR) Amazon EBS memungkinkan Anda membuat volume dari snapshot yang sepenuhnya diinisialisasi saat pembuatan. Hal ini menghilangkan latensi operasi I/O pada blok ketika diakses untuk pertama kalinya. Volume yang dibuat menggunakan pemulihan snapshot cepat secara instans memberikan semua performa yang disediakan.

Untuk memulai, aktifkan pemulihan snapshot cepat untuk snapshot tertentu di Zona Ketersediaan tertentu. Setiap pasangan snapshot dan Zona Ketersediaan merujuk pada satu pemulihan snapshot cepat. Saat Anda membuat volume dari salah satu snapshot ini di salah satu Zona Ketersediaan yang diaktifkan, volume tersebut dipulihkan menggunakan pemulihan snapshot cepat.

Pemulihan snapshot cepat harus diaktifkan secara eksplisit berdasarkan per snapshot. Jika Anda membuat snapshot baru dari volume yang dipulihkan dari snapshot yang mengaktifkan pemulihan snapshot cepat, snapshot baru tidak secara otomatis diaktifkan untuk pemulihan snapshot cepat. Anda harus mengaktifkannya secara eksplisit untuk snapshot baru.

Jumlah volume yang dapat Anda pulihkan dengan manfaat performa penuh dari pemulihan snapshot cepat ditentukan oleh kredit pembuatan volume untuk snapshot tersebut. Untuk informasi selengkapnya, lihat Kredit pembuatan volume.

Anda dapat mengaktifkan pemulihan snapshot cepat untuk snapshot yang Anda miliki dan untuk snapshot publik serta privat yang dibagikan dengan Anda.

Daftar Isi

Pertimbangan

- Pemulihan snapshot cepat tidak didukung dengan AWS Outposts, Local Zones, dan Wavelength Zones.
- Pemulihan snapshot cepat dapat diaktifkan pada snapshot dengan ukuran 16 TiB atau kurang.
- Volume yang disediakan dengan performa hingga 64.000 IOPS dan throughput 1.000 MiB/dtk menerima manfaat performa penuh dari pemulihan snapshot cepat. Untuk volume yang disediakan dengan performa lebih dari 64.000 IOPS atau throughput 1.000 MiB/dtk, sebaiknya inisialisasi volume untuk menerima performa penuhnya.

Kredit pembuatan volume

Jumlah volume yang menerima manfaat kinerja penuh dari pemulihan snapshot cepat ditentukan oleh kredit pembuatan volume untuk snapshot tersebut. Ada satu bucket kredit per snapshot per Zona Ketersediaan. Setiap volume yang Anda buat dari snapshot dengan pemulihan snapshot cepat yang diaktifkan akan menggunakan satu kredit dari bucket kredit. Anda harus memiliki setidaknya satu kredit dalam ember untuk membuat volume yang diinisialisasi dari snapshot. Jika Anda membuat volume tetapi ada kurang dari satu kredit dalam bucket, volume dibuat tanpa manfaat pemulihan snapshot cepat.

Saat Anda mengaktifkan pemulihan snapshot cepat untuk snapshot yang dibagikan kepada Anda, Anda mendapatkan bucket kredit terpisah untuk snapshot yang dibagikan dalam akun Anda. Jika Anda membuat volume dari snapshot yang dibagikan, kredit digunakan dari bucket Anda; kredit tidak digunakan dari bucket kredit pemilik snapshot.

Ukuran bucket kredit dan tingkat pengisian ulangnya tergantung pada ukuran snapshot, bukan ukuran volume yang dibuat dari snapshot.

Saat Anda mengaktifkan pemulihan snapshot cepat untuk snapshot, bucket kredit dimulai dengan nol kredit, dan akan diisi pada tingkat yang ditetapkan hingga mencapai kapasitas kredit maksimumnya. Selain itu, saat Anda menggunakan kredit, bucket kredit diisi ulang dari waktu ke waktu hingga mencapai kapasitas kredit maksimumnya.

Laju pengisian untuk bucket dihitung sebagai berikut:

```
MIN (10, (1024 ÷ snapshot_size_gib))
```

Dan ukuran bucket kredit dihitung sebagai berikut:

```
MAX (1, MIN (10, (1024 ÷ snapshot_size_gib)))
```

Misalnya, jika Anda mengaktifkan pemulihan snapshot cepat untuk snapshot dengan ukuran 128 GiB, tingkat pengisian adalah 0.1333 kredit per menit.

```
MIN (10, (1024 ÷ 128))
 = MIN (10, 8)
 = 8 credits per hour
 = 0.1333 credits per minute
```

Dan ukuran maksimum bucket kredit adalah 8 kredit.

```
MAX (1, MIN (10, (1024 ÷ 128)))
 = MAX (1, MIN (10, 8))
 = MAX (1, 8)
 = 8 credits
```

Dalam contoh ini, saat Anda mengaktifkan pemulihan snapshot cepat, bucket kredit dimulai dengan nol kredit. Setelah 8 menit, bucket kredit memiliki kredit yang cukup untuk membuat satu volume (`0.1333 credits × 8 minutes = 1.066 credits`) yang diinisialisasi. Saat bucket kredit penuh, Anda dapat membuat 8 volume yang diinisialisasi secara bersamaan (8 kredit). Ketika bucket berada di bawah kapasitas maksimumnya, bucket diisi ulang dengan 0.1333 kredit per menit.

Anda dapat menggunakan CloudWatch metrik untuk memantau ukuran bucket kredit Anda dan jumlah kredit yang tersedia di setiap bucket. Untuk informasi selengkapnya, lihat Metrik untuk pemulihan snapshot cepat.

Setelah Anda membuat volume dari pemulihan snapshot dengan pemulihan snapshot cepat, Anda dapat menjelaskan volume menggunakan describe-volumes dan memeriksa bidang `fastRestored` di output untuk menentukan apakah volume dibuat sebagai volume menggunakan pemulihan snapshot cepat.

Kelola pemulihan snapshot cepat

Topik

- Aktifkan atau nonaktifkan pemulihan snapshot cepat.
- Lihat status pemulihan snapshot cepat untuk snapshot
- Lihat volume yang dipulihkan menggunakan pemulihan snapshot cepat

Aktifkan atau nonaktifkan pemulihan snapshot cepat.

Pemulihan snapshot cepat dinonaktifkan untuk snapshot secara default. Anda dapat mengaktifkan atau menonaktifkan pemulihan snapshot cepat untuk snapshot yang Anda miliki dan untuk snapshot yang dibagikan dengan Anda. Saat Anda mengaktifkan atau menonaktifkan pemulihan snapshot cepat untuk snapshot, perubahan hanya berlaku pada akun Anda.

> **Note**
>
> Saat Anda mengaktifkan pemulihan snapshot cepat untuk suatu snapshot, akun Anda akan dikenai biaya untuk setiap menit pengaktifan pemulihan snapshot cepat di Zona Ketersediaan tertentu. Biaya bersifat pro-rata dan memiliki minimal satu jam.

Saat Anda menghapus snapshot yang Anda miliki, pemulihan snapshot cepat secara otomatis dinonaktifkan untuk snapshot tersebut di akun Anda. Jika Anda mengaktifkan pemulihan snapshot cepat untuk snapshot yang dibagikan kepada Anda, dan pemilik snapshot menghapus atau membatalkan pembagiannya, pemulihan snapshot cepat secara otomatis dinonaktifkan untuk snapshot yang dibagikan di akun Anda.

Jika Anda mengaktifkan pemulihan snapshot cepat untuk snapshot yang dibagikan kepada Anda, dan snapshot tersebut telah dienkripsi menggunakan CMK kustom, pemulihan snapshot cepat tidak secara otomatis dinonaktifkan untuk snapshot saat pemilik snapshot mencabut akses ke CMK kustom. Anda harus menonaktifkan pemulihan snapshot cepat untuk snapshot ini secara manual.

Gunakan salah satu metode berikut untuk mengaktifkan atau menonaktifkan pemulihan snapshot cepat untuk snapshot yang Anda miliki atau untuk snapshot yang dibagikan kepada Anda.

Console

Untuk mengaktifkan atau menonaktifkan pemulihan snapshot cepat

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Snapshot.
3. Pilih snapshot, dan pilih Tindakan, Kelola pemulihan snapshot cepat.
4. Bagian Pengaturan pemulihan snapshot cepat mencantumkan semua Zona Ketersediaan di mana Anda dapat mengaktifkan pemulihan snapshot cepat untuk snapshot yang dipilih. Volume status saat ini menunjukkan apakah pemulihan snapshot cepat saat ini diaktifkan atau dinonaktifkan untuk setiap zona.

Untuk mengaktifkan pemulihan snapshot cepat di zona yang saat ini dinonaktifkan, pilih zona, pilih Aktifkan, lalu konfirmasi, pilih Aktifkan.

Untuk menonaktifkan pemulihan snapshot cepat di zona yang saat ini diaktifkan, pilih zona, lalu pilih Nonaktifkan.

5. Setelah Anda membuat perubahan yang diperlukan, pilih Tutup.

AWS CLI

Untuk mengelola pemulihan snapshot cepat menggunakan AWS CLI

- enable-fast-snapshot-restores
- disable-fast-snapshot-restores
- describe-fast-snapshot-restores

> **Note**
>
> Setelah Anda mengaktifkan pemulihan snapshot cepat untuk snapshot, snapshot memasuki status `optimizing`. Snapshot yang ada dalam status `optimizing` memberikan beberapa manfaat performa saat menggunakannya untuk memulihkan volume. Snapshot mulai memberikan manfaat performa penuh dari pemulihan snapshot cepat hanya setelah memasuki status `enabled`.

Lihat status pemulihan snapshot cepat untuk snapshot

Pemulihan snapshot cepat untuk snapshot dapat berada di salah satu status berikut.

- `enabling` – Permintaan dibuat untuk mengaktifkan pemulihan snapshot cepat.
- `optimizing` — Pemulihan snapshot cepat sedang diaktifkan. Akan memakan waktu 60 menit per TiB untuk mengoptimalkan snapshot. Snapshot dalam keadaan ini menawarkan beberapa manfaat performa saat memulihkan volume.
- `enabled` — Pemulihan snapshot cepat sedang diaktifkan. Snapshot yang berada dalam keadaan ini dan memiliki kredit pembuatan volume yang memadai menawarkan manfaat performa penuh saat memulihkan volume.

- `disabling` — Permintaan dibuat untuk menonaktifkan pemulihan snapshot cepat, atau permintaan untuk mengaktifkan pemulihan snapshot cepat yang gagal.
- `disabled` — Pemulihan snapshot cepat sedang dinonaktifkan. Anda dapat mengaktifkan pemulihan snapshot cepat sesuai kebutuhan.

Gunakan salah satu metode berikut untuk melihat status pemulihan snapshot cepat untuk snapshot yang Anda miliki atau untuk snapshot yang dibagikan kepada Anda.

Console

Untuk melihat status pemulihan snapshot cepat menggunakan konsol

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Snapshot.
3. Pilih snapshot.
4. Di tab Detail, Pemulihan snapshot cepat, menunjukkan status pemulihan snapshot cepat.

AWS CLI

Untuk melihat snapshot dengan pemulihan snapshot cepat diaktifkan menggunakan AWS CLI

Gunakan describe-fast-snapshot-restoresperintah untuk menggambarkan snapshot yang diaktifkan untuk pemulihan snapshot cepat.

```
aws ec2 describe-fast-snapshot-restores --filters Name=state,Values=enabled
```

Berikut ini adalah output contoh.

```
{
    "FastSnapshotRestores": [
        {
            "SnapshotId": "snap-0e946653493cb0447",
            "AvailabilityZone": "us-east-2a",
            "State": "enabled",
            "StateTransitionReason": "Client.UserInitiated - Lifecycle state
 transition",
            "OwnerId": "123456789012",
            "EnablingTime": "2020-01-25T23:57:49.596Z",
```

```
            "OptimizingTime": "2020-01-25T23:58:25.573Z",
            "EnabledTime": "2020-01-25T23:59:29.852Z"
        },
        {
            "SnapshotId": "snap-0e946653493cb0447",
            "AvailabilityZone": "us-east-2b",
            "State": "enabled",
            "StateTransitionReason": "Client.UserInitiated - Lifecycle state
 transition",
            "OwnerId": "123456789012",
            "EnablingTime": "2020-01-25T23:57:49.596Z",
            "OptimizingTime": "2020-01-25T23:58:25.573Z",
            "EnabledTime": "2020-01-25T23:59:29.852Z"
        }
    ]
}
```

Lihat volume yang dipulihkan menggunakan pemulihan snapshot cepat

Saat Anda membuat volume dari snapshot yang diaktifkan untuk pemulihan snapshot cepat di Zona Ketersediaan untuk volume, snapshot akan dipulihkan menggunakan pemulihan snapshot cepat.

Gunakan perintah describe-volumes untuk melihat volume yang dibuat dari snapshot yang diaktifkan untuk pemulihan snapshot cepat.

```
aws ec2 describe-volumes --filters Name=fast-restored,Values=true
```

Berikut ini adalah output contoh.

```
{
    "Volumes": [
        {
            "Attachments": [],
            "AvailabilityZone": "us-east-2a",
            "CreateTime": "2020-01-26T00:34:11.093Z",
            "Encrypted": true,
            "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/8c5b2c63-b9bc-45a3-
a87a-5513e232e843",
            "Size": 20,
            "SnapshotId": "snap-0e946653493cb0447",
            "State": "available",
```

```
            "VolumeId": "vol-0d371921d4ca797b0",
            "Iops": 100,
            "VolumeType": "gp2",
            "FastRestored": true
        }
    ]
}
```

Pantau pemulihan snapshot cepat

Amazon EBS memancarkan CloudWatch peristiwa Amazon saat status pemulihan snapshot cepat untuk snapshot berubah. Untuk informasi selengkapnya, lihat Peristiwa pemulihan snapshot cepat EBS.

Kuota pemulihan snapshot cepat

Anda dapat mengaktifkan hingga 5 snapshot untuk pemulihan snapshot cepat per Wilayah. Kuota berlaku untuk snapshot yang Anda miliki dan snapshot yang dibagikan dengan Anda. Jika Anda mengaktifkan pemulihan snapshot cepat untuk snapshot yang dibagikan kepada Anda, itu dihitung terhadap kuota pemulihan snapshot cepat Anda. Ini tidak termasuk dalam kuota pemulihan snapshot cepat pemilik snapshot.

Harga dan Penagihan

Anda dikenai biaya untuk setiap menit pengaktifan pemulihan snapshot cepat untuk snapshot dalam Zona Ketersediaan tertentu. Biaya bersifat pro-rata minimal satu jam.

Misalnya, jika Anda mengaktifkan pemulihan snapshot cepat untuk satu snapshot di `US-East-1a` selama satu bulan (30 hari), Anda dikenai biaya 540 USD (1 snapshot x 1 AZ x `720` jam x `$0.75` per jam). Jika Anda mengaktifkan pemulihan snapshot cepat untuk dua snapshot dalam `us-east-1a`, `us-east-1b`, dan `us-east-1c` untuk periode yang sama, Anda dikenai biaya 3240 USD (2 snapshot x 3 AZ x `720` jam x `$0.75` per jam).

Jika Anda mengaktifkan pemulihan snapshot cepat untuk snapshot publik atau privat yang dibagikan dengan Anda, akun Anda dikenai biaya; pemilik snapshot tidak dikenai biaya. Ketika snapshot yang dibagikan dengan Anda dihapus atau tidak dibagikan oleh pemilik snapshot, pemulihan snapshot cepat dinonaktifkan untuk snapshot di akun Anda dan penagihan dihentikan.

Untuk informasi selengkapnya, lihat harga Amazon EBS.

Kunci snapshot Amazon EBS

Anda dapat mengunci snapshot Amazon EBS untuk melindunginya dari penghapusan yang tidak disengaja atau berbahaya, atau menyimpannya dalam format WORM (write-once-read-many) untuk durasi tertentu. Sementara snapshot dikunci, snapshot tersebut tidak dapat dihapus oleh pengguna mana pun, terlepas dari izin IAM mereka. Anda dapat terus menggunakan snapshot terkunci dengan cara yang sama ketika Anda akan menggunakan snapshot lainnya.

> **Note**
>
> Kunci snapshot telah dinilai oleh Cohasset Associates untuk digunakan di lingkungan yang tunduk pada peraturan SEC 17a-4, CFTC, dan FINRA. Untuk informasi selengkapnya tentang cara kunci snapshot terkait dengan peraturan ini, lihat Penilaian Kepatuhan Cohasset Associates.

Anda dapat mengunci snapshot dalam salah satu dari dua mode: mode kepatuhan atau mode tata kelola, dan snapshot dapat dikunci selama durasi tertentu atau hingga tanggal tertentu. Untuk informasi selengkapnya, lihat Mode Kunci dan Durasi kunci.

Harga

Anda dapat mengunci dan membuka snapshot tanpa biaya tambahan. Anda membayar biaya penyimpanan snapshot Amazon EBS standar untuk snapshot terkunci.

Topik

Konsep kunci snapshot Amazon EBS

Berikut ini adalah konsep penting untuk dipahami saat Anda mulai menggunakan kunci snapshot.

Daftar Isi

Mode Kunci

Anda dapat mengunci snapshot dalam salah satu dari dua mode:

Mode tata kelola

Setelah snapshot dikunci, pengguna dengan izin IAM yang sesuai dapat membuka kunci snapshot dan memodifikasi mode kunci dan durasi kunci atau tanggal kedaluwarsa kapan saja. Saat Anda mengunci snapshot dalam mode tata kelola, snapshot segera dikunci; tidak ada periode pendinginan. Untuk menghapus snapshot setelah dikunci dalam mode tata kelola, Anda harus membuka kunci snapshot terlebih dahulu atau Anda harus menunggu kunci kedaluwarsa.

Anda dapat menggunakan mode tata kelola untuk memenuhi persyaratan tata kelola data organisasi Anda dengan memastikan bahwa hanya pengguna tertentu yang memiliki izin untuk membuka kunci snapshot dan memodifikasi konfigurasi kunci snapshot. Anda juga dapat menggunakan mode tata kelola untuk menguji konfigurasi kunci Anda sebelum mengunci snapshot dalam mode kepatuhan.

Mode kepatuhan

Saat Anda mengunci snapshot dalam mode kepatuhan, Anda dapat secara opsional menentukan periode pendinginan yang dimulai segera setelah Anda mengunci snapshot. Selama periode pendinginan, pengguna dengan izin yang sesuai dapat membuka kunci snapshot, mengubah mode kunci, menambah atau mengurangi periode pendinginan, dan menambah atau mengurangi durasi kunci atau tanggal kedaluwarsa. Setelah periode pendinginan berakhir, Anda tidak dapat membuka kunci snapshot, mengubah mode kunci, atau mengurangi durasi kunci atau tanggal kedaluwarsa; Anda hanya dapat menambah durasi kunci atau tanggal kedaluwarsa. Untuk menghapus snapshot setelah dikunci sesuai dan periode pendinginan telah kedaluwarsa, Anda harus menunggu hingga kunci kedaluwarsa.

> **Note**
>
> Anda dapat mengunci snapshot dalam mode kepatuhan tanpa periode pendinginan dengan menghilangkan periode pendinginan dalam permintaan. Setelah periode pendinginan berakhir, Anda tidak dapat membuka kunci snapshot, mengubah mode kunci, atau mengurangi durasi kunci atau tanggal kedaluwarsa; Anda hanya dapat menambah durasi kunci atau tanggal kedaluwarsa.

Anda dapat menggunakan mode kepatuhan untuk melindungi snapshot yang tidak boleh dihapus untuk periode tertentu karena alasan kepatuhan. Mode kepatuhan menawarkan manfaat sebagai berikut:

- Hal ini memungkinkan konfigurasi WORM (tulis sekali, baca banyak) untuk snapshot Anda.
- Ini memberikan lapisan pertahanan tambahan yang melindungi snapshot dari penghapusan yang tidak disengaja atau berbahaya.
- Ini memberlakukan periode retensi, yang mencegah penghapusan dini oleh pengguna istimewa, untuk memenuhi kebijakan dan prosedur perlindungan data organisasi Anda.

> **Note**
>
> Satu-satunya cara untuk menghapus snapshot yang dikunci dalam mode kepatuhan sebelum kuncinya kedaluwarsa adalah dengan menutup akun terkait AWS .

Durasi kunci

Durasi kunci adalah periode waktu di mana snapshot tetap terkunci. Anda dapat menentukan durasi kunci sebagai salah satu dari berikut ini, tetapi tidak keduanya:

Jumlah hari

Durasi kunci ditentukan sebagai beberapa hari di mana snapshot tetap terkunci. Setelah jumlah hari yang ditentukan berlalu, snapshot secara otomatis dibuka kuncinya. Durasi dapat berkisar dari 1 hari hingga 36500 hari (100 tahun).

Tanggal kedaluwarsa kunci

Durasi kunci ditentukan oleh tanggal kedaluwarsa di masa mendatang. Snapshot tetap terkunci sampai tanggal kedaluwarsa kunci tercapai. Ketika tanggal kedaluwarsa kunci tercapai, snapshot secara otomatis dibuka kuncinya.

Periode pendinginan

Periode pendinginan adalah periode waktu opsional yang dapat Anda tentukan saat Anda mengunci snapshot dalam mode kepatuhan. Selama periode pendinginan, pengguna dengan izin yang sesuai dapat membuka kunci snapshot, mengubah mode kunci, menambah atau mengurangi periode pendinginan, dan menambah atau mengurangi durasi kunci atau tanggal kedaluwarsa. Setelah periode pendinginan berakhir, pengguna tidak dapat membuka kunci snapshot, mengubah mode kunci, mengembalikan periode pendinginan, atau mengurangi durasi penguncian, terlepas dari izin mereka.

Snapshot tidak dapat dihapus selama periode pendinginan.

Jika ditentukan, periode pendinginan dimulai segera setelah Anda mengunci snapshot. Jika dihilangkan, snapshot dikunci dalam mode kepatuhan segera tanpa periode pendinginan.

Periode pendinginan dapat berkisar dari 1 hingga 72 jam. Anda dapat mengunci snapshot dalam mode kepatuhan tanpa periode pendinginan dengan menghilangkan periode pendinginan dalam permintaan.

Status kunci

Kunci snapshot dapat berada di salah satu status berikut:

- `compliance-cooloff` — Snapshot telah dikunci dalam mode kepatuhan, tetapi masih dalam periode pendinginan. Snapshot tidak dapat dihapus, tetapi dapat dibuka kuncinya dan pengaturan kunci dapat dimodifikasi oleh pengguna dengan izin yang sesuai.
- `governance` — Snapshot dikunci dalam mode tata kelola. Snapshot tidak dapat dihapus, tetapi dapat dibuka kuncinya dan pengaturan kunci dapat dimodifikasi oleh pengguna dengan izin yang sesuai.
- `compliance` — Snapshot dikunci dalam mode kepatuhan tanpa periode pendinginan atau periode pendinginan telah kedaluwarsa. Snapshot tidak dapat dibuka atau dihapus. Durasi kunci hanya dapat ditingkatkan oleh pengguna dengan izin yang sesuai.
- `expired` — Snapshot dikunci dalam mode kepatuhan atau tata kelola, tetapi kunci telah kedaluwarsa. Snapshot tidak terkunci dan dapat dihapus.

Pertimbangan untuk kunci snapshot Amazon EBS

- Anda dapat mengunci snapshot hanya jika snapshot ada dalam status `pending` atau `completed`.
  - Jika Anda mengunci snapshot saat berada dalam status `pending`, dan Anda menguncinya untuk durasi tertentu, durasi kunci hanya dimulai ketika snapshot mencapai status `completed`. Snapshot tidak dapat dihapus saat berada dalam status `pending`.
  - Jika Anda mengunci snapshot saat berada dalam status `pending` dan pembuatan snapshot gagal karena alasan apa pun, kunci dibatalkan.
- Jika Anda memperpanjang durasi penguncian untuk snapshot yang terkunci dalam mode kepatuhan setelah periode pendinginan berakhir, Anda tidak dapat menentukan periode pendinginan lainnya. Jika Anda menentukan periode pendinginan, permintaan gagal.
- Anda dapat mengunci snapshot yang diarsipkan. Dan Anda dapat mengarsipkan snapshot yang terkunci.
- Anda dapat mengunci snapshot yang terkait dengan AMI.
- Anda dapat membatalkan pendaftaran AMI yang terkait dengan snapshot yang dikunci.
- Anda dapat menghapus kunci KMS yang digunakan untuk mengenkripsi snapshot yang terkunci.
- Kami menyarankan Anda untuk tidak mengunci snapshot yang dibuat oleh AWS Backup. AWS Backup sudah memastikan bahwa snapshot-nya tidak dihapus sebelum periode retensi mereka berakhir. Untuk menambahkan lapisan keamanan tambahan untuk snapshot yang dikelola oleh AWS Backup, kami sarankan Anda menggunakan AWS Backup Vault Lock. Untuk informasi selengkapnya, lihat Kunci Penyimpanan AWS Backup.
- Anda tidak dapat mengunci snapshot selama pembuatan atau selama pendaftaran AMI.
- Anda tidak dapat mengunci snapshot Amazon EBS lokal di AWS Outposts.
- Satu-satunya cara untuk menghapus snapshot yang dikunci dalam mode kepatuhan sebelum kuncinya kedaluwarsa adalah dengan menutup akun terkait AWS .

  Jika Anda menutup AWS akun Anda saat Anda telah mengunci snapshot, AWS menangguhkan akun Anda selama 90 hari dengan snapshot Anda utuh. Jika Anda tidak membuka kembali akun Anda dalam 90 hari, AWS hapus snapshot Anda, meskipun terkunci.

Izin yang diperlukan untuk kunci snapshot Amazon EBS

Secara default, pengguna tidak memiliki izin untuk bekerja dengan kunci snapshot. Untuk mengizinkan pengguna bekerja dengan sumber daya ini, Anda harus membuat kebijakan IAM

yang memberikan izin menggunakan sumber daya dan tindakan API tertentu. Untuk informasi selengkapnya, lihat Membuat kebijakan IAM dalam Panduan Pengguna IAM.

Topik

Izin yang diperlukan

Untuk bekerja dengan kunci snapshot, pengguna memerlukan izin berikut.

- `ec2:LockSnapshot` — Untuk mengunci snapshot.
- `ec2:UnlockSnapshot` — Untuk membuka kunci snapshot.
- `ec2:DescribeLockedSnapshots` — Untuk melihat pengaturan kunci snapshot.

Berikut ini adalah contoh kebijakan IAM yang memberi pengguna izin untuk mengunci dan membuka kunci snapshot, dan untuk melihat pengaturan kunci snapshot. Ini termasuk izin `ec2:DescribeSnapshots` untuk pengguna konsol. Jika beberapa izin tidak diperlukan, Anda dapat menghapusnya dari kebijakan.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "ec2:LockSnapshot",
            "ec2:UnlockSnapshot",
            "ec2:DescribeLockedSnapshots",
            "ec2:DescribeSnapshots"
        ]
    }]
}
```

Untuk memberikan akses, menambahkan izin ke pengguna, grup, atau peran Anda:

- Pengguna dan grup di AWS IAM Identity Center:

  Buat rangkaian izin. Ikuti instruksi di Buat rangkaian izin di Panduan Pengguna AWS IAM Identity Center .

- Pengguna yang dikelola di IAM melalui penyedia identitas:

  Buat peran untuk federasi identitas. Ikuti instruksi dalam Membuat peran untuk penyedia identitas pihak ketiga (federasi) di Panduan Pengguna IAM.
- Pengguna IAM:
  - Buat peran yang dapat diambil pengguna Anda. Ikuti instruksi dalam Membuat peran untuk pengguna IAM dalam Panduan Pengguna IAM.
  - (Tidak disarankan) Pasang kebijakan langsung ke pengguna atau tambahkan pengguna ke grup pengguna. Ikuti petunjuk dalam Menambahkan izin ke pengguna (konsol) dalam Panduan Pengguna IAM.

Batasi akses dengan kunci syarat

Anda dapat menggunakan kunci syarat untuk membatasi cara pengguna diizinkan untuk mengunci snapshot.

Topik

- EC2: SnapshotLockDuration
- EC2: CoolOffPeriod

EC2: SnapshotLockDuration

Anda dapat menggunakan kunci syarat `ec2:SnapshotLockDuration` untuk membatasi pengguna pada durasi kunci tertentu saat mengunci snapshot.

Contoh kebijakan berikut membatasi pengguna untuk menentukan durasi kunci antara `10` dan `50` hari.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:LockSnapshot",
      "Resource": "arn:aws:ec2:region::snapshot/*"
      "Condition": {
        "NumericGreaterThan" : {
          "ebs:SnapshotLockDuration" : 10
        }
```

```
        "NumericLessThan":{
          "ebs:SnapshotLockDuration": 50
        }
      }
    }
  ]
}
```

EC2: CoolOffPeriod

Anda dapat menggunakan kunci syarat `ec2:CoolOffPeriod` untuk mencegah pengguna mengunci snapshot dalam mode kepatuhan tanpa periode pendinginan.

Contoh kebijakan berikut membatasi pengguna untuk menentukan periode pendinginan lebih dari 48 jam saat mengunci snapshot dalam mode kepatuhan.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:LockSnapshot",
      "Resource": "arn:aws:ec2:region::snapshot/*"
      "Condition": {
        "NumericGreaterThan": {
          "ec2:CoolOffPeriod": 48
        }
      }
    }
  ]
}
```

Bekerja dengan kunci snapshot Amazon EBS

Gunakan prosedur berikut untuk bekerja dengan kunci snapshot Amazon EBS.

Tugas

- Mengunci snapshot
- Membuka kunci snapshot
- Memperbarui pengaturan kunci snapshot
- Untuk melihat pengaturan kunci snapshot

Mengunci snapshot

Anda dapat mengunci snapshot yang ada dalam status `pending` atau `completed`. Untuk informasi selengkapnya, lihat Pertimbangan untuk kunci snapshot Amazon EBS.

Console

Untuk mengunci snapshot

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Snapshot.
3. Pilih snapshot yang akan dikunci dan pilih Tindakan, Pengaturan snapshot, Kelola kunci snapshot.
4. Pilih Kunci snapshot.
5. Untuk Mode kunci, pilih Mode tata kelola atau Mode kepatuhan. Untuk informasi selengkapnya, lihat Mode Kunci.
6. Untuk Durasi Penguncian, lakukan salah satu hal berikut:

   - Untuk mengunci snapshot untuk periode tertentu, pilih Kunci snapshot untuk, lalu masukkan periode dalam beberapa hari atau tahun.
   - Untuk mengunci snapshot hingga tanggal dan waktu tertentu, pilih Kunci snapshot hingga, lalu pilih tanggal dan waktu kedaluwarsa.

   Untuk informasi selengkapnya, lihat Durasi kunci.
7. (Hanya mode kepatuhan) Untuk Periode pendinginan, tentukan periode pendinginan di mana Anda dapat membuka kunci snapshot dan memodifikasi konfigurasi kunci. Untuk informasi selengkapnya, lihat Periode pendinginan.
8. (Hanya mode kepatuhan) Untuk mengonfirmasi bahwa Anda ingin mengunci snapshot dalam mode kepatuhan dan bahwa Anda tidak akan dapat membuka kunci snapshot setelah periode pendinginan berakhir, pilih Akui.
9. Pilih Simpan pengaturan kunci.

AWS CLI

Untuk mengunci snapshot dalam mode tata kelola

Gunakan perintah AWS CLI lock-snapshot. Untuk `--snapshot-id`, tentukan ID snapshot yang akan dikunci. Untuk `--lock-mode`, tentukan `governance`. Untuk mengunci snapshot untuk periode tertentu, untuk `--lock-duration`, tentukan periode untuk mengunci snapshot. Atau, untuk mengunci snapshot hingga tanggal tertentu, untuk `--expiration-date`, tentukan tanggal dan waktu di mana kunci harus kedaluwarsa, di zona waktu UTC (`YYYY-MM-DDThh:mm:ss.sssZ`).

```
$ aws ec2 lock-snapshot --snapshot-id snapshot_id \
 --lock-mode governance \
 --lock-duration 1-36500_days | --expiration-date YYYY-MM-DDThh:mm:ss.sssZ
```

Untuk mengunci snapshot dalam mode kepatuhan

Gunakan perintah AWS CLI lock-snapshot. Untuk `--snapshot-id`, tentukan ID snapshot yang akan dikunci. Untuk `--lock-mode`, tentukan `compliance`. Untuk `--cool-off-period`, secara opsional tentukan periode pendinginan dalam beberapa jam. Untuk mengunci snapshot untuk periode tertentu, untuk `--lock-duration`, tentukan periode untuk mengunci snapshot. Atau, untuk mengunci snapshot hingga tanggal tertentu, untuk `--expiration-date`, tentukan tanggal dan waktu di mana kunci harus kedaluwarsa, di zona waktu UTC (`YYYY-MM-DDThh:mm:ss.sssZ`).

```
$ aws ec2 lock-snapshot --snapshot-id snapshot_id \
 --lock-mode compliance \
 --cool-off-period 1-72_hours \
 --lock-duration 1-36500_days | --expiration-date YYYY-MM-DDThh:mm:ss.sssZ
```

Membuka kunci snapshot

Anda dapat membuka kunci snapshot hanya jika terkunci dalam mode tata kelola, atau jika terkunci dalam mode kepatuhan dan masih dalam periode pendinginan.

Console

Untuk membuka kunci snapshot

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Snapshot.

3. Pilih snapshot yang akan dibuka kuncinya dan pilih Tindakan, Pengaturan snapshot, Kelola kunci snapshot.
4. Pilih Buka kunci snapshot lalu pilih Buka kunci snapshot lagi untuk mengonfirmasi.

AWS CLI

Untuk membuka kunci snapshot

Gunakan perintah AWS CLI unlock-snapshot. Untuk `--snapshot-id`, tentukan ID snapshot yang akan dibuka kuncinya.

```
$ aws ec2 unlock-snapshot --snapshot-id snapshot_id
```

Memperbarui pengaturan kunci snapshot

Pembaruan yang diizinkan tergantung pada status kunci:

- `governance` — Anda dapat mengubah mode kunci dan menambah atau mengurangi durasi kunci atau tanggal kedaluwarsa.
- `compliance-cooloff` — Anda dapat mengubah mode kunci, menambah atau mengurangi periode pendinginan, dan menambah atau mengurangi durasi kunci atau tanggal kedaluwarsa.
- `compliance` — Anda hanya dapat meningkatkan durasi kunci atau tanggal kedaluwarsa.

Console

Untuk memperbarui pengaturan kunci snapshot

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Snapshot.
3. Pilih snapshot yang pengaturan kuncinya akan dimodifikasi dan pilih Tindakan, Pengaturan snapshot, Kelola kunci snapshot.
4. Perbarui pengaturan sesuai kebutuhan, lalu pilih Simpan pengaturan kunci.

AWS CLI

Untuk memperbarui pengaturan kunci snapshot

Gunakan perintah AWS CLI lock-snapshot. Untuk `--snapshot-id`, tentukan ID snapshot untuk memperbarui pengaturan kunci. Kemudian, tentukan hanya opsi untuk dimodifikasi.

Untuk melihat pengaturan kunci snapshot

Gunakan salah satu metode berikut ini untuk melihat pengaturan kunci untuk snapshot.

Console

Untuk melihat pengaturan kunci snapshot

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Snapshot.
3. Pilih snapshot untuk melihat pengaturan kunci dan pilih Tindakan, Pengaturan snapshot, Kelola kunci snapshot.

AWS CLI

Untuk melihat pengaturan kunci snapshot

Gunakan describe-locked-snapshots AWS CLI perintah. Untuk `--snapshot-ids`, tentukan ID snapshot untuk melihat pengaturan kunci.

```
$ aws ec2 describe-locked-snapshots --snapshot-ids snapshot_id
```

Pantau kunci snapshot Amazon EBS menggunakan AWS CloudTrail

Anda dapat memantau panggilan API untuk kunci snapshot sebagai peristiwa, termasuk panggilan dari konsol dan dari panggilan kode ke API. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan.

Untuk informasi selengkapnya, lihat Pencatatan panggilan API AWS CloudTrail.

Pantau kunci snapshot Amazon EBS menggunakan Amazon EventBridge

Amazon EBS memancarkan peristiwa yang terkait dengan tindakan kunci snapshot. Anda dapat menggunakan AWS Lambda dan Amazon EventBridge untuk menangani pemberitahuan acara

secara terprogram. Peristiwa dipancarkan atas dasar upaya terbaik. Untuk informasi selengkapnya, lihat Panduan EventBridge Pengguna Amazon.

Peristiwa berikut dipancarkan:

- Snapshot berhasil dikunci dalam mode tata kelola atau kepatuhan.

```
{
  "version": "0",
  "id": "01234567-01234-0123-0123-012345678901",
  "detail-type": "EBS Snapshot Notification",
  "source": "aws.ec2",
  "account": "012345678901",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2::us-west-2:snapshot/snap-01234567890abcdef"
  ],
  "detail": {
    "event": "lockSnapshot",
    "result": "succeeded",
    "snapshot_id": "arn:aws:ec2::us-west-2:snapshot/snap-01234567890abcdef",
    "source": 012345678901,
    "lockState": "compliance-cooloff",
    "lockCreatedOn": "yyyy-mm-ddThh:mm:ssZ",
    "lockExpiresOn": "yyyy-mm-ddThh:mm:ssZ",
    "lockDuration": 123,
    "lockStartDurationTime": "yyyy-mm-ddThh:mm:ssZ",
    "cooOffPeriod": 24,
    "coolOffPeriodExpiresOn": "yyyy-mm-ddThh:mm:ssZ"
  }
}
```

- Peristiwa penguncian gagal jika snapshot dikunci saat berada dalam status `pending`, dan snapshot gagal mencapai status `completed`.

```
{
  "version": "0",
  "id": "01234567-01234-0123-0123-012345678901",
  "detail-type": "EBS Snapshot Notification",
  "source": "aws.ec2",
  "account": "012345678901",
  "time": "yyyy-mm-ddThh:mm:ssZ",
```

```
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2::us-west-2:snapshot/snap-01234567890abcdef"
  ],
  "detail": {
    "event": "lockSnapshot",
    "result": "failed",
    "cause": "snapshot failed",
    "snapshot_id": "arn:aws:ec2::us-west-2:snapshot/snap-01234567890abcdef",
    "lockState": "pending-compliance",
    "lockCreatedOn": "yyyy-mm-ddThh:mm:ssZ",
    "lockDuration": 123,
    "lockStartDurationTime": "yyyy-mm-ddThh:mm:ssZ",
    "cooOffPeriod": 24,
    "coolOffPeriodExpiresOn": "yyyy-mm-ddThh:mm:ssZ"
  }
}
```

- Kunci kedaluwarsa

```
{
  "version": "0",
  "id": "01234567-01234-0123-0123-012345678901",
  "detail-type": "EBS Snapshot Notification",
  "source": "aws.ec2",
  "account": "012345678901",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2::us-west-2:snapshot/snap-01234567890abcdef"
  ],
  "detail": {
    "event": "lockDurationExpiry",
    "result": "succeeded",
    "snapshot_id": "arn:aws:ec2::us-west-2:snapshot/snap-01234567890abcdef",
    "lockState": "expired",
    "lockCreatedOn": "yyyy-mm-ddThh:mm:ssZ",
    "lockExpiresOn": "yyyy-mm-ddThh:mm:ssZ",
    "lockDuration": 123
  }
}
```

- Periode pendinginan berakhir setelah dikunci dalam mode kepatuhan.

```
{
  "version": "0",
  "id": "01234567-01234-0123-0123-012345678901",
  "detail-type": "EBS Snapshot Notification",
  "source": "aws.ec2",
  "account": "012345678901",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2::us-west-2:snapshot/snap-01234567890abcdef"
  ],
  "detail": {
    "event": "cooloffperiodExpiry",
    "result": "succeeded",
    "snapshot_id": "arn:aws:ec2::us-west-2:snapshot/snap-01234567890abcdef",
    "lockState": "compliance",
    "lockCreatedOn": "yyyy-mm-ddThh:mm:ssZ",
    "lockExpiresOn": "yyyy-mm-ddThh:mm:ssZ",
    "lockDuration": 123,
    "lockStartDurationTime": "yyyy-mm-ddThh:mm:ssZ",
    "cooOffPeriod": 24,
    "coolOffPeriodExpiresOn": "yyyy-mm-ddThh:mm:ssZ"
  }
}
```

Memblokir akses publik untuk snapshot

Untuk mencegah berbagi snapshot secara publik, Anda sekarang dapat mengaktifkan blokir akses publik untuk snapshot. Setelah Anda mengaktifkan blokir akses publik untuk snapshot di Wilayah, setiap upaya untuk membagikan snapshot secara publik di Wilayah tersebut akan diblokir secara otomatis. Pengaktifan ini dapat membantu Anda meningkatkan keamanan snapshot dan untuk melindungi data snapshot Anda dari akses yang tidak terotorisasi atau tidak diinginkan.

Blokir akses publik untuk snapshot dapat diaktifkan dalam salah satu dari dua mode:

- Blokir semua pembagian — Memblokir semua pembagian snapshot Anda ke publik Pengguna di akun tidak dapat meminta berbagi publik baru. Selain itu, snapshot yang sudah dibagikan secara publik diperlakukan sebagai privat dan tidak lagi tersedia secara publik.

- Blokir pembagian baru — Hanya memblokir pembagian snapshot baru ke publik. Pengguna di akun tidak dapat meminta berbagi publik baru. Namun, snapshot yang sudah dibagikan secara publik, tetap tersedia untuk umum.

Harga

Memblokir akses publik untuk snapshot dapat diaktifkan tanpa biaya tambahan.

Daftar Isi

Pertimbangan

- Memblokir akses publik untuk snapshot tidak mencegah berbagi snapshot privat.
- Jika Anda mengaktifkan blokir akses publik untuk snapshot dalam mode Blokir semua pembagian, tidak akan mengubah izin untuk snapshot yang sudah dibagikan secara publik. Sebaliknya, pengaktifan ini mencegah snapshot agar tidak terlihat oleh publik dan dapat diakses publik. Oleh karena itu, atribut untuk snapshot ini masih menunjukkan bahwa snapshot tersebut dibagikan secara publik, meskipun tidak tersedia untuk umum.
- Jika blokir akses publik untuk snapshot diaktifkan dalam mode blokir semua pembagian, dan Anda mengubah mode ke blokir pembagian baru, atau Anda menonaktifkan blokir akses publik, semua snapshot yang sebelumnya dibagikan secara publik tidak lagi diperlakukan sebagai privat dan dapat diakses publik lagi.
- Memblokir akses publik untuk snapshot adalah pengaturan Regional. Hal ini berlaku untuk semua snapshot di Wilayah tempatnya diaktifkan. Anda harus mengaktifkan blokir akses publik untuk snapshot di setiap Wilayah yang tidak Anda inginkan untuk membagi snapshot dengan publik.

- Blok publik akses adalah pengaturan tingkat akun. Ini berlaku untuk semua pengguna, termasuk pengguna administrator, di akun. Anda tidak dapat mengaktifkan blokir akses publik untuk snapshot di tingkat organisasi.
- Memblokir akses publik untuk snapshot tidak mencegah berbagi publik AMI yang didukung EBS. Jika Anda mengaktifkan blokir akses publik untuk snapshot, pengguna masih dapat membagikan AMI yang didukung EBS secara publik. Jika AMI yang didukung EBS dibagikan secara publik, pengguna dengan akses ke AMI tersebut dapat membuat volume dari snapshot terkait. Untuk mencegah berbagi AMI Anda secara publik, aktifkan blokir akses publik untuk AMI.
- Blokir akses publik untuk snapshot tidak didukung dengan snapshot lokal aktif. AWS Outposts

Izin IAM

Secara default, pengguna tidak memiliki izin untuk bekerja dengan blokir akses publik untuk snapshot. Untuk mengizinkan pengguna bekerja dengan sumber daya ini, Anda harus membuat kebijakan IAM yang memberikan izin menggunakan sumber daya dan tindakan API tertentu. Setelah kebijakan dibuat, Anda harus menambahkan izin ke pengguna, grup, atau peran.

Untuk bekerja dengan blok akses publik untuk snapshot, pengguna memerlukan izin berikut.

- `ec2:EnableSnapshotBlockPublicAccess` — Mengaktifkan blokir akses publik untuk snapshot dan mengubah mode.
- `ec2:DisableSnapshotBlockPublicAccess` — Menonaktifkan blokir akses publik untuk snapshot.
- `ec2:GetSnapshotBlockPublicAccessState` — Melihat blokir akses publik untuk pengaturan snapshot untuk Wilayah.

Berikut ini adalah contoh kebijakan IAM. Jika beberapa izin tidak diperlukan, Anda dapat menghapusnya dari kebijakan.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "ec2:EnableSnapshotBlockPublicAccess",
            "ec2:DisableSnapshotBlockPublicAccess",
            "ec2:GetSnapshotBlockPublicAccessState"
```

```
        ],
        "Resource": "*"
    }]
}
```

Untuk memberikan akses, menambahkan izin ke pengguna, grup, atau peran Anda:

- Pengguna dan grup di AWS IAM Identity Center:

  Buat rangkaian izin. Ikuti instruksi di Buat rangkaian izin di Panduan Pengguna AWS IAM Identity Center .

- Pengguna yang dikelola di IAM melalui penyedia identitas:

  Buat peran untuk federasi identitas. Ikuti instruksi dalam Membuat peran untuk penyedia identitas pihak ketiga (federasi) di Panduan Pengguna IAM.

- Pengguna IAM:
  - Buat peran yang dapat diambil pengguna Anda. Ikuti instruksi dalam Membuat peran untuk pengguna IAM dalam Panduan Pengguna IAM.
  - (Tidak disarankan) Pasang kebijakan langsung ke pengguna atau tambahkan pengguna ke grup pengguna. Ikuti petunjuk dalam Menambahkan izin ke pengguna (konsol) dalam Panduan Pengguna IAM.

Mengaktifkanblokir akses publik untuk snapshot

Gunakan prosedur berikut untuk mengonfigurasi dan memantau blokir akses publik untuk snapshot.

Tugas

- Memgonfigurasi blokir akses publik untuk snapshot
- Melihat pengaturan guna memblokir akses publik untuk snapshot
- Menonaktifkan blokir akses publik untuk snapshot

Memgonfigurasi blokir akses publik untuk snapshot

Blokir akses publik untuk snapshot guna mencegah berbagi snapshot secara publik di Wilayah. Setelah fitur ini diaktifkan, permintaan untuk membagikan snapshot secara publik di Wilayah diblokir.

> **Important**
>
> Jika blokir akses publik untuk snapshot diaktifkan dalam mode blokir semua pembagian, dan Anda mengubah modenya ke blokir pembagian baru, semua snapshot yang sebelumnya dibagikan secara publik tidak lagi diperlakukan sebagai privat dan menjadi dapat diakses publik lagi.

Console

Untuk mengonfigurasi blokir akses publik untuk snapshot

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Dasbor EC2, lalu di Atribut akun (di sisi kanan), pilih Perlindungan dan keamanan data.
3. Di bagian Blokir akses publik untuk snapshot EBS, pilih Kelola.
4. Pilih Blokir akses publik dan kemudian pilih salah satu opsi berikut:
   - Blokir semua akses publik — Untuk memblokir semua pembagian snapshot untuk publik. Pengguna di akun tidak dapat meminta berbagi publik baru. Selain itu, snapshot yang sudah dibagikan secara publik diperlakukan sebagai privat dan tidak lagi tersedia secara publik.
   - Blokir semua pembagian publik — Untuk memblokir hanya pembagian snapshot untuk publik baru. Pengguna di akun tidak dapat meminta berbagi publik baru. Namun, snapshot yang sudah dibagikan secara publik, tetap tersedia untuk umum.
5. Pilih Perbarui.

AWS CLI

Untuk mengaktifkan atau memodifikasi blokir akses publik untuk snapshot

Gunakan perintah enable-snapshot-block-public-access. Untuk `--state`, tentukan salah satu dari nilai-nilai berikut:

- `block-all-sharing` — Untuk memblokir semua pembagian snapshot untuk publik. Pengguna di akun tidak dapat meminta berbagi publik baru. Selain itu, snapshot yang sudah dibagikan secara publik diperlakukan sebagai privat dan tidak lagi tersedia secara publik.

- `block-new-sharing` — Untuk memblokir hanya pembagian snapshot untuk publik baru. Pengguna di akun tidak dapat meminta berbagi publik baru. Namun, snapshot yang sudah dibagikan secara publik, tetap tersedia untuk publik.

```
aws ec2 enable-snapshot-block-public-access --state block-all-sharing|block-new-sharing
```

Melihat pengaturan guna memblokir akses publik untuk snapshot

Blokir akses publik dapat berada dalam salah satu status berikut untuk setiap Wilayah di akun Anda.

- Blokir semua pembagian — Semua pembagian snapshot untuk publik diblokir. Pengguna di akun tidak dapat meminta berbagi publik baru. Selain itu, snapshot yang sudah dibagikan secara publik diperlakukan sebagai privat dan tidak tersedia secara publik.
- Blokir pembagian baru — Hanya pembagian snapshot publik baru yang diblokir. Pengguna di akun tidak dapat meminta berbagi publik baru. Namun, snapshot yang sudah dibagikan secara publik, tetap tersedia untuk publik.
- Tidak diblokir — Pembagian publik tidak diblokir. Pengguna dapat berbagi snapshot secara publik.

Console

Untuk melihat pengaturan guna memblokir akses publik untuk snapshot

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Dasbor EC2, lalu di Atribut akun (di sisi kanan), pilih Perlindungan dan keamanan data.
3. Bagian Blokir akses publik untuk snapshot EBS menunjukkan pengaturan saat ini.

AWS CLI

Untuk melihat pengaturan guna memblokir akses publik untuk snapshot

Gunakan perintah get-snapshot-block-public-access-state.

```
aws ec2 get-snapshot-block-public-access-state
```

Menonaktifkan blokir akses publik untuk snapshot

Nonaktifkan blokir akses publik untuk snapshot guna mengizinkan berbagi snapshot secara publik. Setelah fitur ini dinonaktifkan, pengguna dapat membagikan snapshot secara publik di Wilayah.

> **Important**
>
> Jika blokir akses publik untuk snapshot diaktifkan dalam mode blokir semua pembagian, dan Anda menonaktifkan blokir akses publik, semua snapshot yang sebelumnya dibagikan secara publik tidak lagi diperlakukan sebagai privat dan dapat diakses publik lagi.

Console

Untuk menonaktifkan blokir akses publik untuk snapshot

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Dasbor EC2, lalu di Atribut akun (di sisi kanan), pilih Perlindungan dan keamanan data.
3. Di bagian Blokir akses publik untuk snapshot EBS, pilih Kelola.
4. Hapus Blokir akses publik dan pilih Perbarui.

AWS CLI

Untuk menonaktifkan blokir akses publik untuk snapshot

Gunakan perintah disable-snapshot-block-public-access.

```
aws ec2 disable-snapshot-block-public-access
```

Monitor memblokir akses publik untuk snapshot menggunakan Amazon EventBridge

Amazon EBS memancarkan peristiwa terkait guna memblokir akses publik untuk snapshot. Anda dapat menggunakan AWS Lambda dan Amazon EventBridge untuk menangani pemberitahuan acara secara terprogram. Peristiwa dipancarkan atas dasar upaya terbaik. Untuk informasi selengkapnya, lihat Panduan EventBridge Pengguna Amazon.

Peristiwa berikut dipancarkan:

- Mengaktifkan blokir akses publik untuk snapshot dalam mode blokir semua pembagian

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "EBS Snapshot Block Public Access Enabled",
  "source": "aws.ec2",
  "account": "123456789012",
  "time": "2019-05-31T21:49:54Z",
  "region": "us-east-1",
  "detail": {
    "SnapshotBlockPublicAccessState": "block-all-sharing",
    "message": "Block Public Access was successfully enabled in 'block-all-sharing'
 mode"
  }
}
```

- Mengaktifkan blokir akses publik untuk snapshot dalam mode blokir pembagian baru

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "EBS Snapshot Block Public Access Enabled",
  "source": "aws.ec2",
  "account": "123456789012",
  "time": "2019-05-31T21:49:54Z",
  "region": "us-east-1",
  "detail": {
    "SnapshotBlockPublicAccessState": "block-new-sharing",
    "message": "Block Public Access was successfully enabled in 'block-new-sharing'
 mode"
  }
}
```

- Menonaktifkan blokir akses publik untuk snapshot

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "EBS Snapshot Block Public Access Disabled",
  "source": "aws.ec2",
```

```
  "account": "123456789012",
  "time": "2019-05-31T21:49:54Z",
  "region": "us-east-1",
  "detail": {
    "SnapshotBlockPublicAccessState": "unblocked",
    "message": "Block Public Access was successfully disabled"
  }
}
```

Recycle Bin untuk snapshot

Keranjang Sampah adalah fitur pemulihan data yang memungkinkan Anda memulihkan snapshot dan AMI yang didukung Amazon EBS yang terhapus secara tidak sengaja. Saat menggunakan Keranjang Sampah, jika sumber daya Anda dihapus, sumber daya tersebut dipertahankan di Keranjang Sampah untuk jangka waktu yang Anda tentukan sebelum dihapus secara permanen.

Anda dapat memulihkan sumber daya dari Keranjang Sampah kapan saja sebelum periode retensi berakhir. Setelah memulihkan sumber daya dari Keranjang Sampah, sumber daya tersebut akan dihapus dari Keranjang Sampah dan Anda dapat menggunakannya dengan cara yang sama seperti menggunakan sumber daya lain dari tipe tersebut di akun Anda. Jika periode retensi berakhir dan sumber daya tidak dipulihkan, sumber daya tersebut akan dihapus secara permanen dari Keranjang Sampah dan tidak lagi tersedia untuk pemulihan.

Snapshot di Keranjang Sampah ditagih dengan tarif yang sama dengan snapshot reguler di akun Anda. Tidak ada biaya tambahan untuk menggunakan Keranjang Sampah dan aturan penyimpanan. Untuk informasi selengkapnya, lihat harga Amazon EBS.

Untuk informasi selengkapnya, lihat Recycle Bin.

Topik

- Izin untuk bekerja dengan snapshot di Keranjang Sampah
- Lihat snapshot di Keranjang Sampah
- Mengembalikan snapshot dari Keranjang Sampah

Izin untuk bekerja dengan snapshot di Keranjang Sampah

Secara default, pengguna tidak memiliki izin untuk bekerja dengan snapshot yang ada di Keranjang Sampah. Untuk mengizinkan pengguna bekerja dengan sumber daya ini, Anda harus membuat

kebijakan IAM yang memberikan izin menggunakan sumber daya dan tindakan API tertentu. Setelah kebijakan dibuat, Anda harus menambahkan izin ke pengguna, grup, atau peran.

Untuk melihat dan memulihkan snapshot yang ada di Keranjang Sampah, pengguna harus memiliki izin berikut:

- `ec2:ListSnapshotsInRecycleBin`
- `ec2:RestoreSnapshotFromRecycleBin`

Untuk mengelola tanda untuk snapshot di Keranjang Sampah, pengguna memerlukan izin tambahan berikut.

- `ec2:CreateTags`
- `ec2:DeleteTags`

Untuk menggunakan konsol Keranjang Sampah, pengguna memerlukan `ec2:DescribeTags` izin.

Berikut ini adalah contoh kebijakan IAM. Ini termasuk izin `ec2:DescribeTags` untuk pengguna konsol, dan itu termasuk izin `ec2:CreateTags` dan `ec2:DeleteTags` untuk mengelola tag. Jika beberapa izin tidak diperlukan, Anda dapat menghapusnya dari kebijakan.

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Action": [
            "ec2:ListSnapshotsInRecycleBin",
            "ec2:RestoreSnapshotFromRecycleBin"
        ],
        "Resource": "*"
      },
      {
        "Effect": "Allow",
        "Action": [
            "ec2:CreateTags",
            "ec2:DeleteTags",
            "ec2:DescribeTags"
        ],
        "Resource": "arn:aws:ec2:Region:account-id:snapshot/*"
```

```
      },

    ]
}
```

Untuk memberikan akses, menambahkan izin ke pengguna, grup, atau peran Anda:

- Pengguna dan grup di AWS IAM Identity Center:

  Buat rangkaian izin. Ikuti instruksi di Buat rangkaian izin di Panduan Pengguna AWS IAM Identity Center .
- Pengguna yang dikelola di IAM melalui penyedia identitas:

  Buat peran untuk federasi identitas. Ikuti instruksi dalam Membuat peran untuk penyedia identitas pihak ketiga (federasi) di Panduan Pengguna IAM.
- Pengguna IAM:
  - Buat peran yang dapat diambil pengguna Anda. Ikuti instruksi dalam Membuat peran untuk pengguna IAM dalam Panduan Pengguna IAM.
  - (Tidak disarankan) Pasang kebijakan langsung ke pengguna atau tambahkan pengguna ke grup pengguna. Ikuti instruksi dalam Menambahkan izin ke pengguna (konsol) dalam Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang izin yang diperlukan untuk menggunakan Recycle Bin, lihat Izin IAM yang diperlukan.

Lihat snapshot di Keranjang Sampah

Saat snapshot ada di Keranjang Sampah, Anda dapat melihat informasi terbatas tentangnya, termasuk:

- ID snapshot.
- Deskripsi snapshot.
- ID volume tempat snapshot dibuat.
- Tanggal dan waktu snapshot dihapus dan masuk Keranjang Sampah.
- Tanggal dan waktu ketika periode retensi kedaluwarsa. Snapshot akan dihapus secara permanen dari Keranjang Sampah saat ini.

Anda dapat melihat snapshot di Keranjang Sampah menggunakan salah satu metode berikut.

Recycle Bin console

Untuk melihat snapshot di Keranjang Sampah menggunakan konsol

1. Buka konsol Keranjang Sampah di https://console.aws.amazon.com/rbin/home/
2. Di panel navigasi, pilih Keranjang Sampah.
3. Kisi mencantumkan semua snapshot yang saat ini ada di Keranjang Sampah. Untuk melihat detail AMI tertentu, pilih di kisi, dan pilih Tindakan, Lihat detail.

AWS CLI

Untuk melihat snapshot di Recycle Bin menggunakan AWS CLI

Gunakan AWS CLI perintah list-snapshots-in-recycle-bin. Sertakan opsi `--snapshot-id` untuk melihat snapshot tertentu. Atau hilangkan opsi `--snapshot-id` untuk melihat semua snapshot di Keranjang Sampah.

```
$ C:\> aws ec2 list-snapshots-in-recycle-bin --snapshot-id snapshot_id
```

Misalnya, perintah berikut memberikan informasi tentang snapshot `snap-01234567890abcdef` di Keranjang Sampah.

```
$ C:\> aws ec2 list-snapshots-in-recycle-bin --snapshot-id snap-01234567890abcdef
```

Contoh output:

```
{
    "SnapshotRecycleBinInfo": [
        {
            "Description": "Monthly data backup snapshot",
            "RecycleBinEnterTime": "2021-12-01T13:00:00.000Z",
            "RecycleBinExitTime": "2021-12-15T13:00:00.000Z",
            "VolumeId": "vol-abcdef09876543210",
            "SnapshotId": "snap-01234567890abcdef"
        }
    ]
}
```

Mengembalikan snapshot dari Keranjang Sampah

Anda tidak dapat menggunakan snapshot dengan cara apa pun saat berada di Keranjang Sampah. Untuk menggunakan AMI, Anda harus memulihkannya terlebih dahulu. Saat Anda memulihkan snapshot dari Keranjang Sampah, snapshot segera tersedia untuk digunakan, dan akan dihapus dari Keranjang Sampah. Anda dapat menggunakan AMI yang dipulihkan dengan cara yang sama seperti Anda menggunakan AMI lainnya di akun Anda.

Anda dapat memulihkan snapshot dari Keranjang Sampah menggunakan salah satu metode berikut.

Recycle Bin console

Untuk memulihkan snapshot dari Keranjang Sampah menggunakan konsol

1. Buka konsol Keranjang Sampah di https://console.aws.amazon.com/rbin/home/
2. Di panel navigasi, pilih Keranjang Sampah.
3. Kisi mencantumkan semua snapshot yang saat ini ada di Keranjang Sampah. Pilih snapshot yang akan dipulihkan, lalu pilih Pulihkan.
4. Saat diminta, pilih Pulihkan.

AWS CLI

Untuk mengembalikan snapshot yang dihapus dari Recycle Bin menggunakan AWS CLI

Gunakan AWS CLI perintah restore-snapshot-from-recycle-bin. Untuk `--snapshot-id`, tentukan ID snapshot yang akan dipulihkan.

```
$ C:\> aws ec2 restore-snapshot-from-recycle-bin --snapshot-id snapshot_id
```

Misalnya, perintah berikut memulihkan snapshot `snap-01234567890abcdef` dari Keranjang Sampah.

```
$ C:\> aws ec2 restore-snapshot-from-recycle-bin --snapshot-id
 snap-01234567890abcdef
```

Contoh output:

```
{
```

```
    "SnapshotId": "snap-01234567890abcdef",
    "Description": "Monthly data backup snapshot",
    "Encrypted": false,
    "OwnerId": "111122223333",
    "Progress": "100%",
    "StartTime": "2021-12-01T13:00:00.000000+00:00",
    "State": "recovering",
    "VolumeId": "vol-ffffffff",
    "VolumeSize": 30
}
```

Snapshot lokal Amazon EBS di Outposts

Snapshot Amazon EBS adalah point-in-time salinan volume EBS Anda.

Secara default, snapshot dari volume EBS pada Outpost disimpan di Amazon S3 di Wilayah Outpost. Anda juga dapat menggunakan snapshot lokal Amazon EBS di Outposts untuk menyimpan snapshot volume di Outpost secara lokal di Amazon S3 di Outposts itu sendiri. Hal ini memastikan bahwa data snapshot berada di Outpost, dan on-premise Anda. Selain itu, Anda dapat menggunakan kebijakan dan izin AWS Identity and Access Management (IAM) untuk menyiapkan kebijakan penegakan residensi data agar data snapshot tidak keluar dari Outpost. Ini sangat berguna jika Anda tinggal di negara atau wilayah yang belum dilayani oleh suatu AWS Wilayah dan yang memiliki persyaratan residensi data.

Topik ini memberikan informasi tentang bekerja dengan Amazon EBS snapshot lokal di Outposts. Untuk informasi selengkapnya tentang snapshot Amazon EBS dan tentang bekerja dengan snapshot di suatu AWS Wilayah, lihat. Snapshot Amazon EBS

Untuk informasi selengkapnya AWS Outposts, lihat AWS Outposts Fitur dan Panduan AWS Outposts Pengguna. Untuk informasi harga, lihat harga AWS Outposts.

Topik

- Pertanyaan umum
- Prasyarat
- Pertimbangan
- Mengendalikan akses dengan IAM
- Bekerja dengan snapshot lokal

Pertanyaan umum

1. Apa itu snapshot lokal?

   Secara default, snapshot volume Amazon EBS di Outposts disimpan di Amazon S3 di Wilayah Outposts. Jika Outposts disediakan dengan Amazon S3 on Outposts, Anda dapat memilih untuk menyimpan snapshot secara lokal di Outpost itu sendiri. Snapshot lokal bersifat inkremental, yang berarti hanya blok pada volume yang berubah setelah snapshot terbaru Anda disimpan. Anda dapat menggunakan snapshot ini untuk memulihkan volume pada Outposts yang sama sebagai snapshot setiap saat. Untuk informasi selengkapnya tentang snapshot Amazon EBS, lihat Snapshot Amazon EBS.

2. Mengapa saya harus menggunakan snapshot lokal?

   Snapshot adalah cara mudah untuk mencadangkan data Anda. Dengan snapshot lokal, semua data snapshot Anda disimpan secara lokal di Outpost. Ini berarti bahwa itu tidak meninggalkan tempat Anda. Ini sangat berguna jika Anda tinggal di negara atau wilayah yang belum dilayani oleh suatu AWS Wilayah dan yang memiliki persyaratan tempat tinggal.

   Selain itu, menggunakan snapshot lokal dapat membantu untuk mengurangi bandwidth yang digunakan untuk komunikasi antara Wilayah dan Outposts di bandwidth yang dibatasi lingkungan.

3. Bagaimana cara memberlakukan residensi data snapshot di Outposts?

   Anda dapat menggunakan kebijakan AWS Identity and Access Management (IAM) untuk mengontrol izin yang dimiliki kepala sekolah (AWS akun, pengguna IAM, dan peran IAM) saat bekerja dengan snapshot lokal dan untuk menerapkan residensi data. Anda dapat membuat kebijakan yang mencegah prinsipal membuat snapshot dari volume dan instance Outpost dan menyimpan snapshot di Region. AWS Saat ini, menyalin snapshot dan gambar dari Outpost ke Wilayah tidak didukung. Untuk informasi selengkapnya, lihat Mengendalikan akses dengan IAM.

4. Apakah snapshot lokal multivolume dan crash-consistent didukung?

   Ya, Anda dapat membuat snapshot lokal multivolume dan crash-consistent dari instans di Outposts.

5. Bagaimana cara membuat snapshot lokal?

   Anda dapat membuat snapshot secara manual menggunakan AWS Command Line Interface (AWS CLI) atau konsol Amazon EC2. Untuk informasi selengkapnya, lihat Bekerja dengan snapshot lokal. Anda juga dapat mengotomatiskan siklus hidup snapshot lokal menggunakan

Amazon Data Lifecycle Manager. Untuk informasi selengkapnya, lihat Mengotomatisasi snapshot di Outposts.

6. Dapatkah saya membuat, menggunakan, atau menghapus snapshot lokal jika Outposts saya kehilangan koneksi ke Wilayah?

Tidak. Outpost haruss memiliki konektivitas dengan Wilayah karena Wilayah menyediakan akses, otorisasi, pembuatan log, dan layanan pemantauan yang sangat penting untuk kondisi snapshot Anda. Jika tidak ada konektivitas, Anda tidak dapat membuat snapshot lokal baru, membuat volume atau meluncurkan instans dari snapshot lokal yang ada, atau menghapus snapshot lokal.

7. Seberapa cepat kapasitas penyimpanan Amazon S3 tersedia setelah menghapus snapshot lokal?

Kapasitas penyimpanan Amazon S3 tersedia dalam waktu 72 jam setelah menghapus snapshot lokal dan volume yang mereferensikannya.

8. Bagaimana saya dapat memastikan bahwa saya tidak kehabisan kapasitas Amazon S3 di Outposts saya?

Kami menyarankan Anda menggunakan CloudWatch alarm Amazon untuk memantau kapasitas penyimpanan Amazon S3 Anda, dan menghapus snapshot dan volume yang tidak perlu lagi Anda hindari kehabisan kapasitas penyimpanan. Jika Anda menggunakan Amazon Data Lifecycle Manager untuk mengotomatisasi siklus hidup snapshot lokal, pastikan bahwa kebijakan penyimpanan snapshot Anda tidak mempertahankan snapshot lebih lama dari yang diperlukan.

9. Apa yang terjadi jika saya kehabisan kapasitas Amazon S3 lokal di Outposts saya?

Jika Anda kehabisan kapasitas Amazon S3 lokal di Outposts Anda, Amazon Data Lifecycle Manager tidak akan berhasil membuat snapshot lokal di Outposts. Amazon Data Lifecycle Manager akan mencoba membuat snapshot lokal di Outposts, tetapi snapshot segera bertransisi ke status `error` dan akhirnya dihapus oleh Amazon Data Lifecycle Manager. Kami menyarankan Anda menggunakan CloudWatch metrik `SnapshotsCreateFailed` Amazon untuk memantau kebijakan siklus hidup snapshot Anda untuk kegagalan pembuatan snapshot. Untuk informasi selengkapnya, lihat Pantau kebijakan Anda menggunakan Amazon CloudWatch.

10. Dapatkah saya menggunakan snapshot lokal dan AMI yang didukung oleh snapshot lokal dengan Instans Spot dan Armada Spot?

Tidak, Anda tidak dapat menggunakan snapshot lokal atau AMI yang didukung oleh snapshot lokal untuk meluncurkan Instans Spot atau Armada Spot.

11. Dapatkah saya menggunakan snapshot lokal dan AMI yang didukung oleh snapshot lokal dengan Amazon EC2 Auto Scaling?

Ya, Anda dapat menggunakan snapshot lokal dan AMI yang didukung oleh snapshot lokal untuk meluncurkan grup Auto Scaling di subnet yang ada di Outposts yang sama sebagai snapshot. Peran terkait layanan grup Amazon EC2 Auto Scaling harus memiliki izin untuk menggunakan kunci KMS yang digunakan untuk mengenkripsi snapshot.

Anda tidak dapat menggunakan snapshot lokal atau AMI yang didukung oleh snapshot lokal untuk meluncurkan grup Auto Scaling di suatu Wilayah. AWS

Prasyarat

Untuk menyimpan snapshot di Outposts, Anda harus memiliki Outposts yang disediakan dengan Amazon S3 on Outposts. Untuk informasi selengkapnya tentang Amazon S3 on Outposts, lihat Menggunakan Amazon S3 on Outposts di Panduan Pengguna Amazon Simple Storage Service.

Pertimbangan

Ingatlah hal-hal berikut ini saat bekerja dengan snapshot lokal.

- Outposts harus memiliki konektivitas ke AWS Wilayah mereka untuk menggunakan snapshot lokal.
- Metadata snapshot disimpan di AWS Wilayah yang terkait dengan Pos Luar. Hal ini tidak mencakup data snapshot.
- Snapshot yang disimpan di Outposts dienkripsi secara default. Snapshot yang tidak dienkripsi tidak didukung. Snapshots yang dibuat di Outposts dan snapshot yang disalin ke Outposts dienkripsi menggunakan kunci KMS default untuk Wilayah atau kunci KMS yang berbeda yang Anda tentukan pada saat diminta.
- Ketika Anda membuat volume di Outposts dari snapshot lokal, Anda tidak dapat mengenkripsi ulang volume menggunakan kunci KMS yang berbeda. Volume yang dibuat dari snapshot lokal harus dienkripsi menggunakan kunci KMS yang sama dengan snapshot sumber.
- Setelah Anda menghapus snapshot lokal dari Outposts, kapasitas penyimpanan Amazon S3 yang digunakan oleh snapshot yang dihapus tersedia dalam waktu 72 jam. Untuk informasi selengkapnya, lihat Menghapus snapshot lokal.
- Anda tidak dapat mengekspor snapshot lokal dari Outposts.
- Anda tidak dapat mengaktifkan pemulihan snapshot cepat untuk snapshot lokal.

- API langsung EBS tidak didukung dengan snapshot lokal.
- Anda tidak dapat menyalin snapshot lokal atau AMI dari Pos Luar ke AWS Wilayah, dari satu Pos Luar ke pos lain, atau dalam Pos Luar. Namun, Anda dapat menyalin snapshot dari Wilayah AWS ke Outposts. Untuk informasi selengkapnya, lihat Salin snapshot dari AWS Wilayah ke Pos Terdepan.
- Saat menyalin snapshot dari AWS Wilayah ke Pos Luar, data ditransfer melalui tautan layanan. Menyalin banyak snapshot secara bersamaan dapat memengaruhi layanan lain yang berjalan di Outposts.
- Anda tidak dapat membagikan snapshot lokal.
- Anda harus menggunakan kebijakan IAM untuk memastikan bahwa persyaratan residensi data Anda terpenuhi. Untuk informasi selengkapnya, lihat Mengendalikan akses dengan IAM.
- Snapshot lokal bersifat cadangan inkremental. Hanya blok dalam volume yang telah berubah setelah snapshot terakhir yang disimpan. Setiap snapshot berisi semua informasi yang diperlukan untuk memulihkan data Anda (dari saat ketika snapshot diambil) ke volume EBS baru. Untuk informasi selengkapnya, lihat Cara kerja snapshot.
- Anda tidak dapat menggunakan kebijakan IAM untuk menegakkan residensi data dan tindakan. CopySnapshotCopyImage

Mengendalikan akses dengan IAM

Anda dapat menggunakan kebijakan AWS Identity and Access Management (IAM) untuk mengontrol izin yang dimiliki kepala sekolah (AWS akun, pengguna IAM, dan peran IAM) saat bekerja dengan snapshot lokal. Berikut ini adalah contoh kebijakan yang dapat Anda gunakan untuk memberikan atau menolak izin untuk melakukan tindakan tertentu dengan snapshot lokal.

> **Important**
>
> Menyalin snapshot dan citra dari Outposts ke Wilayah saat ini tidak didukung. Akibatnya, saat ini Anda tidak dapat menggunakan kebijakan IAM untuk menegakkan residensi data dan tindakan. CopySnapshotCopyImage

Topik

- Memberlakukan residensi data untuk snapshot
- Mencegah pengguna utama menghapus snapshot lokal

Memberlakukan residensi data untuk snapshot

Contoh kebijakan berikut mencegah semua prinsipal membuat snapshot dari volume dan instance di Outpost `arn:aws:outposts:us-east-1:123456789012:outpost/op-1234567890abcdef` dan menyimpan data snapshot di Region. AWS Pengguna utama masih dapat membuat snapshot lokal. Kebijakan ini memastikan bahwa semua snapshot tetap berada di Outposts.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "ec2:CreateSnapshot",
                "ec2:CreateSnapshots"
            ],
            "Resource": "arn:aws:ec2:us-east-1::snapshot/*",
            "Condition": {
                "StringEquals": {
                    "ec2:SourceOutpostArn": "arn:aws:outposts:us-
east-1:123456789012:outpost/op-1234567890abcdef0"
                },
                "Null": {
                    "ec2:OutpostArn": "true"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateSnapshot",
                "ec2:CreateSnapshots"
            ],
            "Resource": "*"
        }
    ]
}
```

Mencegah pengguna utama menghapus snapshot lokal

Kebijakan contoh berikut mencegah semua pengguna utama menghapus snapshot lokal yang disimpan di Outposts `arn:aws:outposts:us-east-1:123456789012:outpost/op-1234567890abcdef0`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "ec2:DeleteSnapshot"
            ],
            "Resource": "arn:aws:ec2:us-east-1::snapshot/*",
            "Condition": {
                "StringEquals": {
                    "ec2:OutpostArn": "arn:aws:outposts:us-east-1:123456789012:outpost/
op-1234567890abcdef0"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DeleteSnapshot"
            ],
            "Resource": "*"
        }
    ]
}
```

Bekerja dengan snapshot lokal

Bagian berikut menjelaskan cara menggunakan snapshot lokal.

Topik

- Aturan untuk menyimpan snapshot
- Membuat snapshot lokal dari volume di Outposts
- Membuat snapshot lokal multi-volume dari instans di Outposts

- Membuat AMI dari snapshot lokal
- Salin snapshot dari AWS Wilayah ke Pos Terdepan
- Salin AMI dari AWS Wilayah ke Pos Terdepan
- Membuat volume dari snapshot lokal
- Meluncurkan instans dari AMI didukung oleh snapshot lokal
- Menghapus snapshot lokal
- Mengotomatisasi snapshot di Outposts

Aturan untuk menyimpan snapshot

Aturan berikut berlaku untuk penyimpanan snapshot:

- Jika snapshot terbaru dari volume disimpan di Outpost, semua snapshot berturut-turut harus disimpan pada Outposts yang sama.
- Jika snapshot terbaru dari volume disimpan di AWS Wilayah, maka semua snapshot berturut-turut harus disimpan di Wilayah yang sama. Untuk mulai membuat snapshot lokal dari volume tersebut, lakukan hal berikut:

  1. Buat snapshot volume di AWS Wilayah.
  2. Salin snapshot ke Outpost dari Region. AWS
  3. Buat volume baru dari snapshot lokal.
  4. Lampirkan volume ke instans di Outposts.

  Untuk volume baru di Outposts, snapshot berikutnya dapat disimpan di Outposts atau Wilayah AWS . Semua snapshot berturut-turut kemudian harus disimpan di lokasi yang sama.
- Snapshot lokal, termasuk snapshot yang dibuat di Outpost dan snapshot yang disalin ke Outpost dari AWS Wilayah, hanya dapat digunakan untuk membuat volume di Outpost yang sama.
- Jika Anda membuat volume di Outposts dari snapshot di Wilayah, semua snapshot berturut-turut dari volume baru tersebut harus berada di Wilayah yang sama.
- Jika Anda membuat volume di Outposts dari snapshot lokal, semua snapshot berturut-turut dari volume baru tersebut harus berada di Outposts yang sama.

Membuat snapshot lokal dari volume di Outposts

Anda dapat membuat snapshot lokal dari volume di Outposts. Anda dapat memilih untuk menyimpan snapshot pada Outposts yang sama sebagai volume sumber, atau di Wilayah untuk Outposts.

Snapshot lokal hanya dapat digunakan untuk membuat volume di Outposts yang sama.

Anda dapat membuat snapshot lokal dari volume di Outposts menggunakan salah satu metode berikut.

Console

Untuk membuat snapshot lokal dari volume di Outposts

Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.

1. Di panel navigasi, pilih Volume.
2. Pilih volume di Outposts, lalu pilih Tindakan, Buat Snapshot.
3. (Opsional) Untuk Deskripsi, masukkan deskripsi singkat untuk snapshot tersebut.
4. Untuk Tujuan snapshot, pilih AWS Outposts. Snapshot akan dibuat di Outposts yang sama dengan volume sumber. Bidang ARN Outposts menunjukkan Amazon Resource Name (ARN) dari Outposts tujuan.
5. (Opsional) Pilih Tambahkan Tanda untuk menambahkan tanda ke snapshot Anda. Untuk setiap tanda, berikan kunci tanda dan nilai tanda.
6. Pilih Buat Snapshot.

Command line

Untuk membuat snapshot lokal dari volume di Outposts

Menggunakan perintah create-snapshot. Tentukan ID volume tempat pembuatan snapshot, dan ARN Outposts tujuan untuk menyimpan snapshot. Jika Anda menghilangkan ARN Outpost, snapshot disimpan di Region for the AWS Outpost.

Misalnya, perintah berikut membuat snapshot volume `vol-1234567890abcdef0` lokal, dan menyimpan snapshot di Outposts `arn:aws:outposts:us-east-1:123456789012:outpost/op-1234567890abcdef0`.

```
$ aws ec2 create-snapshot --volume-id vol-1234567890abcdef0 --outpost-arn
 arn:aws:outposts:us-east-1:123456789012:outpost/op-1234567890abcdef0 --description
 "single volume local snapshot"
```

Membuat snapshot lokal multi-volume dari instans di Outposts

Anda dapat membuat snapshot lokal multivolume dan crash-consistent dari instans di Outposts. Anda dapat memilih untuk menyimpan snapshot di Outposts yang sama sebagai instans sumber, atau di Wilayah untuk Outposts.

Snapshot lokal multi-volume hanya dapat digunakan untuk membuat volume di Outposts yang sama.

Anda dapat membuat snapshot lokal multi-volume dari instans di Outposts menggunakan salah satu metode berikut.

Console

Untuk membuat snapshot lokal multi-volume dari instans di Outposts

Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.

1. Di panel navigasi, pilih Snapshot.
2. Pilih Buat Snapshot.
3. Untuk Pilih tipe sumber daya, pilih Instans.
4. Untuk ID Instans, pilih instans di Outposts untuk membuat snapshot.
5. (Opsional) Untuk Deskripsi, masukkan deskripsi singkat untuk snapshot tersebut.
6. Untuk Tujuan snapshot, pilih AWS Outpost. Snapshot akan dibuat pada Outposts yang sama dengan instans sumber. ARN Outposts menunjukkan ARN dari Outposts tujuan.
7. Untuk mengecualikan volume root instans dari set snapshot multi-volume, pilih Kecualikan volume root. Jika Anda melakukan ini, Amazon EBS tidak akan membuat snapshot dari volume root instans.
8. Untuk mengecualikan volume data tertentu dari set snapshot multi-volume, pilih Kecualikan volume data tertentu. Bagian Volume data terlampir mencantumkan semua volume data yang saat ini dilampirkan ke instans yang dipilih.

Di bagian Volume data terlampir, batalkan pilihan volume data yang akan dikecualikan dari set snapshot multi-volume. Hanya volume yang tetap tidak dipilih yang akan disertakan dalam set snapshot multi-volume.

9. (Opsional) Untuk secara otomatis menyalin tanda dari volume sumber ke snapshot yang sesuai, untuk Salin tanda dari volume sumber, pilih Salin tanda. Tindakan ini akan mengatur metadata snapshot—seperti kebijakan akses, informasi lampiran, dan alokasi biaya—agar cocok dengan volume sumber.
10. (Opsional) Untuk menetapkan tanda kustom tambahan ke snapshot, di bagian Tanda, pilih Tambahkan tanda, lalu masukkan pasangan kunci-nilai. Anda dapat menambahkan hingga 50 tanda.
11. Pilih Buat Snapshot.

    Selama pembuatan snapshot, snapshot dikelola bersama. Jika salah satu snapshot dalam set volume gagal, snapshot lainnya dalam set volume dipindahkan ke status kesalahan.

Command line

Untuk membuat snapshot lokal multi-volume dari instans di Outposts

Gunakan perintah create-snapshots. Tentukan ID instans tempat pembuatan snapshot, dan ARN Outposts tujuan untuk menyimpan snapshot. Jika Anda menghilangkan ARN Outpost, snapshot disimpan di Region for the AWS Outpost.

Misalnya, perintah berikut membuat snapshot dari volume yang dilampirkan ke instans `i-1234567890abcdef0` dan menyimpan snapshot di Outpost `arn:aws:outposts:us-east-1:123456789012:outpost/op-1234567890abcdef0`.

```
$ aws ec2 create-snapshots --instance-specification InstanceId=i-1234567890abcdef0
 --outpost-arn arn:aws:outposts:us-east-1:123456789012:outpost/op-1234567890abcdef0
 --description "multi-volume local snapshots"
```

Membuat AMI dari snapshot lokal

Anda dapat membuat Amazon Machine Image (AMI) menggunakan kombinasi snapshot lokal dan snapshot yang disimpan di Wilayah Outposts. Misalnya, jika Anda memiliki Outposts di `us-east-1`, Anda dapat membuat AMI dengan volume data yang didukung oleh snapshot lokal pada Outposts itu, dan volume root yang didukung oleh snapshot di Wilayah `us-east-1`.

Note

- Anda tidak dapat membuat AMI yang mencakup pencadangan snapshot yang disimpan di banyak Outposts.
- Saat ini Anda tidak dapat membuat AMI langsung dari instans di Outposts CreateImagemenggunakan API atau konsol Amazon EC2 untuk Outposts yang diaktifkan dengan Amazon S3 di Outposts.
- AMI yang didukung oleh snapshot lokal hanyadapat digunakan untuk meluncurkan instans pada Outposts yang sama.

Untuk membuat AMI di Outposts dari snapshot di Wilayah

1. Salin snapshot dari Wilayah ke Outposts. Untuk informasi selengkapnya, lihat Salin snapshot dari AWS Wilayah ke Pos Terdepan.
2. Gunakan konsol Amazon EC2 atau perintah register-image untuk membuat AMI menggunakan salinan snapshot di Outposts. Untuk informasi selengkapnya, lihat Membuat AMI dari suatu snapshot.

Untuk membuat AMI di Outposts dari instans di Outposts

1. Buat snapshot dari instans di Outposts dan simpan snapshot di Outposts. Untuk informasi selengkapnya, lihat Membuat snapshot lokal multi-volume dari instans di Outposts.
2. Gunakan konsol Amazon EC2 atau perintah register-image untuk membuat AMI menggunakan snapshot lokal. Untuk informasi selengkapnya, lihat Membuat AMI dari suatu snapshot.

Untuk membuat AMI di Wilayah dari instans di Outposts

1. Membuat snapshot dari instans di Outposts dan menyimpan snapshot di Wilayah. Untuk informasi lebih lanjut, lihat Membuat snapshot lokal dari volume di Outposts atau Membuat snapshot lokal multi-volume dari instans di Outposts.
2. Gunakan konsol Amazon EC2 atau perintah register-image untuk membuat AMI menggunakan salinan snapshot di Wilayah. Untuk informasi selengkapnya, lihat Membuat AMI dari suatu snapshot.

Salin snapshot dari AWS Wilayah ke Pos Terdepan

Anda dapat menyalin snapshot dari AWS Wilayah ke Pos Luar. Anda dapat melakukannya hanya jika snapshot berada di Wilayah untuk Outposts. Jika snapshot berada di Wilayah lain, Anda harus terlebih dahulu menyalin snapshot ke Wilayah untuk Outposts, kemudian menyalinnya dari Wilayah tersebut ke Outposts.

> **Note**
>
> Anda tidak dapat menyalin snapshot lokal dari Outposts ke Wilayah, dari satu Outposts ke Outposts yang lain, atau dalam Outposts yang sama.

Anda dapat menyalin snapshot dari Wilayah ke Outposts menggunakan salah satu metode berikut.

Console

Untuk menyalin snapshot dari AWS Wilayah ke Pos Terdepan

Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.

1. Di panel navigasi, pilih Snapshot.
2. Pilih snapshot di Wilayah, dan pilih Tindakan, Salin.
3. Untuk Wilayah Tujuan, pilih Wilayah untuk Outposts tujuan.
4. Untuk Tujuan Snapshot, pilih AWS Outposts.

   Bidang Tujuan Snapshot hanya muncul jika Anda memiliki Outposts di Wilayah tujuan yang dipilih. Jika bidang tidak muncul, Anda tidak memiliki Outposts di Wilayah tujuan yang dipilih.
5. Untuk ARN Outposts Tujuan, masukkan ARN Outposts untuk menyalin snapshot.
6. (Opsional) Untuk Deskripsi, masukkan deskripsi singkat untuk snapshot yang disalin tersebut.
7. Enkripsi diaktifkan secara default untuk salinan snapshot. Enkripsi tidak dapat dinonaktifkan. Untuk Kunci KMS, pilih tombol KMS untuk digunakan.
8. Pilih Salin.

Command line

Untuk menyalin sebuah snapshot dari satu Wilayah ke satu Outposts

Gunakan perintah copy-snapshot. Tentukan ID snapshot yang akan disalin, Wilayah tempat untuk menyalin snapshot, dan ARN Outposts tujuan.

Misalnya, perintah berikut menyalin snapshot `snap-1234567890abcdef0` dari Wilayah `us-east-1` ke Outposts `arn:aws:outposts:us-east-1:123456789012:outpost/op-1234567890abcdef0`.

```
$ aws ec2 copy-snapshot --source-region us-east-1 --source-snapshot-
id snap-1234567890abcdef0 --destination-outpost-arn arn:aws:outposts:us-
east-1:123456789012:outpost/op-1234567890abcdef0 --description "Local snapshot copy"
```

Salin AMI dari AWS Wilayah ke Pos Terdepan

Anda dapat menyalin AMI dari AWS Wilayah ke Pos Terdepan. Ketika Anda menyalin AMI dari Wilayah ke Outposts, semua snapshot yang terkait dengan AMI disalin dari Wilayah ke Outposts.

Anda dapat menyalin AMI dari Wilayah ke Outposts hanya jika snapshot yang terkait dengan AMI berada di Wilayah untuk Outposts. Jika snapshot berada di Wilayah lain, Anda harus terlebih dahulu menyalin AMI ke Wilayah untuk Outposts, kemudian menyalinnya dari Wilayah tersebut ke Outposts.

> **Note**
>
> Anda tidak dapat menyalin AMI dari Outposts ke suatu Wilayah, dari satu Outposts ke Outposts yang lain, atau dalam satu Outposts

Anda dapat menyalin AMI dari Wilayah ke Pos Luar menggunakan AWS CLI satu-satunya.

Command line

Menyalin AMI dari suatu Wilayah ke Outposts

Gunakan perintah copy-image. Tentukan ID dari AMI yang akan disalin, Wilayah sumber, dan ARN dari Outposts tujuan.

Misalnya, perintah berikut menyalin snapshot AMI `ami-1234567890abcdef0` dari Wilayah `us-east-1` ke Outpost `arn:aws:outposts:us-east-1:123456789012:outpost/op-1234567890abcdef0`.

```
$ aws ec2 copy-image --source-region us-east-1 --source-image-
id ami-1234567890abcdef0 --name "Local AMI copy"  --destination-outpost-arn
 arn:aws:outposts:us-east-1:123456789012:outpost/op-1234567890abcdef0
```

Membuat volume dari snapshot lokal

Anda dapat membuat volume pada Outposts dari snapshot lokal. Volume harus dibuat pada Outposts yang sama dengan snapshot sumber. Anda tidak dapat menggunakan snapshot lokal untuk membuat volume di Wilayah untuk Outposts.

Ketika Anda membuat volume dari snapshot lokal, Anda tidak dapat mengenkripsi ulang volume menggunakan kunci KMS yang berbeda. Volume yang dibuat dari snapshot lokal harus dienkripsi menggunakan kunci KMS yang sama dengan snapshot sumber.

Untuk informasi selengkapnya, lihat Membuat volume dari snapshot.

Meluncurkan instans dari AMI didukung oleh snapshot lokal

Anda dapat meluncurkan instans dari AMI didukung oleh snapshot lokal Anda harus meluncurkan Instans pada Outposts yang sama dengan AMI sumber. Untuk informasi selengkapnya, lihat Meluncurkan instans pada Outposts di Panduan Pengguna AWS Outposts .

Menghapus snapshot lokal

Anda dapat menghapus snapshot lokal dari Outposts. Setelah Anda menghapus snapshot dari Outposts, kapasitas penyimpanan Amazon S3 yang digunakan oleh snapshot yang dihapus tersedia dalam waktu 72 jam setelah penghapusan snapshot dan volume yang mereferensikan snapshot tersebut.

Karena kapasitas penyimpanan Amazon S3 tidak segera tersedia, kami sarankan Anda menggunakan CloudWatch alarm Amazon untuk memantau kapasitas penyimpanan Amazon S3 Anda. Hapus snapshot dan volume yang tidak Anda perlukan lagi untuk menghindari kehabisan kapasitas penyimpanan.

Untuk informasi selengkapnya tentang menghapus snapshot, lihat Menghapus snapshot.

Mengotomatisasi snapshot di Outposts

Anda dapat membuat kebijakan siklus hidup snapshot Amazon Data Lifecycle Manager yang secara otomatis membuat, menyalin, mempertahankan, dan menghapus snapshot dari volume

dan instans Anda di Outposts. Anda dapat memilih akan menyimpan snapshot di suatu Wilayah atau menyimpannya secara lokal pada Outposts. Selain itu, Anda dapat secara otomatis menyalin snapshot yang dibuat dan disimpan di AWS Wilayah ke Pos Luar.

Tabel berikut memberikan gambaran umum fitur yang didukung.

Lokasi sumber daya	Tujuan snapshot	Penyalinan lintas wilayah		Pemulihan snapshot cepat	Berbagi lintas akun
		Ke Wilayah	Ke Outposts		
Region	Region	✓	✓	✓	✓
Outpost	Region	✓	✓	✓	✓
Outpost	Outpost	✗	✗	✗	✗

Pertimbangan

- Hanya kebijakan siklus hidup snapshot Amazon EBS yang saat ini didukung. EBS-didukung AMI kebijakan dan lintas akun berbagi peristiwa kebijakan tidak didukung.
- Jika kebijakan mengelola snapshot untuk volume atau instans di suatu Wilayah, snapshot dibuat di Wilayah yang sama dengan sumber daya sumber.
- Jika kebijakan mengelola snapshot untuk volume atau instans di Outposts, snapshot dapat dibuat pada Outposts sumber, atau di suatu Wilayah untuk Outposts tersebut.
- Kebijakan tunggal tidak dapat mengelola snapshot di Wilayah dan snapshot di Outposts. Jika Anda perlu untuk mengotomatisasi snapshot di Wilayah dan Outposts, Anda harus membuat kebijakan terpisah.
- Pemulihan snapshot cepat tidak didukung untuk snapshot dibuat pada Outposts, atau untuk snapshot disalin ke Outposts.
- Berbagi lintas akun tidak didukung untuk snapshot yang dibuat di Outposts.

Untuk informasi selengkapnya tentang membuat siklus hidup snapshot yang mengelola snapshot lokal, lihat Mengotomatisasi siklus hidup snapshot.

Enkripsi EBS Amazon

Gunakan enkripsi Amazon EBS sebagai solusi enkripsi langsung untuk sumber daya EBS yang terkait dengan instans EC2 Anda. Gunakan enkripsi Amazon EBS, dan Anda tidak perlu membangun, memelihara, atau mengamankan infrastruktur manajemen kunci Anda sendiri. Enkripsi Amazon EBS menggunakan AWS KMS keys saat membuat volume dan snapshot yang terenkripsi.

Operasi enkripsi terjadi pada server yang meng-host instans EC2, memastikan keamanan keduanya data-at-rest dan data-in-transit antara instance dan penyimpanan EBS terlampirnya.

Anda dapat melampirkan volume terenkripsi maupun tak terenkripsi ke suatu instans secara bersamaan.

Daftar Isi

Cara kerja enkripsi EBS

Anda dapat mengenkripsi volume boot dan data dari instans EC2.

Saat Anda membuat volume EBS terenkripsi dan melampirkannya ke tipe instans yang didukung, tipe data berikut dienkripsi:

- Data diam di dalam volume
- Semua data yang bergerak antara volume dan instans
- Semua snapshot yang dibuat dari volume
- Semua volume yang dibuat dari snapshot tersebut

Amazon EBS mengenkripsi volume Anda dengan kunci data menggunakan enkripsi data AES-256 standar industri. Kunci data dihasilkan oleh AWS KMS dan kemudian dienkripsi AWS KMS dengan

AWS KMS kunci Anda sebelum disimpan dengan informasi volume Anda. Semua snapshot, dan volume berikutnya yang dibuat dari snapshot tersebut menggunakan kunci yang sama berbagi AWS KMS kunci data yang sama. Untuk informasi selengkapnya, lihat Kunci data di Panduan Developer AWS Key Management Service .

Amazon EC2 berfungsi AWS KMS untuk mengenkripsi dan mendekripsi volume EBS Anda dengan cara yang sedikit berbeda tergantung pada apakah snapshot dari mana Anda membuat volume terenkripsi dienkripsi atau tidak dienkripsi.

Cara kerja enkripsi EBS saat snapshot dienkripsi

Saat Anda membuat volume terenkripsi dari snapshot terenkripsi yang Anda miliki, Amazon EC2 berfungsi AWS KMS untuk mengenkripsi dan mendekripsi volume EBS Anda sebagai berikut:

1. Amazon EC2 mengirimkan GenerateDataKeyWithoutPlaintextpermintaan ke AWS KMS, menentukan kunci KMS yang Anda pilih untuk enkripsi volume.
2. Jika volume dienkripsi menggunakan kunci KMS yang sama dengan snapshot, AWS KMS gunakan kunci data yang sama dengan snapshot dan mengenkripsinya di bawah kunci KMS yang sama. Jika volume dienkripsi menggunakan kunci KMS yang berbeda, AWS KMS buat kunci data baru dan enkripsi di bawah kunci KMS yang Anda tentukan. Kunci data terenkripsi dikirimkan ke Amazon EBS untuk disimpan dengan metadata volume.
3. Saat Anda melampirkan volume terenkripsi ke instans, Amazon EC2 mengirimkan CreateGrantpermintaan agar dapat AWS KMS mendekripsi kunci data.
4. AWS KMS mendekripsi kunci data terenkripsi dan mengirimkan kunci data yang didekripsi ke Amazon EC2.
5. Amazon EC2 menggunakan kunci data teks biasa di perangkat keras Nitro untuk mengenkripsi I/O disk ke volume. Kunci data teks biasa tetap ada di memori selama volumenya dilampirkan pada instans.

Cara kerja enkripsi EBS saat snapshot yang tidak terenkripsi

Ketika Anda membuat volume terenkripsi dari snapshot yang tidak terenkripsi yang Anda miliki, Amazon EC2 bekerja dengan AWS KMS untuk mengenkripsi dan mendekripsi volume EBS Anda sebagai berikut:

1. Amazon EC2 mengirimkan CreateGrantpermintaan ke AWS KMS, sehingga dapat mengenkripsi volume yang dibuat dari snapshot.

2. Amazon EC2 mengirimkan GenerateDataKeyWithoutPlaintextpermintaan ke AWS KMS, menentukan kunci KMS yang Anda pilih untuk enkripsi volume.
3. AWS KMS menghasilkan kunci data baru, mengenkripsinya di bawah kunci KMS yang Anda pilih untuk enkripsi volume, dan mengirimkan kunci data terenkripsi ke Amazon EBS untuk disimpan dengan metadata volume.
4. Amazon EC2 mengirimkan permintaan Dekripsi AWS KMS untuk mendekripsi kunci data terenkripsi, yang kemudian digunakan untuk mengenkripsi data volume.
5. Saat Anda melampirkan volume terenkripsi ke instans, Amazon EC2 mengirimkan CreateGrantpermintaan AWS KMS ke, sehingga dapat mendekripsi kunci data.
6. Saat Anda melampirkan volume terenkripsi ke instans, Amazon EC2 mengirimkan permintaan Dekripsi AWS KMS ke, yang menentukan kunci data terenkripsi.
7. AWS KMS mendekripsi kunci data terenkripsi dan mengirimkan kunci data yang didekripsi ke Amazon EC2.
8. Amazon EC2 menggunakan kunci data teks biasa di perangkat keras Nitro untuk mengenkripsi I/O disk ke volume. Kunci data teks biasa tetap ada di memori selama volumenya dilampirkan pada instans.

Untuk informasi selengkapnya, lihat Cara Amazon Elastic Block Store (Amazon EBS) menggunakan AWS KMS dan Contoh dua Amazon EC2 dalam Panduan Developer AWS Key Management Service .

Bagaimana kunci KMS yang tidak dapat digunakan memengaruhi kunci data

Ketika kunci KMS menjadi tidak dapat digunakan, efeknya hampir seketika (tergantung pada konsistensi akhirnya). Status kunci dari perubahan kunci KMS untuk mencerminkan kondisi barunya, dan semua permintaan untuk menggunakan kunci KMS dalam operasi kriptografi gagal.

Saat Anda melakukan tindakan yang membuat kunci KMS tidak dapat digunakan, tidak akan ada efek langsung pada instans EC2 atau volume EBS yang dilampirkan. Amazon EC2 menggunakan kunci data, bukan kunci KMS, untuk mengenkripsi semua I/O disk saat volume dilampirkan ke instans.

Namun, saat volume EBS terenkripsi dicopot dari instans EC2, Amazon EBS menghapus kunci data dari perangkat keras Nitro. Pada saat volume EBS yang terenkripsi dilampirkan ke instans EC2,

pelampiran akan gagal karena Amazon EBS tidak dapat menggunakan kunci KMS untuk mendekripsi kunci data terenkripsi dari volume tersebut. Untuk menggunakan volume EBS lagi, Anda harus membuat kunci KMS dapat digunakan lagi.

> **Tip**
>
> Jika Anda tidak lagi menginginkan akses ke data yang disimpan dalam volume EBS yang dienkripsi dengan kunci data yang dihasilkan dari kunci KMS yang ingin Anda buat agar tidak dapat digunakan, sebaiknya copot volume EBS dari instans EC2 sebelum Anda membuat kunci KMS tidak dapat digunakan.

Untuk informasi selengkapnya, lihat Bagaimana kunci KMS yang tidak dapat digunakan memengaruhi kunci data di Panduan Developer AWS Key Management Service .

Persyaratan untuk enkripsi Amazon EBS

Sebelum memulai, verifikasi bahwa persyaratan berikut dipenuhi.

Persyaratan

- Tipe volume yang mendukung
- Tipe instans yang didukung
- Izin untuk pengguna
- Izin untuk instans

Tipe volume yang mendukung

Enkripsi mendukung oleh semua tipe volume EBS. Anda dapat mengharapkan performa IOPS yang sama pada volume terenkripsi seperti pada volume yang tidak terenkripsi, dengan efek minimal pada latensi. Anda dapat mengakses volume terenkripsi dengan cara yang sama seperti Anda mengakses volume yang tidak terenkripsi. Enkripsi dan dekripsi ditangani secara transparan, dan tidak memerlukan tindakan tambahan dari Anda atau aplikasi Anda.

Tipe instans yang didukung

Enkripsi Amazon EBS tersedia di semua jenis instans generasi saat ini dan generasi sebelumnya.

Izin untuk pengguna

Bila Anda menggunakan kunci KMS untuk enkripsi EBS, kebijakan kunci KMS memungkinkan setiap pengguna dengan akses ke AWS KMS tindakan yang diperlukan untuk menggunakan kunci KMS ini untuk mengenkripsi atau mendekripsi sumber daya EBS. Anda harus memberikan izin kepada pengguna untuk melakukan tindakan berikut agar dapat menggunakan enkripsi EBS:

- `kms:CreateGrant`
- `kms:Decrypt`
- `kms:DescribeKey`
- `kms:GenerateDataKeyWithoutPlainText`
- `kms:ReEncrypt`

> **Tip**
>
> Untuk mengikuti prinsip hak akses paling rendah, jangan biarkan akses penuh ke `kms:CreateGrant`. Sebagai gantinya, gunakan tombol `kms:GrantIsForAWSResource` kondisi untuk memungkinkan pengguna membuat hibah pada kunci KMS hanya ketika hibah dibuat atas nama pengguna oleh AWS layanan, seperti yang ditunjukkan pada contoh berikut.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "kms:CreateGrant",
            "Resource": [
                "arn:aws:kms:us-east-2:123456789012:key/abcd1234-a123-456d-a12b-
a123b4cd56ef"
            ],
            "Condition": {
                "Bool": {
                    "kms:GrantIsForAWSResource": true
                }
            }
        }
```

```
    ]
}
```

Untuk informasi selengkapnya, lihat Mengizinkan akses ke AWS akun dan mengaktifkan kebijakan IAM di bagian Kebijakan kunci default di Panduan AWS Key Management Service Pengembang.

Izin untuk instans

Saat instans mencoba berinteraksi dengan AMI, volume, atau snapshot terenkripsi, pemberian kunci KMS dikeluarkan untuk peran khusus identitas instans. Peran khusus identitas adalah peran IAM yang digunakan oleh instans untuk berinteraksi dengan AMI, volume, atau snapshot yang terenkripsi atas nama Anda.

Peran khusus identitas tidak perlu dibuat atau dihapus secara manual, dan tidak memiliki kebijakan yang terkait dengannya. Selain itu, Anda tidak dapat mengakses kredensial peran khusus identitas.

> **Note**
>
> Peran khusus identitas tidak digunakan oleh aplikasi pada instans Anda untuk mengakses sumber daya AWS KMS terenkripsi lainnya, seperti objek Amazon S3 atau tabel Dynamo DB. Operasi ini dilakukan dengan menggunakan kredensil peran instans Amazon EC2, atau kredenal AWS lain yang telah Anda konfigurasikan pada instans Anda.

Peran khusus identitas tunduk pada kebijakan kontrol layanan (SCP), dan kebijakan kunci KMS. Jika kunci SCP atau KMS menolak akses peran identitas saja ke kunci KMS, Anda mungkin gagal meluncurkan instans EC2 dengan volume terenkripsi, atau menggunakan AMI atau snapshot terenkripsi.

Jika Anda membuat SCP atau kebijakan kunci yang menolak akses berdasarkan lokasi jaringan menggunakan`aws:SourceIp`,,`aws:VpcSourceIp`, atau kunci kondisi `aws:SourceVpce` AWS global`aws:SourceVpc`, maka Anda harus memastikan bahwa pernyataan kebijakan ini tidak berlaku untuk peran khusus instance. Untuk contoh kebijakan, lihat Contoh Kebijakan Perimeter Data.

ARN peran khusus identitas menggunakan format berikut:

```
arn:aws-partition:iam::account_id:role/aws:ec2-infrastructure/instance_id
```

Ketika pemberian kunci diberikan kepada sebuah instans, pemberian kunci tersebut dikeluarkan untuk sesi peran yang diasumsikan khusus untuk instans tersebut. ARN pengguna utama penerima menggunakan format berikut:

```
arn:aws-partition:sts::account_id:assumed-role/aws:ec2-infrastructure/instance_id
```

Bekerja dengan enkripsi Amazon EBS

Gunakan prosedur berikut untuk bekerja dengan enkripsi Amazon EBS.

Tugas

- Pilih kunci KMS untuk enkripsi EBS
- Aktifkan enkripsi secara default
- Kelola enkripsi secara default menggunakan API dan CLI

Pilih kunci KMS untuk enkripsi EBS

Amazon EBS secara otomatis membuat unik Kunci yang dikelola AWS di setiap Wilayah tempat Anda menyimpan AWS sumber daya. Kunci KMS ini memiliki alias `alias/aws/ebs`. Secara default, Amazon EBS menggunakan kunci KMS ini untuk enkripsi. Alternatifnya, Anda dapat menentukan kunci enkripsi yang dikelola pelanggan simetris yang Anda buat sebagai kunci KMS default untuk enkripsi EBS. Penggunaan kunci KMS sendiri akan memberikan Anda fleksibilitas yang lebih baik, termasuk kemampuan untuk membuat, memutar, dan menonaktifkan kunci KMS.

Important

Amazon EBS tidak mendukung kunci KMS enkripsi asimetris. Untuk informasi selengkapnya, lihat Menggunakan kunci KMS enkripsi simetris dan asimetris di Panduan Developer AWS Key Management Service .

Amazon EC2 console

Untuk mengonfigurasi kunci KMS default guna enkripsi EBS untuk suatu Wilayah

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Pada bilah navigasi, pilih Wilayah.

3. Dari panel navigasi, pilih Dasbor EC2.
4. Di sudut kanan atas halaman, pilih Atribut Akun, Perlindungan dan keamanan data.
5. Pilih Kelola.
6. Untuk Kunci enkripsi default, pilih kunci enkripsi yang dikelola pelanggan simetris.
7. Pilih Perbarui enkripsi EBS.

Aktifkan enkripsi secara default

Anda dapat mengonfigurasi AWS akun Anda untuk menerapkan enkripsi volume EBS baru dan salinan snapshot yang Anda buat. Misalnya, Amazon EBS mengenkripsi volume EBS yang dibuat saat Anda meluncurkan instans dan snapshot yang Anda salin dari snapshot yang tidak dienkripsi. Untuk contoh transisi dari sumber daya EBS tidak terenkripsi menjadi terenkripsi, lihat Mengenkripsi sumber daya yang tidak terenkripsi.

Enkripsi secara default tidak berpengaruh pada volume atau snapshot EBS yang ada.

Pertimbangan

- Enkripsi secara default adalah pengaturan khusus Wilayah. Jika Anda aktifkan untuk sebuah Wilayah, Anda tidak dapat menonaktifkannya untuk volume atau snapshot individual di Wilayah tersebut.
- Enkripsi Amazon EBS secara default didukung pada semua jenis instans generasi saat ini dan generasi sebelumnya.
- Jika Anda menyalin snapshot dan mengenkripsinya ke kunci KMS baru, salinan lengkap (tidak inkremental) dibuat. Hal ini menyebabkan biaya penyimpanan tambahan.
- Saat memigrasi server menggunakan AWS Server Migration Service (SMS), jangan nyalakan enkripsi secara default. Jika enkripsi secara default sudah aktif dan Anda mengalami kegagalan replikasi delta, matikan enkripsi secara default. Sebaliknya, aktifkan enkripsi AMI saat Anda membuat tugas replikasi.

Amazon EC2 console

Untuk mengaktifkan enkripsi secara default untuk Wilayah

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Pada bilah navigasi, pilih Wilayah.

3. Dari panel navigasi, pilih Dasbor EC2.
4. Di sudut kanan atas halaman, pilih Atribut Akun, Perlindungan dan keamanan data.
5. Pilih Kelola.
6. Pilih Aktifkan. Anda menyimpan Kunci yang dikelola AWS dengan alias yang `alias/aws/ebs` dibuat atas nama Anda sebagai kunci enkripsi default, atau memilih kunci enkripsi terkelola pelanggan simetris.
7. Pilih Perbarui enkripsi EBS.

AWS CLI

Untuk melihat pengaturan enkripsi secara default

- Untuk Wilayah tertentu

```
$ aws ec2 get-ebs-encryption-by-default --region region
```

- Untuk semua Wilayah di akun Anda

```
$ for region in $(aws ec2 describe-regions --region us-east-1 --query "Regions[*].
[RegionName]" --output text); do   default=$(aws ec2 get-ebs-encryption-by-default
 --region $region --query "{Encryption_By_Default:EbsEncryptionByDefault}" --
output text); kms_key=$(aws ec2 get-ebs-default-kms-key-id --region $region | jq
 '.KmsKeyId'); echo "$region  --- $default  --- $kms_key"; done
```

Untuk mengaktifkan enkripsi secara default

- Untuk Wilayah tertentu

```
$ aws ec2 enable-ebs-encryption-by-default --region region
```

- Untuk semua Wilayah di akun Anda

```
$ for region in $(aws ec2 describe-regions --region us-east-1 --query "Regions[*].
[RegionName]" --output text); do   default=$(aws ec2 enable-ebs-encryption-by-
default --region $region --query "{Encryption_By_Default:EbsEncryptionByDefault}"
 --output text); kms_key=$(aws ec2 get-ebs-default-kms-key-id --region $region |
 jq '.KmsKeyId'); echo "$region  --- $default  --- $kms_key"; done
```

Untuk menonaktifkan enkripsi secara default

- Untuk Wilayah tertentu

```
$ aws ec2 disable-ebs-encryption-by-default --region region
```

- Untuk semua Wilayah di akun Anda

```
$ for region in $(aws ec2 describe-regions --region us-east-1 --query "Regions[*].
[RegionName]" --output text); do   default=$(aws ec2 disable-ebs-encryption-by-
default --region $region --query "{Encryption_By_Default:EbsEncryptionByDefault}"
 --output text); kms_key=$(aws ec2 get-ebs-default-kms-key-id --region $region |
 jq '.KmsKeyId'); echo "$region  --- $default  --- $kms_key"; done
```

PowerShell

Untuk melihat pengaturan enkripsi secara default

- Untuk Wilayah tertentu

```
PS C:\> Get-EC2EbsEncryptionByDefault -Region region
```

- Untuk semua Wilayah di akun Anda

```
PS C:\> (Get-EC2Region).RegionName | ForEach-Object { [PSCustomObject]@{ Region
 = $_; EC2EbsEncryptionByDefault = Get-EC2EbsEncryptionByDefault -Region $_;
 EC2EbsDefaultKmsKeyId = Get-EC2EbsDefaultKmsKeyId -Region $_ } } | Format-Table -
AutoSize
```

Untuk mengaktifkan enkripsi secara default

- Untuk Wilayah tertentu

```
PS C:\> Enable-EC2EbsEncryptionByDefault -Region region
```

- Untuk semua Wilayah di akun Anda

```
PS C:\> (Get-EC2Region).RegionName | ForEach-Object { [PSCustomObject]@{ Region
 = $_; EC2EbsEncryptionByDefault = Enable-EC2EbsEncryptionByDefault -Region $_;
```

```
 EC2EbsDefaultKmsKeyId = Get-EC2EbsDefaultKmsKeyId -Region $_ } } | Format-Table -
AutoSize
```

Untuk menonaktifkan enkripsi secara default

- Untuk Wilayah tertentu

```
PS C:\> Disable-EC2EbsEncryptionByDefault -Region region
```

- Untuk semua Wilayah di akun Anda

```
PS C:\> (Get-EC2Region).RegionName | ForEach-Object { [PSCustomObject]@{ Region
 = $_; EC2EbsEncryptionByDefault = Disable-EC2EbsEncryptionByDefault -Region $_;
 EC2EbsDefaultKmsKeyId = Get-EC2EbsDefaultKmsKeyId -Region $_ } } | Format-Table -
AutoSize
```

Anda tidak dapat mengubah kunci KMS yang terkait dengan snapshot yang ada atau volume terenkripsi. Namun, Anda dapat mengaitkan kunci KMS yang berbeda selama operasi salinan snapshot sehingga snapshot salinan yang dihasilkan dienkripsi oleh kunci KMS yang baru.

Kelola enkripsi secara default menggunakan API dan CLI

Anda dapat mengelola enkripsi secara default dan kunci KMS default menggunakan tindakan API dan perintah CLI berikut.

Tindakan API	Perintah CLI	Deskripsi
DisableEbsEncryptionByDefault	disable-ebs-encryption-by-default	Menonaktifkan enkripsi secara default.
EnableEbsEncryptionByDefault	enable-ebs-encryption-by-default	Menonaktifkan enkripsi secara default.
GetEbsDefaultKmsKeyId	get-ebs-default-kms-kunci-id	Menjelaskan kunci KMS default.

Tindakan API	Perintah CLI	Deskripsi
GetEbsEncryptionByDefault	get-ebs-encryption-by-default	Menunjukkan apakah enkripsi secara default diaktifkan.
ModifyEbsDefaultKmsKeyId	modify-ebs-default-kms-kunci-id	Mengubah kunci KMS default yang digunakan untuk mengenkripsi volume EBS.
ResetEbsDefaultKmsKeyId	reset-ebs-default-kms-kunci-id	Mengatur ulang Kunci yang dikelola AWS sebagai kunci KMS default yang digunakan untuk mengenkripsi volume EBS.

Enkripsi sumber daya EBS

Anda mengenkripsi volume EBS dengan mengaktifkan enkripsi, menggunakan enkripsi secara default atau dengan mengaktifkan enkripsi saat Anda membuat volume yang ingin Anda enkripsi.

Saat Anda mengenkripsi volume, Anda dapat menentukan kunci KMS enkripsi simetris untuk mengenkripsi volume. Jika Anda tidak menentukan kunci KMS, kunci KMS yang digunakan untuk enkripsi tergantung pada kondisi enkripsi snapshot sumber dan kepemilikannya. Untuk informasi selengkapnya, lihat tabel hasil enkripsi.

> **Note**
>
> Jika Anda menggunakan API atau AWS CLI untuk menentukan kunci KMS, ketahuilah bahwa AWS mengautentikasi kunci KMS secara asinkron. Jika Anda menentukan ID kunci KMS, suatu alias, atau ARN yang tidak valid, tindakan dapat muncul untuk diselesaikan, tetapi akhirnya akan gagal.

Anda tidak dapat mengubah kunci KMS yang terkait dengan snapshot atau volume yang ada. Namun, Anda dapat mengaitkan kunci KMS yang berbeda selama operasi salinan snapshot sehingga snapshot salinan yang dihasilkan dienkripsi oleh kunci KMS yang baru.

Enkripsi volume kosong pada saat pembuatan

Saat Anda membuat volume EBS baru yang kosong, Anda dapat mengenkripsinya dengan mengaktifkan enkripsi untuk operasi pembuatan volume tertentu. Jika Anda mengaktifkan enkripsi EBS secara default, volume akan dienkripsi secara otomatis menggunakan kunci KMS default untuk enkripsi EBS. Sebagai alternatif, Anda dapat menentukan kunci KMS enkripsi simetris yang berbeda untuk operasi pembuatan volume spesifik. Volume dienkripsi saat pertama kali tersedia, sehingga data Anda selalu aman. Untuk prosedur terperinci, lihat Buat volume Amazon EBS.

Secara default, kunci KMS yang Anda pilih saat membuat volume mengenkripsi snapshot yang Anda buat dari volume dan volume yang Anda pulihkan dari snapshot yang dienkripsi tersebut. Anda tidak dapat menghapus enkripsi dari volume atau snapshot terenkripsi, yang berarti bahwa volume yang dipulihkan dari snapshot terenkripsi, atau salinan snapshot terenkripsi, selalu dienkripsi.

Snapshot publik dari volume terenkripsi tidak didukung, tetapi Anda dapat berbagi snapshot terenkripsi dengan akun tertentu. Untuk petunjuk terperinci, lihat Membagikan snapshot Amazon EBS.

Mengenkripsi sumber daya yang tidak terenkripsi

Anda tidak dapat secara langsung mengenkripsi volume atau snapshot yang tidak terenkripsi. Namun, Anda dapat membuat volume atau snapshot terenkripsi dari volume atau snapshot yang tidak terenkripsi. Jika Anda mengaktifkan enkripsi secara default, Amazon EBS secara otomatis mengenkripsi volume dan snapshot baru menggunakan kunci KMS default Anda untuk enkripsi EBS. Jika tidak, Anda dapat mengaktifkan enkripsi saat membuat volume atau snapshot individual, menggunakan kunci KMS default untuk enkripsi Amazon EBS atau kunci enkripsi simetris yang dikelola pelanggan. Untuk informasi lebih lanjut, lihat Buat volume Amazon EBS dan Menyalin snapshot Amazon EBS.

Untuk mengenkripsi salinan snapshot ke kunci yang dikelola pelanggan, Anda harus mengaktifkan enkripsi dan menentukan kunci KMS, seperti yang ditunjukkan dalam Menyalin snapshot yang tidak terenkripsi (enkripsi secara default tidak diaktifkan).

> **Important**
>
> Amazon EBS tidak mendukung kunci KMS enkripsi asimetris. Untuk informasi selengkapnya, lihat Menggunakan kunci KMS enkripsi Simetris dan Asimetris di Panduan Developer AWS Key Management Service .

Anda juga dapat menerapkan status enkripsi baru saat meluncurkan instans dari AMI yang didukung EBS. Hal ini karena AMI yang didukung EBS menyertakan snapshot volume EBS yang dapat dienkripsi sebagaimana dijelaskan. Untuk informasi selengkapnya, lihat Menggunakan enkripsi dengan AMI yang didukung EBS.

Tombol berputar AWS KMS

Praktik terbaik kriptografi mencegah penggunaan ulang kunci enkripsi secara ekstensif.

Untuk membuat materi kriptografi baru untuk digunakan dengan enkripsi Amazon EBS, Anda dapat membuat kunci terkelola pelanggan baru, dan kemudian mengubah aplikasi Anda untuk menggunakan kunci KMS baru itu. Atau, Anda dapat mengaktifkan rotasi kunci otomatis untuk kunci terkelola pelanggan yang ada.

Saat Anda mengaktifkan rotasi kunci otomatis untuk kunci yang dikelola pelanggan, AWS KMS hasilkan materi kriptografi baru untuk kunci KMS setiap tahun. AWS KMS menyimpan semua versi sebelumnya dari materi kriptografi sehingga Anda dapat terus mendekripsi dan menggunakan volume dan snapshot yang sebelumnya dienkripsi dengan materi kunci KMS tersebut. AWS KMS tidak menghapus materi kunci yang diputar sampai Anda menghapus kunci KMS.

Saat Anda menggunakan kunci terkelola pelanggan yang diputar untuk mengenkripsi volume atau snapshot baru, AWS KMS gunakan materi kunci (baru) saat ini. Saat Anda menggunakan kunci terkelola pelanggan yang diputar untuk mendekripsi volume atau snapshot, AWS KMS gunakan versi materi kriptografi yang digunakan untuk mengenkripsi itu. Jika volume atau snapshot dienkripsi dengan versi sebelumnya dari materi kriptografi, AWS KMS terus gunakan versi sebelumnya untuk mendekripsi itu. AWS KMS tidak mengenkripsi ulang volume atau snapshot yang sebelumnya dienkripsi untuk menggunakan materi kriptografi baru setelah rotasi kunci. Mereka tetap dienkripsi dengan bahan kriptografi yang awalnya dienkripsi. Anda dapat dengan aman menggunakan kunci terkelola pelanggan yang diputar dalam aplikasi dan AWS layanan tanpa perubahan kode.

> **Note**
>
> - Rotasi kunci otomatis hanya didukung untuk kunci yang dikelola pelanggan simetris dengan materi utama yang AWS KMS dibuat.
> - AWS KMS secara otomatis berputar Kunci yang dikelola AWS setiap tahun. Anda tidak dapat mengaktifkan atau menonaktifkan rotasi kunci untuk Kunci yang dikelola AWS.

Untuk informasi selengkapnya, lihat Merotasi kunci KMS di Panduan Developer AWS Key Management Service .

Contoh enkripsi Amazon EBS

Saat Anda membuat sumber daya EBS terenkripsi, sumber daya tersebut dienkripsi dengan kunci KMS default akun Anda untuk enkripsi EBS kecuali Anda menentukan kunci yang dikelola pelanggan yang berbeda dalam parameter pembuatan volume atau pemetaan perangkat blok untuk AMI atau instans. Untuk informasi selengkapnya, lihat Pilih kunci KMS untuk enkripsi EBS.

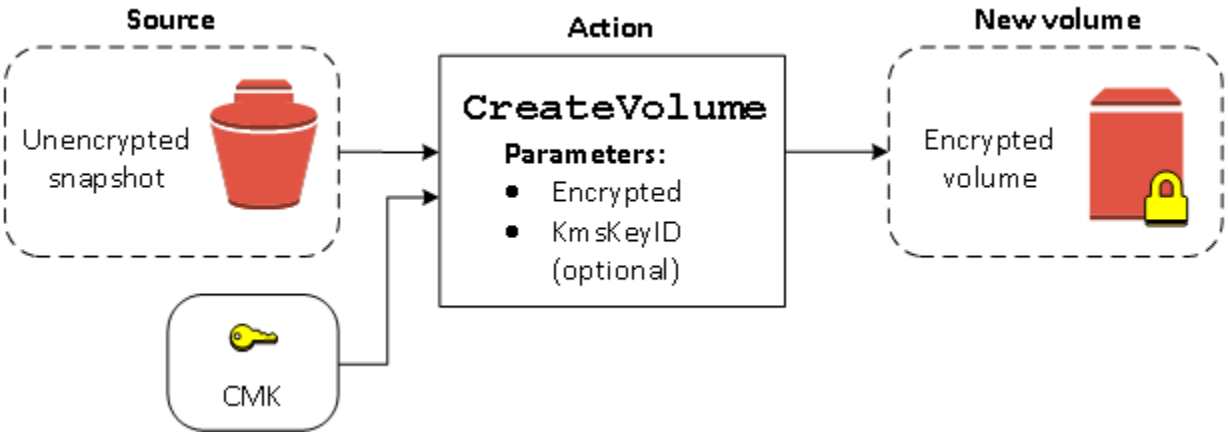
Contoh berikut ini menggambarkan cara mengelola status enkripsi volume dan snapshot Anda. Untuk daftar lengkap kasus enkripsi, lihat tabel hasil enkripsi.

Contoh

- Mengembalikan volume yang tidak terenkripsi (enkripsi secara default tidak diaktifkan)
- Mengembalikan volume yang tidak terenkripsi (enkripsi secara default diaktifkan)
- Menyalin snapshot yang tidak terenkripsi (enkripsi secara default tidak diaktifkan)
- Menyalin snapshot yang tidak terenkripsi (enkripsi secara default tidak diaktifkan)
- Mengenkripsi ulang volume yang dienkripsi
- Mengenkripsi ulang snapshot yang dienkripsi
- Memigrasikan data antara volume terenkripsi dan tidak terenkripsi
- Hasil enkripsi

Mengembalikan volume yang tidak terenkripsi (enkripsi secara default tidak diaktifkan)

Tanpa enkripsi yang diaktifkan secara default, volume yang dipulihkan dari snapshot yang tidak dienkripsi tidak akan dienkripsi secara default. Namun, Anda dapat mengenkripsi volume yang dihasilkan dengan mengatur `Encrypted` dan, secara opsional, `KmsKeyId` parameter. Diagram berikut menggambarkan prosesnya.

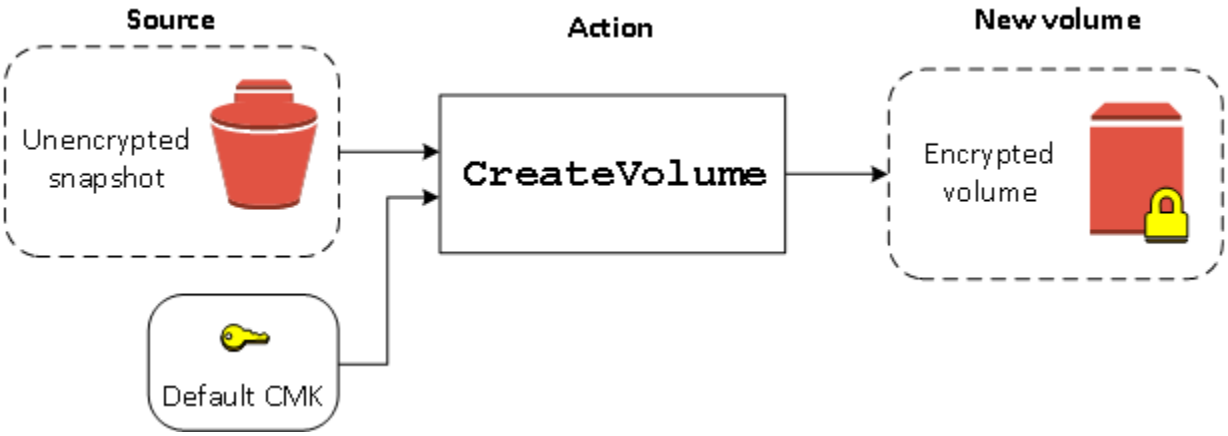


Jika Anda meninggalkan parameter `KmsKeyId`, volume yang dihasilkan dienkripsi menggunakan kunci KMS default Anda untuk enkripsi EBS. Anda harus menentukan ID kunci KMS untuk mengenkripsi volume ke kunci KMS yang berbeda.

Untuk informasi selengkapnya, lihat Membuat volume dari snapshot.

Mengembalikan volume yang tidak terenkripsi (enkripsi secara default diaktifkan)

Jika Anda telah mengaktifkan enkripsi secara default, enkripsi wajib dilakukan untuk volume yang dipulihkan dari snapshot yang tidak terenkripsi, dan tidak ada parameter enkripsi yang diperlukan agar kunci KMS default Anda dapat digunakan. Diagram berikut menunjukkan kasus default sederhana ini:

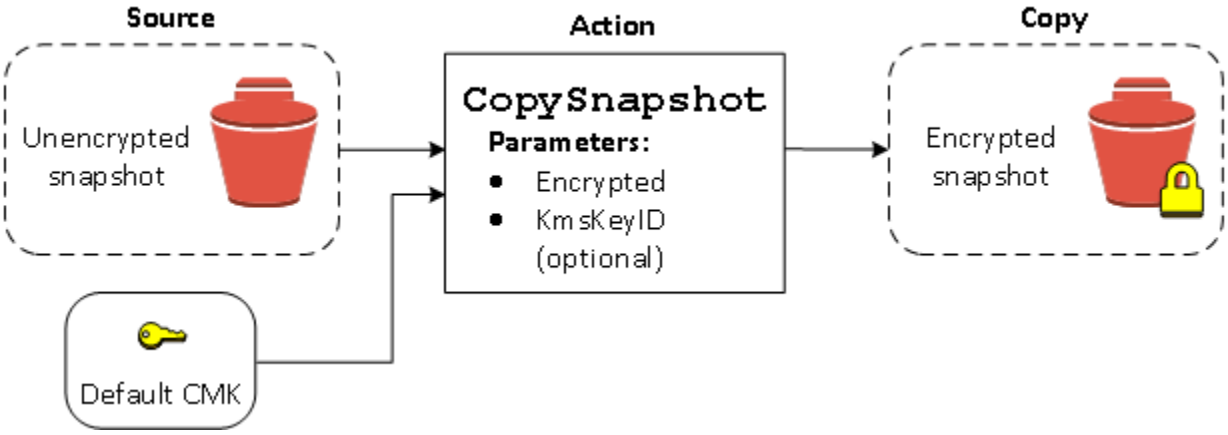


Jika Anda ingin mengenkripsi volume yang dipulihkan ke kunci enkripsi yang dikelola pelanggan simetris, Anda harus menyediakan `Encrypted` dan `KmsKeyId` parameter seperti ditunjukkan dalam Mengembalikan volume yang tidak terenkripsi (enkripsi secara default tidak diaktifkan).

Menyalin snapshot yang tidak terenkripsi (enkripsi secara default tidak diaktifkan)

Tanpa enkripsi yang diaktifkan secara default, salinan snapshot yang tidak dienkripsi tidak akan dienkripsi secara default. Namun, Anda dapat mengenkripsi snapshot yang dihasilkan dengan mengatur parameter `Encrypted` dan, secara opsional, parameter `KmsKeyId`. Jika Anda menghilangkan `KmsKeyId`, snapshot yang dihasilkan dienkripsi oleh kunci KMS default Anda. Anda harus menentukan ID kunci KMS untuk mengenkripsi volume ke kunci KMS enkripsi simetris yang berbeda.

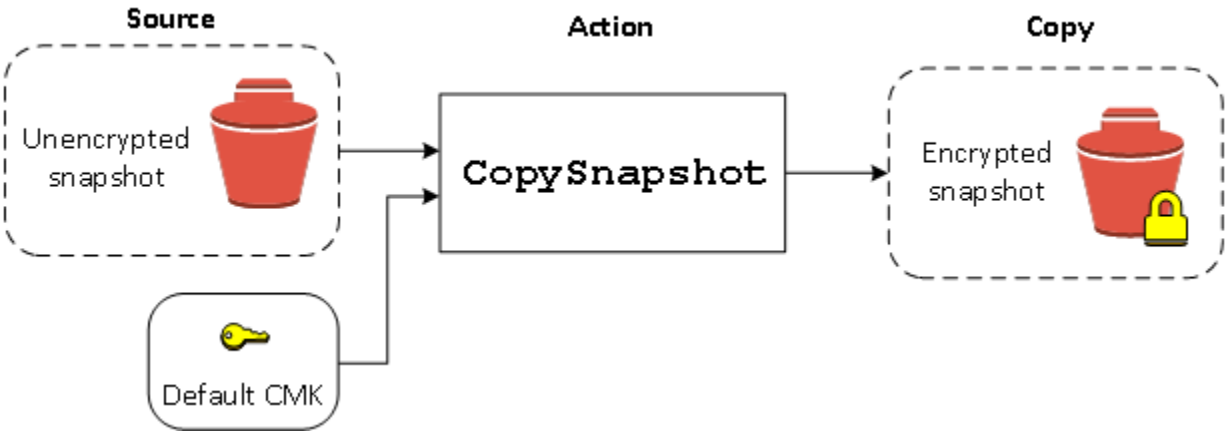
Diagram berikut menggambarkan prosesnya.



Anda dapat mengenkripsi volume EBS dengan menyalin snapshot yang tidak dienkripsi ke snapshot yang dienkripsi, lalu membuat volume dari snapshot yang dienkripsi. Untuk informasi selengkapnya, lihat Menyalin snapshot Amazon EBS.

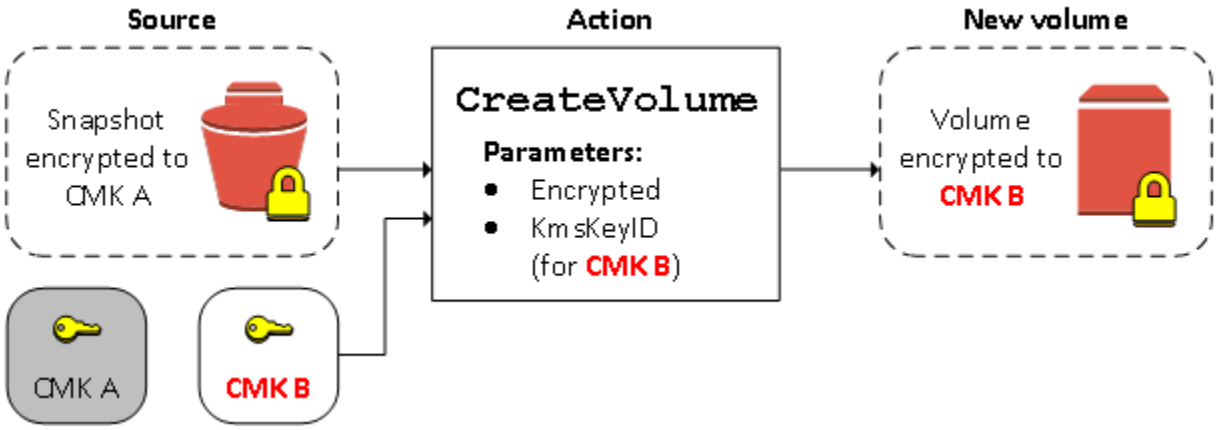
Menyalin snapshot yang tidak terenkripsi (enkripsi secara default tidak diaktifkan)

Ketika Anda telah mengaktifkan enkripsi secara default, enkripsi diwajibkan untuk salinan snapshot yang tidak dienkripsi, dan tidak ada parameter enkripsi yang diperlukan jika kunci KMS default Anda digunakan. Diagram berikut menggambarkan kasus default ini:



Mengenkripsi ulang volume yang dienkripsi

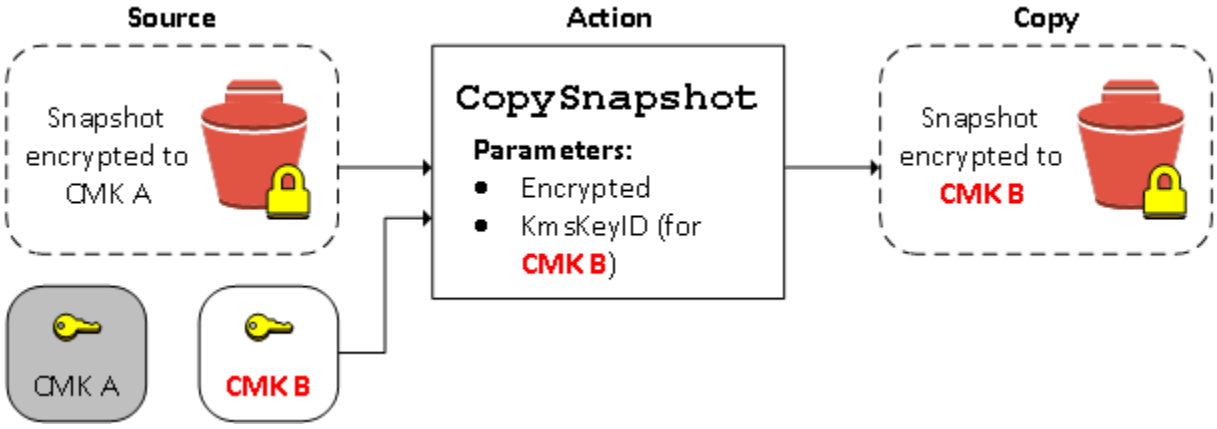
Saat tindakan `CreateVolume` beroperasi pada snapshot terenkripsi, Anda memiliki opsi mengenkripsi ulang kunci KMS yang berbeda. Diagram berikut menggambarkan prosesnya. Dalam contoh ini, Anda memiliki dua kunci KMS, kunci KMS A dan kunci KMS B. Snapshot sumber dienkripsi oleh kunci KMS A. Selama pembuatan volume, dengan ID kunci KMS dari kunci KMS B ditentukan sebagai parameter, data sumber adalah didekripsi secara otomatis, kemudian dienkripsi ulang dengan kunci KMS B.



Untuk informasi selengkapnya, lihat Membuat volume dari snapshot.

Mengenkripsi ulang snapshot yang dienkripsi

Kemampuan untuk mengenkripsi snapshot selama penyalinan memungkinkan Anda menerapkan kunci KMS enkripsi simetris baru ke snapshot yang sudah terenkripsi yang Anda miliki. Volume yang dipulihkan dari salinan hasil hanya dapat diakses menggunakan kunci KMS baru. Diagram berikut menggambarkan prosesnya. Dalam contoh ini, Anda memiliki dua kunci KMS, kunci KMS A dan kunci KMS B. Snapshot sumber dienkripsi oleh kunci KMS A. Selama penyalinan, dengan ID kunci KMS dari kunci KMS B ditentukan sebagai parameter, data sumber secara otomatis dienkripsi ulang dengan kunci KMS B.



Dalam skenario terkait, Anda dapat memilih untuk menerapkan parameter enkripsi baru ke salinan snapshot yang telah dibagikan dengan Anda. Secara default, salinan tersebut dienkripsi dengan kunci KMS yang dibagikan oleh pemilik snapshot. Namun, sebaiknya buat salinan snapshot yang dibagikan menggunakan kunci KMS lain yang Anda kontrol. Hal ini melindungi akses Anda ke volume jika kunci KMS awal terancam, atau jika pemilik mencabut kunci KMS karena alasan apa pun. Untuk informasi selengkapnya, lihat Enkripsi dan penyalinan snapshot.

Memigrasikan data antara volume terenkripsi dan tidak terenkripsi

Saat Anda memiliki akses ke volume terenkripsi dan tidak terenkripsi, Anda dapat dengan bebas mentransfer data di antara keduanya. EC2 menjalankan operasi enkripsi dan dekripsi secara transparan.

Instans Linux

Misalnya, gunakan perintah rsync untuk menyalin data. Dalam perintah berikut, data sumber terletak di `/mnt/source` dan volume tujuan dipasang pada `/mnt/destination`.

```
[ec2-user ~]$ sudo rsync -avh --progress /mnt/source/ /mnt/destination/
```

Instans Windows

Misalnya, gunakan perintah robocopy untuk menyalin data. Dalam perintah berikut, data sumber terletak di `D:\` dan volume tujuan dipasang pada `E:\`.

```
PS C:\> robocopy D:\sourcefolder E:\destinationfolder /e /copyall /eta
```

Kami menyarankan untuk menyalin dari folder, daripada seluruh volume, untuk menghindari potensi masalah dari folder tersembunyi.

Hasil enkripsi

Tabel berikut menjelaskan hasil enkripsi untuk setiap kombinasi pengaturan yang memungkinkan.

Apakah enkripsi diaktifkan?	Apakah enkripsi secara default diaktifkan?	Sumber volume	Default (tidak ada kunci dikelola pelanggan yang ditentukan)	Kustom (kunci dikelola pelanggan ditentukan)
Tidak	Tidak	Volume (kosong) baru	Tidak terenkripsi	T/A
Tidak	Tidak	Snapshot tidak terenkripsi yang Anda miliki	Tidak terenkripsi	

Apakah enkripsi diaktifkan?	Apakah enkripsi secara default diaktifkan?	Sumber volume	Default (tidak ada kunci dikelola pelanggan yang ditentukan)	Kustom (kunci dikelola pelanggan ditentukan)
Tidak	Tidak	Snapshot terenkripsi yang Anda miliki	Dienkripsi dengan kunci yang sama	
Tidak	Tidak	Snapshot yang tidak terenkripsi yang dibagikan dengan Anda	Tidak terenkripsi	
Tidak	Tidak	Snapshot terenkripsi yang dibagikan dengan Anda	Dienkripsi secara default dengan kunci yang dikelola pelanggan*	
Ya	Tidak	Volume baru	Dienkripsi secara default dengan kunci yang dikelola pelanggan	Dienkripsi oleh kunci yang dikelola pelanggan yang ditentukan**
Ya	Tidak	Snapshot tidak terenkripsi yang Anda miliki	Dienkripsi secara default dengan kunci yang dikelola pelanggan	
Ya	Tidak	Snapshot terenkripsi yang Anda miliki	Dienkripsi dengan kunci yang sama	

Apakah enkripsi diaktifkan?	Apakah enkripsi secara default diaktifkan?	Sumber volume	Default (tidak ada kunci dikelola pelanggan yang ditentukan)	Kustom (kunci dikelola pelanggan ditentukan)
Ya	Tidak	Snapshot yang tidak terenkripsi yang dibagikan dengan Anda	Dienkripsi secara default dengan kunci yang dikelola pelanggan	
Ya	Tidak	Snapshot terenkripsi yang dibagikan dengan Anda	Dienkripsi secara default dengan kunci yang dikelola pelanggan	
Tidak	Ya	Volume (kosong) baru	Dienkripsi secara default dengan kunci yang dikelola pelanggan	T/A
Tidak	Ya	Snapshot tidak terenkripsi yang Anda miliki	Dienkripsi secara default dengan kunci yang dikelola pelanggan	
Tidak	Ya	Snapshot terenkripsi yang Anda miliki	Dienkripsi dengan kunci yang sama	
Tidak	Ya	Snapshot yang tidak terenkripsi yang dibagikan dengan Anda	Dienkripsi secara default dengan kunci yang dikelola pelanggan	

Apakah enkripsi diaktifkan?	Apakah enkripsi secara default diaktifkan?	Sumber volume	Default (tidak ada kunci dikelola pelanggan yang ditentukan)	Kustom (kunci dikelola pelanggan ditentukan)
Tidak	Ya	Snapshot terenkripsi yang dibagikan dengan Anda	Dienkripsi secara default dengan kunci yang dikelola pelanggan	
Ya	Ya	Volume baru	Dienkripsi secara default dengan kunci yang dikelola pelanggan	Dienkripsi oleh kunci yang dikelola pelanggan yang ditentukan
Ya	Ya	Snapshot tidak terenkripsi yang Anda miliki	Dienkripsi secara default dengan kunci yang dikelola pelanggan	
Ya	Ya	Snapshot terenkripsi yang Anda miliki	Dienkripsi dengan kunci yang sama	
Ya	Ya	Snapshot yang tidak terenkripsi yang dibagikan dengan Anda	Dienkripsi secara default dengan kunci yang dikelola pelanggan	
Ya	Ya	Snapshot terenkripsi yang dibagikan dengan Anda	Dienkripsi secara default dengan kunci yang dikelola pelanggan	

* Ini adalah kunci terkelola pelanggan default yang digunakan untuk enkripsi EBS untuk AWS akun dan Wilayah. Secara default ini adalah unik Kunci yang dikelola AWS untuk EBS, atau Anda dapat menentukan kunci yang dikelola pelanggan. Untuk informasi selengkapnya, lihat Pilih kunci KMS untuk enkripsi EBS.

** Ini adalah kunci yang dikelola pelanggan yang ditentukan untuk volume saat waktu peluncuran. Kunci yang dikelola pelanggan ini digunakan sebagai pengganti kunci terkelola pelanggan default untuk AWS akun dan Wilayah.

Performa volume Amazon EBS

Beberapa faktor, termasuk karakteristik I/O dan konfigurasi instans dan volume Anda, dapat memengaruhi performa Amazon EBS. Jika Anda mengikuti panduan di halaman detail produk Amazon EBS dan Amazon EC2 kami, Anda biasanya akan mencapai kinerja yang baik. Namun, ada beberapa kasus di mana Anda mungkin perlu melakukan beberapa penyetelan untuk mencapai kinerja puncak. Kami merekomendasikan Anda untuk menyesuaikan performa dengan informasi dari beban kerja Anda yang sebenarnya, selain tolok ukur, untuk menentukan konfigurasi optimal Anda. Setelah Anda mempelajari dasar menggunakan volume EBS, ada baiknya untuk melihat performa I/O yang Anda perlukan dan pilihan Anda untuk meningkatkan performa Amazon EBS agar dapat memenuhi persyaratan tersebut.

AWS pembaruan kinerja tipe volume EBS mungkin tidak langsung berpengaruh pada volume Anda yang ada. Untuk melihat performa penuh pada volume yang lebih lama, Anda mungkin harus melakukan tindakan `ModifyVolume` terlebih dahulu. Untuk informasi selengkapnya, lihat Ubah volume menggunakan Amazon EBS Elastic Volumes.

Daftar Isi

Kiat performa Amazon EBS

Kiat ini menunjukkan praktik terbaik untuk mendapatkan performa optimal dari volume EBS Anda dalam berbagai skenario pengguna.

Gunakan instans yang dioptimalkan EBS

Pada instans tanpa dukungan untuk throughput yang dioptimalkan EBS, lalu lintas jaringan dapat bersaing dengan lalu lintas di antara instans dan volume EBS; pada instans yang dioptimalkan EBS, dua jenis lalu lintas itu akan dipisahkan. Beberapa konfigurasi instans yang dioptimalkan

EBS memerlukan biaya tambahan (seperti C3, R3, dan M3), sementara instans lain yang selalu dioptimalkan EBS tidak memerlukan biaya tambahan (seperti M4, C4, C5, dan D2). Untuk informasi selengkapnya, lihat Optimalkan kinerja Amazon EBS.

Memahami cara menghitung performa

Saat Anda mengukur performa volume EBS, penting untuk memahami unit pengukuran yang terlibat dan cara performa dihitung. Untuk informasi selengkapnya, lihat Karakteristik dan pemantauan Amazon EBS I/O.

Memahami beban kerja Anda

Ada hubungan antara performa maksimal volume EBS, ukuran dan jumlah operasi I/O, dan waktu yang diperlukan untuk menyelesaikan setiap tindakan. Masing-masing faktor ini (performa, I/O, dan latensi) memengaruhi yang lain, dan aplikasi yang berbeda bersifat lebih sensitif terhadap satu faktor atau yang lain. Untuk informasi selengkapnya, lihat Tolok ukur volume EBS.

Waspadai penalti performa saat menginisialisasi volume dari snapshot

Terdapat peningkatan latensi yang signifikan saat Anda pertama kali mengakses setiap blok data pada volume EBS baru yang dibuat dari snapshot. Anda dapat menghindari lonjakan performa ini menggunakan salah satu opsi berikut:

- Akses setiap blok sebelum memasukkan volume ke dalam produksi. Proses ini disebut menginisialisasi (sebelumnya dikenal sebagai pra-pemanasan). Untuk informasi selengkapnya, lihat Inisialisasi volume Amazon EBS.
- Mengaktifkan pemulihan snapshot cepat pada snapshot untuk memastikan bahwa volume EBS yang dibuat sepenuhnya diinisialisasi pada saat pembuatan dan secara instan menyampaikan semua performa yang diberikan. Untuk informasi selengkapnya, lihat Pemulihan snapshot cepat Amazon EBS.

Faktor yang dapat menurunkan performa HDD

Saat Anda membuat snapshot dari volume HDD Throughput Dioptimalkan (`st1`) atau Cold HDD (`sc1`), performa dapat menurun sejauh nilai acuan volume saat snapshot sedang berlangsung. Perilaku ini khusus untuk tipe volume ini. Faktor lain yang dapat membatasi performa termasuk mendorong lebih banyak throughput daripada yang dapat didukung oleh instans, penalti performa

yang ditemui saat menginisialisasi volume yang dibuat dari snapshot, dan jumlah I/O kecil acak yang berlebihan pada volume. Untuk informasi selengkapnya tentang penghitungan throughput untuk volume HDD, lihat Tipe volume Amazon EBS.

Performa Anda juga dapat terpengaruh jika aplikasi Anda tidak mengirim cukup permintaan I/O. Hal ini dapat dipantau dengan melihat panjang antrean volume dan ukuran I/O. Panjang antrean adalah jumlah permintaan I/O tertunda dari aplikasi Anda ke volume Anda. Untuk konsistensi maksimum, volume yang didukung HDD harus mempertahankan panjang antrean (dibulatkan ke angka bulat terdekat) sebesar 4 atau lebih ketika melakukan 1 MiB I/O berurutan. Untuk informasi selengkapnya tentang memastikan performa yang konsisten dari volume Anda, lihat Karakteristik dan pemantauan Amazon EBS I/O

Tingkatkan read-ahead untuk throughput tinggi, beban kerja read-heavy pada dan (hanya instance Linux) *st1 sc1*

Beberapa beban kerja adalah read-heavy dan mengakses perangkat blok melalui cache halaman sistem operasi (misalnya, dari sistem file). Dalam hal ini, untuk mencapai throughput maksimal, kami sarankan Anda mengonfigurasi pengaturan read-ahead menjadi 1 MiB. Ini adalah per-block-device pengaturan yang seharusnya hanya diterapkan pada volume HDD Anda.

Guna memeriksa nilai read-ahead saat ini untuk perangkat blok Anda, gunakan perintah berikut:

```
[ec2-user ~]$ sudo blockdev --report /dev/<device>
```

Informasi perangkat blok dikembalikan dalam format berikut:

```
RO    RA   SSZ   BSZ   StartSec            Size   Device
rw   256   512  4096       4096      8587820544   /dev/<device>
```

Perangkat yang ditampilkan melaporkan nilai read-ahead sebesar 256 (default). Kalikan angka ini dengan ukuran sektor (512 bita) untuk mendapatkan ukuran buffer read-ahead, yang dalam hal ini adalah 128 KiB. Untuk mengatur nilai buffer ke 1 MiB, gunakan perintah berikut:

```
[ec2-user ~]$ sudo blockdev --setra 2048 /dev/<device>
```

Pastikan bahwa pengaturan read-ahead sekarang menampilkan 2.048 dengan menjalankan kembali perintah pertama.

Hanya gunakan pengaturan ini jika beban kerja Anda terdiri atas I/O berurutan yang besar. Jika terdiri dari I/O yang kecil dan acak, pengaturan ini akan benar-benar menurunkan performa Anda. Secara umum, jika beban kerja Anda sebagian besar terdiri dari I/O kecil atau acak, Anda harus mempertimbangkan untuk menggunakan volume SSD Tujuan Umum (`gp2` dan `gp3`), bukan volume `st1` atau `sc1`.

Gunakan kernel Linux modern (hanya instance Linux)

Gunakan kernel Linux modern dengan dukungan untuk deskriptor tidak langsung. Setiap kernel 3.8 Linux dan di atasnya memiliki dukungan ini, serta setiap instans EC2 generasi saat ini. Jika ukuran I/O rata-rata Anda berada pada atau mendekati 44 KiB, Anda dapat menggunakan instans atau kernel tanpa dukungan deskriptor tidak langsung. Untuk informasi tentang menurunkan ukuran I/O rata-rata dari CloudWatch metrik Amazon, lihat. Karakteristik dan pemantauan Amazon EBS I/O

Untuk mencapai throughput maksimal pada volume `st1` atau `sc1`, kami sarankan untuk menerapkan nilai 256 pada parameter `xen_blkfront.max` (untuk versi kernel Linux di bawah 4.6) atau parameter `xen_blkfront.max_indirect_segments` (untuk versi kernel Linux 4.6 dan yang lebih tinggi). Parameter yang sesuai dapat diatur di baris perintah boot OS Anda.

Misalnya, di dalam AMI Amazon Linux dengan kernel sebelumnya, Anda dapat menambahkannya ke akhir baris kernel di konfigurasi GRUB yang ditemukan di `/boot/grub/menu.lst`:

```
kernel /boot/vmlinuz-4.4.5-15.26.amzn1.x86_64 root=LABEL=/ console=ttyS0
 xen_blkfront.max=256
```

Untuk kernel berikutnya, perintah akan serupa dengan yang berikut ini:

```
kernel /boot/vmlinuz-4.9.20-11.31.amzn1.x86_64 root=LABEL=/ console=tty1 console=ttyS0
 xen_blkfront.max_indirect_segments=256
```

Boot ulang instans Anda agar pengaturan ini berfungsi.

Untuk informasi selengkapnya, lihat Mengkonfigurasi GRUB untuk AMI paravirtual. Distribusi Linux lainnya, terutama yang tidak menggunakan GRUB boot loader, mungkin memerlukan pendekatan yang berbeda untuk menyesuaikan parameter kernel.

Untuk informasi selengkapnya tentang karakteristik I/O EBS, lihat Amazon EBS: Merancang Performa re:Invent presentasi tentang topik ini.

Gunakan RAID 0 untuk memaksimalkan pemanfaatan sumber daya instans

Beberapa tipe instans dapat mendorong lebih banyak throughput I/O dibandingkan yang dapat Anda sediakan untuk satu volume EBS. Anda dapat menggabungkan beberapa volume dalam konfigurasi RAID 0 untuk menggunakan bandwidth yang tersedia untuk instans ini. Untuk informasi selengkapnya, lihat Konfigurasi Amazon EBS dan RAID.

Lacak kinerja menggunakan Amazon CloudWatch

Amazon Web Services menyediakan metrik kinerja untuk Amazon EBS yang dapat Anda analisis dan lihat dengan Amazon CloudWatch dan pemeriksaan status yang dapat Anda gunakan untuk memantau kesehatan volume Anda. Untuk informasi selengkapnya, lihat Pantau volume Amazon EBS Anda.

Optimalkan kinerja Amazon EBS

Instans yang dioptimalkan Amazon EBS menggunakan tumpukan konfigurasi yang dioptimalkan dan memberikan tambahan, kapasitas khusus untuk I/O Amazon EBS. Optimisasi ini memberikan performa terbaik untuk volume EBS Anda dengan meminimalkan pendapat antara I/O Amazon EBS dan lalu lintas lain dari instans Anda.

Instans yang dioptimalkan EBS memberikan bandwidth khusus untuk Amazon EBS. Jika dipasangkan ke instans yang dioptimalkan EBS, volume SSD Tujuan Umum (`gp2` dan `gp3`) dirancang untuk memberikan setidaknya 90% performa IOPS yang tersedia selama 99% waktu di tahun tertentu, dan volume SSD IOPS yang Tersedia (`io1` dan `io2`) dirancang untuk memberikan setidaknya 90% dari performa IOPS yang tersedia selama 99,9% waktu di tahun tertentu. HDD Throughput yang Dioptimalkan (`st1`) dan Cold HDD (`sc1`) memberikan setidaknya 90% performa throughput yang diharapkan 99% dari waktu pada tahun tertentu. Periode yang tidak sesuai didistribusikan kurang lebih secara seragam, yang menargetkan 99% dari total throughput yang diharapkan setiap jam. Untuk informasi selengkapnya, lihat Tipe volume Amazon EBS.

Untuk informasi selengkapnya, lihat Instans Amazon EBS yang dioptimalkan di Panduan Pengguna Amazon EC2.

Karakteristik dan pemantauan Amazon EBS I/O

Pada konfigurasi volume tertentu, karakteristik I/O tertentu mendorong perilaku performa untuk volume EBS Anda. Volume yang didukung SSD—SSD Tujuan Umum (`gp2` dan `gp3`) dan SSD IOPS

yang Tersedia (`io1` dan `io2`)—memberikan performa yang konsisten baik operasi I/O acak atau berurutan. Volume yang didukung HDD—HDD Throughput Dioptimalkan (`st1`) dan Cold HDD (`sc1`)—memberikan performa yang optimal hanya ketika operasi I/O berukuran besar dan berurutan. Untuk memahami cara volume SSD dan HDD akan berjalan di dalam aplikasi Anda, penting untuk mengetahui koneksi antara permintaan volume, jumlah IOPS yang tersedia, waktu yang dibutuhkan untuk penyelesaian operasi I/O, dan batas throughput volume.

Topik

IOPS

IOPS adalah unit pengukuran yang mewakili operasi input/output per detik. Operasi diukur dalam KiB, dan teknologi drive yang mendasarinya menentukan jumlah maksimum data yang dihitung dari tipe volumenya sebagai I/O tunggal. Ukuran I/O dibatasi pada 256 KiB untuk volume SSD dan 1.024 KiB untuk volume HDD karena volume SSD menangani I/O kecil atau acak dengan jauh lebih efisien dibandingkan volume HDD.

Ketika operasi I/O kecil berurutan secara fisik, Amazon EBS mencoba menggabungkannya ke dalam operasi I/O tunggal hingga ukuran I/O maksimum. Demikian pula, ketika operasi I/O lebih besar dari ukuran I/O maksimum, Amazon EBS mencoba untuk membaginya ke dalam operasi I/O yang lebih kecil. Tabel berikut menunjukkan beberapa contoh.

Tipe volume	Ukuran I/O maksimum	Operasi I/O dari aplikasi Anda	Jumlah IOPS	Catatan
SSD	256 KiB	Operasi I/O 1 x 1024 KiB	4 (1.024÷256=4)	Amazon EBS membagi 1.024 operasi I/O menjadi empat operasi berukuran 256 KiB yang lebih kecil.

Tipe volume	Ukuran I/O maksimum	Operasi I/O dari aplikasi Anda	Jumlah IOPS	Catatan
		8 x 32 KiB operasi I/O berurutan	1 (8x32=256)	Amazon EBS menggabungkan delapan operasi I/O berurutan berukuran 32 KiB menjadi 256 operasi KiB tunggal.
		8 acak 32 KiB operasi I/O	8	Amazon EBS menghitung operasi I/O acak secara terpisah.
HDD	1.024 KiB	Operasi I/O 1 x 1024 KiB	1	Operasi I/O sudah sama dengan ukuran I/O maksimum. Hal ini tidak digabung atau dibagi.
		8 x 128 KiB operasi I/O berurutan	1 (8x128=1.024)	Amazon EBS menggabungkan delapan operasi I/O berurutan berukuran 128 KiB menjadi 1.024 operasi I/O KiB tunggal.
		8 acak 32 KiB operasi I/O	8	Amazon EBS menghitung operasi I/O acak secara terpisah.

Akibatnya, ketika Anda membuat volume yang didukung SSD yang mendukung 3.000 IOPS (baik dengan menyediakan volume SSD IOPS yang Tersedia pada 3.000 IOPS atau dengan menskalakan volume SSD Tujuan Umum sebesar 1.000 GiB), dan Anda memasanganya ke instans yang dioptimalkan EBS yang dapat menyediakan cukup bandwidth, Anda dapat mentransfer hingga 3.000 I/O data per detik, dengan throughput ditentukan oleh ukuran I/O.

Panjang antrean volume dan latensi

Panjang antrean volume adalah jumlah permintaan I/O tertunda untuk perangkat. Latensi adalah waktu end-to-end klien sebenarnya dari operasi I/O, dengan kata lain, waktu yang berlalu antara mengirim I/O ke EBS dan menerima pengakuan dari EBS bahwa I/O membaca atau menulis selesai. Panjang antrean harus dikalibrasi dengan benar pada ukuran dan latensi I/O untuk menghindari timbulnya kemacetan pada sistem operasi tamu atau pada tautan jaringan ke EBS.

Lama antrean yang optimal bervariasi untuk setiap beban kerja, tergantung pada sensitivitas aplikasi tertentu Anda terhadap IOPS dan latensi. Jika beban kerja Anda tidak cukup memenuhi permintaan I/O untuk sepenuhnya menggunakan performa yang tersedia bagi volume EBS Anda, volume Anda mungkin tidak dapat mencapai IOPS atau throughput yang telah Anda sediakan.

Aplikasi intensif transaksi bersifat peka terhadap latensi I/O yang meningkat dan sangat cocok untuk volume yang didukung SSD. Anda dapat mempertahankan IOPS yang tinggi sekaligus menjaga latensi tetap rendah dengan mempertahankan panjang antrean yang rendah dan sejumlah besar IOPS yang tersedia untuk volume. Mendorong lebih banyak IOPS ke volume dibandingkan yang tersedia dapat menyebabkan peningkatan latensi I/O.

Aplikasi dengan throughput tinggi kurang sensitif terhadap peningkatan latensi I/O, dan sangat cocok untuk volume yang didukung HDD. Anda dapat mempertahankan throughput yang tinggi ke volume yang didukung HDD dengan mempertahankan panjang antrean yang tinggi ketika melakukan I/O besar yang berurutan.

Ukuran I/O dan batas throughput volume

Untuk volume yang didukung SSD, jika ukuran I/O Anda sangat besar, Anda dapat mengalami jumlah IOPS yang lebih kecil daripada yang Anda sediakan karena Anda mencapai batas throughput volume. Misalnya, volume gp2 di bawah 1.000 GiB dengan kredit lonjakan tersedia memiliki batas IOPS sebesar 3.000 dan batas throughput volume 250 MiB/dtk. Jika Anda menggunakan Ukuran I/O 256 KiB, volume Anda mencapai batas throughput pada 1000 IOPS (1000 x 256 KiB = 250 MiB). Untuk ukuran I/O yang lebih kecil (seperti 16 KiB), volume yang sama dapat mempertahankan 3.000 IOPS karena throughput jauh di bawah 250 MiB/dtk. (Contoh ini mengasumsikan bahwa I/O volume

Anda tidak menyentuh batas throughput dari instans.) Untuk informasi selengkapnya tentang batas throughput untuk setiap tipe volume EBS, lihat Tipe volume Amazon EBS.

Untuk operasi I/O yang lebih kecil, Anda mungkin melihat nilai higher-than-provisioned IOPS yang diukur dari dalam instance Anda. Hal ini terjadi saat sistem operasi instans menggabungkan operasi I/O kecil ke dalam operasi yang lebih besar sebelum meneruskannya ke Amazon EBS.

Jika beban kerja Anda menggunakan I/O berurutan pada volume `st1` dan `sc1` yang didukung HDD, Anda mungkin mengalami jumlah IOPS yang lebih tinggi dari yang diharapkan, yang diukur dari dalam instans Anda. Hal ini terjadi ketika sistem operasi instans menggabungkan I/O berurutan dan menghitungnya di 1.024 unit berukuran KiB. Jika beban kerja Anda menggunakan I/O yang kecil atau acak, Anda dapat mengalami throughput yang lebih rendah dari yang Anda harapkan. Hal ini karena kami menghitung I/O acak tidak berurutan untuk total jumlah IOPS, yang dapat menyebabkan Anda mencapai batas IOPD volume dengan cepat dari yang diharapkan.

Apa pun tipe volume EBS Anda, jika Anda tidak mengalami IOPS atau throughput yang Anda harapkan dalam konfigurasi, pastikan bahwa bandwidth instans EC2 Anda tidak menjadi faktor pembatas. Anda harus selalu menggunakan instans yang dioptimalkan EBS generasi saat ini (atau yang mencakup hubungan jaringan sebesar 10 Gb/dtk) untuk performa yang optimal. Penyebab lain yang mungkin terjadi karena tidak mengalami IOPS yang diharapkan adalah Anda tidak mendorong cukup I/O ke volume EBS.

Pantau karakteristik I/O menggunakan CloudWatch

Anda dapat memantau karakteristik I/O ini dengan metrik volume masing-masing CloudWatch volume. Metrik penting yang perlu dipertimbangkan meliputi hal berikut:

- `VolumeStalledIOCheck`
- `BurstBalance`
- `VolumeReadBytes` | `VolumeWriteBytes`
- `VolumeReadOps` | `VolumeWriteOps`
- `VolumeQueueLength`

`VolumeStalledIOCheck` memantau status volume EBS Anda untuk menentukan kapan volume Anda terganggu. Metrik adalah nilai biner yang akan mengembalikan status `0` (lulus) atau 1 (gagal) berdasarkan apakah volume EBS dapat menyelesaikan operasi I/O atau tidak. Pemeriksaan ini mendeteksi masalah mendasar dengan infrastruktur Amazon EBS, seperti berikut ini:

- Masalah perangkat keras atau perangkat lunak pada subsistem penyimpanan yang mendasari volume EBS
- Masalah perangkat keras pada host fisik yang memengaruhi jangkauan volume EBS dari instans EC2
- Masalah konektivitas antara instans dan volume EBS

Jika `VolumeStalledIOCheck` metrik gagal, Anda dapat menunggu AWS untuk menyelesaikan masalah, atau Anda dapat mengambil tindakan, seperti mengganti volume yang terpengaruh atau menghentikan dan memulai ulang instance tempat volume dilampirkan. Dalam kebanyakan kasus, ketika metrik ini gagal, EBS akan secara otomatis mendiagnosis dan memulihkan volume Anda dalam beberapa menit. Anda dapat menggunakan aksi Jeda I/O AWS Fault Injection Service untuk menjalankan eksperimen terkontrol untuk menguji arsitektur dan pemantauan Anda berdasarkan metrik ini untuk meningkatkan ketahanan Anda terhadap kesalahan penyimpanan.

Anda dapat mengukur latensi I/O penyimpanan Amazon EBS menggunakan `VolumeReadOps`, `VolumeWriteOps`, `VolumeTotalReadTime` dan `VolumeTotalWriteTime`. Anda dapat menggunakan rumus berikut untuk memantau latensi I/O rata-rata volume Anda:

```
Average I/O latency in ms/op = (VolumeTotalReadTime + VolumeTotalWriteTime) /
 (VolumeReadOps + VolumeWriteOps)
```

Jika latensi I/O Anda lebih tinggi dari yang Anda butuhkan, periksa IOPS Anda dan pastikan bahwa aplikasi Anda tidak mencoba untuk mendorong lebih banyak IOPS daripada yang Anda sediakan. Anda dapat menggunakan rumus berikut untuk memantau rata-rata IOPS yang didorong pada volume:

```
Estimated average IOPS in ops/s = (Sum(VolumeReadOps) + Sum(VolumeWriteOps)) / (Period
 - Sum(VolumeIdleTime))
```

Jika aplikasi Anda membutuhkan jumlah IOPS yang lebih besar daripada yang dapat diberikan volume, Anda harus mempertimbangkan untuk menggunakan salah satu dari berikut ini:

- Volume `gp3`, `io2`, atau `io1` yang disediakan dengan IOPS yang cukup untuk mencapai latensi yang diperlukan
- Volume `gp2` yang lebih besar yang memberikan performa IOPS dasar yang cukup

Volume `st1` dan `sc1` yang didukung HDD dirancang untuk melakukan beban kerja terbaik yang memanfaatkan ukuran I/O maksimum 1.024 KiB. Untuk menentukan ukuran I/O rata-rata volume Anda, bagi `VolumeWriteBytes` dengan `VolumeWriteOps`. Penghitungan yang sama berlaku untuk membaca operasi. Jika ukuran I/O rata-rata di bawah 64 KiB, menambah ukuran operasi I/O yang dikirim ke volume `st1` atau `sc1` akan meningkatkan performa.

> **Note**
>
> Jika ukuran I/O rata-rata Anda berada pada atau mendekati 44 KiB, Anda dapat menggunakan instans atau kernel tanpa dukungan deskriptor tidak langsung. Setiap kernel 3.8 Linux dan di atasnya memiliki dukungan ini, serta setiap instans generasi saat ini.

`BurstBalance` menampilkan saldo bucket lonjakan untuk volume `gp2`, `st1`, dan `sc1` sebagai persentase dari saldo yang tersisa. Saat bucket lonjakan Anda habis, I/O volume (untuk volume `gp2`) atau throughput volume (untuk volume `st1` dan `sc1`) dibatasi sesuai acuan. Periksa nilai `BurstBalance` untuk menentukan apakah volume Anda dipacu karena alasan ini. Untuk daftar lengkap metrik Amazon EBS yang tersedia, lihat dan metrik CloudWatch Metrik Amazon untuk Amazon EBS Amazon EBS untuk instans berbasis Nitro.

Sumber daya terkait

Untuk informasi selengkapnya tentang karakteristik I/O Amazon EBS, lihat presentasi re:Invent berikut ini: Amazon EBS: Merancang Performa.

Inisialisasi volume Amazon EBS

Volume EBS yang kosong akan mencapai performa maksimalnya saat dibuat dan tidak memerlukan inisialisasi (sebelumnya dikenal sebagai pra-pemanasan).

Untuk volume, dengan tipe apa pun, yang dibuat dari snapshot, blok penyimpanan harus dihancurkan dari Amazon S3 dan ditulis ke volume sebelum Anda dapat mengaksesnya. Tindakan awal ini memakan banyak waktu dan dapat menyebabkan peningkatan yang signifikan dalam latensi operasi I/O, pada kali pertama setiap blok diakses. Performa volume dicapai setelah semua blok diunduh dan ditulis ke volume.

> **Important**
>
> Saat menginisialisasi volume SSD IOPS yang Tersedia yang dibuat dari snapshot, performa volume dapat turun di bawah 50 persen dari tingkat yang diharapkan, yang menyebabkan volume menampilkan status `warning` dalam pemeriksaan status Performa I/O. Hal ini wajar, dan Anda dapat mengabaikan status `warning` pada volume SSD IOPS yang Tersedia saat Anda menginisialisasinya. Untuk informasi selengkapnya, lihat Pemeriksaan status volume EBS.

Untuk sebagian besar aplikasi, amortisasi biaya inisialisasi selama masa pakai volume dapat diterima. Untuk menghindari lonjakan performa awal di lingkungan produksi, Anda dapat menggunakan salah satu opsi berikut:

- Paksa inisialisasi segera dari seluruh volume. Untuk informasi selengkapnya, lihat Instans Linux (instance Linux) atau Instans Windows (instance Windows).
- Mengaktifkan pemulihan snapshot cepat pada snapshot untuk memastikan bahwa volume EBS yang dibuat sepenuhnya diinisialisasi pada saat pembuatan dan secara instan menyampaikan semua performa yang diberikan. Untuk informasi selengkapnya, lihat Pemulihan snapshot cepat Amazon EBS.

Instans Linux

Untuk menginisialisasi volume yang dibuat dari snapshot di Linux

1. Lampirkan volume yang baru dipulihkan ke instans Linux Anda.
2. Gunakan perintah lsblk untuk mencantumkan perangkat blok pada instans Anda.

```
[ec2-user ~]$ lsblk
NAME  MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvdf  202:80   0  30G  0 disk
xvda1 202:1    0   8G  0 disk /
```

Di sini Anda dapat melihat volume baru, `/dev/xvdf`, terlampir, tetapi tidak terpasang (karena tidak ada jalur yang tercantum di bawah kolom `MOUNTPOINT`).

3. Gunakan utilitas dd atau fio untuk membaca semua blok pada perangkat. Perintah dd diinstal secara default pada sistem Linux, tetapi fio jauh lebih cepat karena memungkinkan pembacaan multialur.

> **Note**
>
> Langkah ini dapat memakan waktu beberapa menit hingga beberapa jam, bergantung pada bandwidth instans EC2, IOPS yang disediakan untuk volume, dan ukuran volume.

[dd] Parameter `if` (file input) harus diatur ke drive yang ingin Anda inisialisasi. Parameter `of` (file output) parameter harus diatur ke perangkat virtual null Linux, `/dev/null`. Parameter `bs` menetapkan ukuran blok operasi baca; untuk performa yang optimal, harus diatur menjadi 1 MB.

> **Important**
>
> Penggunaan yang salah dd dapat dengan mudah menghancurkan data volume. Pastikan untuk mengikuti perintah contoh di bawah ini dengan tepat. Hanya parameter `if=/dev/xvdf` akan bervariasi tergantung pada nama perangkat yang Anda baca.

```
[ec2-user ~]$ sudo dd if=/dev/xvdf of=/dev/null bs=1M
```

[fio] Jika Anda memiliki fio yang diinstal di sistem Anda, gunakan perintah berikut untuk menginisialisasi volume Anda. Parameter `--filename` (file input) harus diatur ke drive yang ingin Anda inisialisasi.

```
[ec2-user ~]$ sudo fio --filename=/dev/xvdf --rw=read --bs=1M --iodepth=32 --
ioengine=libaio --direct=1 --name=volume-initialize
```

Untuk menginstal fio di Amazon Linux, gunakan perintah berikut:

```
sudo yum install -y fio
```

Untuk menginstal fio di Ubuntu, gunakan perintah berikut:

```
sudo apt-get install -y fio
```

Setelah operasi selesai, Anda akan melihat laporan operasi yang sudah dibaca. Volume Anda sekarang siap digunakan. Untuk informasi selengkapnya, lihat Buat volume Amazon EBS tersedia untuk digunakan.

Instans Windows

Sebelum menggunakan alat, kumpulkan informasi lebih jelas tentang disk pada sistem Anda sebagai berikut:

Untuk mengumpulkan informasi tentang disk sistem

1. Gunakan perintah wmic untuk mencantumkan disk yang tersedia di sistem Anda:

   ```
   wmic diskdrive get size,deviceid
   ```

   Berikut ini adalah output contoh:

   ```
   DeviceID            Size
   \\.\PHYSICALDRIVE2  80517265920
   \\.\PHYSICALDRIVE1  80517265920
   \\.\PHYSICALDRIVE0  128849011200
   \\.\PHYSICALDRIVE3  107372805120
   ```

2. Identifikasi disk untuk menginisialisasi menggunakan dd atau fio. Drive `C:` berada di `\\.\PHYSICALDRIVE0`. Anda dapat menggunakan utilitas `diskmgmt.msc` untuk membandingkan huruf drive dengan nomor drive disk jika Anda tidak yakin nomor drive mana yang harus digunakan.

Use the dd utility

Selesaikan prosedur berikut untuk memasang dan menggunakan dd untuk menginisialisasi volume.

Pertimbangan penting

- Langkah ini dapat memakan waktu beberapa menit hingga beberapa jam, bergantung pada bandwidth instans EC2, IOPS yang disediakan untuk volume, dan ukuran volume.

- Penggunaan dd yang salah dapat dengan mudah menghancurkan data volume. Pastikan untuk mengikuti prosedur ini secara tepat.

Untuk menginstal dd untuk Windows

dd untuk program Windows memberikan pengalaman yang serupa dengan program dd yang umumnya tersedia untuk sistem Linux dan Unix, dan memungkinkan menginisialisasi volume Amazon EBS yang dibuat dari snapshot. Versi beta yang paling terbaru mendukung perangkat virtual `/dev/null`. Jika Anda menginstal versi sebelumnya, Anda dapat menggunakan perangkat virtual `nul` sebagai gantinya. Dokumentasi lengkap tersedia di http://www.chrysocome.net/dd.

1. Unduh versi biner yang paling terbaru dari dd untuk Windows dari http://www.chrysocome.net/dd.
2. (Opsional) Buat folder untuk utilitas baris perintah yang mudah ditemukan dan diingat, seperti `C:\bin`. Jika Anda sudah memiliki folder khusus untuk baris perintah, Anda dapat menggunakan folder tersebut pada langkah berikut.
3. Buka paket biner dan salin file `dd.exe` ke folder utilitas baris perintah (misalnya, `C:\bin`).
4. Tambahkan baris perintah folder ke variabel lingkungan Jalur Anda sehingga Anda dapat menjalankan program di folder tersebut dari mana saja.
   a. Pilih Mulai, buka menu konteks (klik kanan) untuk Komputer, lalu pilih Properti.
   b. Pilih Pengaturan sistem lanjutan, Variabel Lingkungan.
   c. Untuk Variabel Sistem, pilih variabel Jalur dan pilih Edit.
   d. Untuk Nilai variabel, tambahkan titik koma dan lokasi folder utilitas baris perintah (**`;C:\bin\`**) ke akhir nilai yang ada.
   e. Pilih OK untuk menutup jendela Edit Variabel Sistem.
5. Buka jendela prompt perintah baru. Langkah sebelumnya tidak memperbarui variabel lingkungan di jendela prompt perintah Anda saat ini. Jendela perintah yang Anda buka sekarang setelah menyelesaikan langkah sebelumnya diperbarui.

Untuk menginisialisasi suatu volume menggunakan dd untuk Windows

Jalankan perintah berikut untuk membaca semua blok pada perangkat yang ditentukan (dan mengirim output ke perangkat virtual `/dev/null`). Perintah ini menginisialisasi data yang ada secara aman.

```
dd if=\\.\PHYSICALDRIVEn of=/dev/null bs=1M --progress --size
```

Anda mungkin mendapatkan kesalahan jika dd mencoba membaca di luar akhir volume. Anda dapat mengabaikan kesalahan ini dengan aman.

Jika Anda menggunakan versi sebelumnya dari perintah dd,perintah tidak mendukung perangkat `/dev/null`. Sebaliknya, Anda dapat menggunakan pernagkat `nul` seperti berikut.

```
dd if=\\.\PHYSICALDRIVEn of=nul bs=1M --progress --size
```

Use the fio utility

Selesaikan prosedur berikut untuk memasang dan menggunakan fio untuk menginisialisasi volume.

Untuk memasang fio untuk Windows

fio untuk program Windows memberikan pengalaman yang serupa dengan program fio yang umumnya tersedia untuk sistem Linux dan Unix, dan memungkinkan Anda untuk menginisialisasi volume Amazon EBS yang dibuat dari snapshot. Untuk informasi selengkapnya, lihat https://github.com/axboe/fio.

1. Unduh penginstal MSI fio dengan memperluas Aset untuk rilis terbaru dan memilih penginstal MSI.
2. Instal fio.

Untuk menginisialisasi suatu volume menggunakan fio untuk Windows

1. Jalankan perintah yang mirip dengan berikut ini untuk menginisialisasi volume:

   ```
   fio --filename=\\.\PHYSICALDRIVEn  --rw=read --bs=128k --iodepth=32 --direct=1
    --name=volume-initialize
   ```

2. Setelah operasi selesai, Anda siap untuk menggunakan volume baru Anda. Untuk informasi selengkapnya, lihat Buat volume Amazon EBS tersedia untuk digunakan.

Konfigurasi Amazon EBS dan RAID

Dengan Amazon EBS, Anda dapat menggunakan salah satu konfigurasi RAID standar yang dapat Anda gunakan dengan server bare metal tradisional, selama konfigurasi RAID tersebut didukung oleh sistem operasi untuk instans Anda. Ini karena semua RAID dilakukan di tingkat perangkat lunak.

Data volume Amazon EBS direplikasi di banyak server di Zona Ketersediaan untuk mencegah hilangnya data akibat kegagalan komponen apa pun. Replikasi ini membuat volume Amazon EBS mencapai 10 kali lebih dapat diandalkan dibandingkan drive disk yang biasa. Untuk informasi selengkapnya, lihat Ketersediaan dan Ketahanan Amazon EBS di halaman detail produk Amazon EBS.

Daftar Isi

Opsi konfigurasi RAID

Membuat rangkaian RAID 0 memungkinkan Anda mencapai tingkat performa yang lebih tinggi untuk sistem file daripada yang dapat Anda berikan dalam satu volume Amazon EBS. Gunakan RAID 0 ketika performa I/O adalah yang paling penting. Dengan RAID 0, I/O didistribusikan di seluruh volume dalam stripe. Jika Anda menambah volume, Anda mendapatkan penambahan langsung dari throughput dan IOPS. Namun, perlu diingat bahwa performa stripe terbatas pada volume beperforma terburuk di set, dan bahwa hilangnya satu volume dalam hasil set dalam kehilangan data lengkap untuk array.

Ukuran yang dihasilkan dari array RAID 0 adalah jumlah ukuran volume didalamnya, dan bandwidth-nya adalah jumlah bandwidth yang tersedia dari volume di dalamnya. Misalnya, dua volume `io1` 500 GiB dengan 4.000 IOPS yang Tersedia masing-masing menciptakan 1000 GiB array RAID 0 dengan bandwidth yang tersedia sebesar 8.000 IOPS dan 1.000 MiB/dtk throughput.

> **Important**
>
> RAID 5 dan RAID 6 tidak disarankan untuk Amazon EBS karena operasi tulis paritas pada mode RAID ini menghabiskan beberapa IOPS yang tersedia untuk volume Anda. Bergantung pada konfigurasi array RAID Anda, mode RAID ini menyediakan IOPS yang dapat digunakan

20-30% lebih sedikit dibandingkan konfigurasi RAID 0. Peningkatan biaya adalah faktor yang juga memengaruhi mode RAID ini; ketika menggunakan ukuran volume dan kecepatan yang sama, array RAID 0 2 volume dapat melampaui array RAID 6 4 volume yang berbiaya dua kali lebih banyak.
RAID 1 juga tidak disarankan untuk digunakan dengan Amazon EBS. RAID 1 memerlukan lebih banyak Amazon EC2 untuk bandwidth Amazon EBS daripada konfigurasi non-RAID karena data ditulis ke beberapa volume secara bersamaan. Selain itu, RAID 1 tidak memberikan peningkatan performa tulis.

Buat array RAID 0

Gunakan prosedur berikut untuk membuat rangkaian RAID 0.

Pertimbangan

- Sebelum Anda melakukan prosedur ini, Anda harus memutuskan seberapa besar array RAID 0 Anda dan berapa banyak IOPS yang akan disediakan.
- Buat ukuran yang sama dan nilai performa IOPS untuk array Anda. Pastikan Anda tidak membuat array yang melebihi bandwidth yang tersedia dari instans EC2 Anda.
- Anda harus menghindari boot dari volume RAID. Jika salah satu perangkat gagal, Anda mungkin tidak dapat mem-boot sistem operasi.

Instans Linux

Untuk membuat array RAID 0 di Linux

1. Buat volume Amazon EBS untuk array Anda. Untuk informasi selengkapnya, lihat Buat volume Amazon EBS.
2. Lampirkan volume Amazon EBS ke instans di mana Anda ingin melakukan hosting array tersebut. Untuk informasi selengkapnya, lihat Lampirkan volume Amazon EBS ke instans.
3. Gunakan perintah mdadm untuk membuat perangkat RAID logis dari volume Amazon EBS yang baru dipasang. Mengganti jumlah volume dalam array Anda untuk *number_of_volumes* dan nama perangkat untuk setiap volume dalam array (seperti `/dev/xvdf`) untuk *device_name*. Anda juga dapat mengganti *MY_RAID* dengan nama unik Anda untuk array ini.

> **Note**
>
> Anda dapat mencantumkan perangkat di instans Anda dengan perintah lsblk untuk menemukan nama perangkat.

Untuk membuat rangkaian RAID 0, jalankan perintah berikut (perhatikan opsi `--level=0` untuk membuat rangkaian strip):

```
[ec2-user ~]$ sudo mdadm --create --verbose /dev/md0 --level=0 --name=MY_RAID --raid-devices=number_of_volumes device_name1 device_name2
```

> **Tip**
>
> Jika Anda mendapatkan kesalahan `mdadm: command not found`, gunakan perintah berikut untuk menginstal mdadm: `sudo yum install mdadm`.

4. Berikan waktu untuk array RAID untuk diinisialisasi dan disinkronkan. Anda dapat melacak kemajuan operasi ini dengan perintah berikut:

```
[ec2-user ~]$ sudo cat /proc/mdstat
```

Berikut ini adalah output contoh:

```
Personalities : [raid0]
md0 : active raid0 xvdc[1] xvdb[0]
      41910272 blocks super 1.2 512k chunks

unused devices: <none>
```

Secara umum, Anda dapat menampilkan informasi terperinci tentang rangkaian RAID Anda dengan perintah berikut:

```
[ec2-user ~]$ sudo mdadm --detail /dev/md0
```

Berikut ini adalah output contoh:

```
/dev/md0:
           Version : 1.2
     Creation Time : Wed May 19 11:12:56 2021
        Raid Level : raid0
        Array Size : 41910272 (39.97 GiB 42.92 GB)
      Raid Devices : 2
     Total Devices : 2
       Persistence : Superblock is persistent

       Update Time : Wed May 19 11:12:56 2021
             State : clean
    Active Devices : 2
   Working Devices : 2
    Failed Devices : 0
     Spare Devices : 0

        Chunk Size : 512K

Consistency Policy : none

              Name : MY_RAID
              UUID : 646aa723:db31bbc7:13c43daf:d5c51e0c
            Events : 0

    Number   Major   Minor   RaidDevice State
       0     202       16        0      active sync   /dev/sdb
       1     202       32        1      active sync   /dev/sdc
```

5. Buat sistem file di array RAID Anda, dan berikan label pada sistem file untuk digunakan saat Anda memasangnya nanti. Misalnya, untuk membuat sistem file ext4 dengan label *MY_RAID*, jalankan perintah berikut:

```
[ec2-user ~]$ sudo mkfs.ext4 -L MY_RAID /dev/md0
```

   Bergantung pada persyaratan aplikasi Anda atau batasan sistem operasi Anda, Anda dapat menggunakan jenis sistem file yang berbeda, seperti ext3 atau XFS (baca dokumentasi sistem file Anda untuk perintah pembuatan sistem file terkait).

6. Untuk memastikan bahwa array RAID dirakit ulang secara otomatis di boot, buat file konfigurasi untuk memuat informasi RAID:

```
[ec2-user ~]$ sudo mdadm --detail --scan | sudo tee -a /etc/mdadm.conf
```

> **Note**
>
> Jika Anda menggunakan distribusi Linux selain Amazon Linux, Anda mungkin perlu memodifikasi perintah ini. Misalnya, Anda mungkin perlu menempatkan file di lokasi yang berbeda, atau Anda mungkin perlu menambahkan parameter `--examine`. Untuk informasi selengkapnya, jalankan man mdadm.conf di instans Linux Anda.

7. Buat image ramdisk baru untuk memuat modul perangkat blok sebelumnya dengan benar untuk konfigurasi RAID Anda yang baru:

```
[ec2-user ~]$ sudo dracut -H -f /boot/initramfs-$(uname -r).img $(uname -r)
```

8. Buat titik pemasangan untuk array RAID Anda.

```
[ec2-user ~]$ sudo mkdir -p /mnt/raid
```

9. Terakhir, pasang perangkat RAID pada titik pemasangan yang Anda buat:

```
[ec2-user ~]$ sudo mount LABEL=MY_RAID /mnt/raid
```

   Perangkat RAID Anda sekarang siap digunakan.

10. (Opsional) Untuk memasang volume Amazon EBS ini pada setiap boot ulang sistem, tambahkan entri untuk perangkat ke file `/etc/fstab` Anda.

    a. Buat cadangan dari file `/etc/fstab` Anda yang dapat digunakan jika Anda secara tidak sengaja menghancurkan atau menghapus file ini saat mengedit.

    ```
    [ec2-user ~]$ sudo cp /etc/fstab /etc/fstab.orig
    ```

    b. Buka file `/etc/fstab` menggunakan editor teks favorit Anda seperti nano atau vim.

    c. Berikan komentar untuk setiap baris yang dimulai dengan "`UUID=`" dan, di akhir file, tambahkan baris baru untuk volume RAID Anda menggunakan format berikut:

    ```
    device_label mount_point file_system_type fs_mntops fs_freq fs_passno
    ```

Tiga kolom terakhir pada baris ini adalah opsi pemasangan sistem file, frekuensi pembuangan sistem file, dan urutan pemeriksaan sistem file yang dilakukan pada waktu booting. Jika Anda tidak tahu nilai-nilai ini, gunakan nilai-nilai di bawah ini untuk mereka (`defaults,nofail 0 2`). Untuk informasi selengkapnya tentang entri `/etc/fstab`, lihat halaman manual fstab (dengan memasukkan man fstab pada baris perintah). Misalnya, untuk memasang sistem file ext4 di perangkat dengan label MY_RAID di titik pemasangan `/mnt/raid`, tambahkan entri berikut ke `/etc/fstab`.

> **Note**
>
> Jika Anda ingin melakukan boot instans tanpa volume terlampir ini (misalnya, sehingga volume ini dapat berpindah bolak-balik antar instans yang berbeda), Anda harus menambahkan opsi pemasangan `nofail` yang memungkinkan instans melakukan boot meskipun terdapat kesalahan dalam pemasangan volume. Derivatif Debian, seperti Ubuntu, juga harus menambah opsi pemasangan `nobootwait`.

```
LABEL=MY_RAID       /mnt/raid   ext4    defaults,nofail        0       2
```

d. Setelah Anda menambahkan entri baru ke `/etc/fstab`, Anda perlu memeriksa bahwa entri Anda berfungsi. Jalankan perintah sudo mount -a untuk memasang semua sistem file di `/etc/fstab`.

```
[ec2-user ~]$ sudo mount -a
```

Jika perintah sebelumnya tidak menghasilkan kesalahan, file `/etc/fstab` Anda baik-baik saja dan sistem file Anda akan terpasang secara otomatis di boot berikutnya. Jika perintah tidak menyebabkan kesalahan apa pun, AMI kesalahan tersebut dan coba koreksi `/etc/fstab`.

> **Warning**
>
> Kesalahan dalam file `/etc/fstab` dapat membuat sistem tidak dapat dibooting. Jangan mematikan sistem yang memiliki kesalahan di file `/etc/fstab` Anda.

e. (Opsional) Jika Anda tidak yakin cara mengoreksi kesalahan `/etc/fstab`, Anda selalu dapat memulihkan file `/etc/fstab` cadangan dengan perintah berikut.

```
[ec2-user ~]$ sudo mv /etc/fstab.orig /etc/fstab
```

Instans Windows

Untuk membuat array RAID 0 di Windows

1. Buat volume Amazon EBS untuk array Anda. Untuk informasi selengkapnya, lihat Buat volume Amazon EBS.
2. Lampirkan volume Amazon EBS ke instans di mana Anda ingin melakukan hosting array tersebut. Untuk informasi selengkapnya, lihat Lampirkan volume Amazon EBS ke instans.
3. Hubungkan ke instans Windows Anda. Untuk informasi selengkapnya, lihat Terhubung ke instans Windows Anda.
4. Buka jendela perintah dan ketikkan perintah diskpart.

```
diskpart

Microsoft DiskPart version 6.1.7601
Copyright (C) 1999-2008 Microsoft Corporation.
On computer: WIN-BM6QPPL51CO
```

5. Pada perintah DISKPART, buat daftar disk yang tersedia dengan perintah berikut.

```
DISKPART> list disk

  Disk ###  Status         Size     Free     Dyn  Gpt
  --------  -------------  -------  -------  ---  ---
  Disk 0    Online           30 GB      0 B
  Disk 1    Online            8 GB      0 B
  Disk 2    Online            8 GB      0 B
```

   Identifikasi disk yang ingin Anda gunakan dalam array Anda dan catat nomor disknya.

6. Setiap disk yang ingin Anda gunakan dalam array Anda harus berupa disk dinamik online yang tidak berisi volume apa pun yang ada. Gunakan langkah-langkah berikut untuk mengonversi disk dasar menjadi disk dinamik dan untuk menghapus volume yang ada.

   a. Pilih disk yang ingin Anda gunakan dalam array Anda dengan perintah berikut, dengan menggantikan *n* dengan nomor disk Anda.

```
DISKPART> select disk n

Disk n is now the selected disk.
```

b. Jika disk yang dipilih tercantum sebagai `Offline`, bawa online dengan menjalankan perintah online disk.

c. Jika disk yang dipilih tidak memiliki tanda bintang dalam kolom Dyn di output perintah list disk sebelumnya, Anda perlu mengonversinya ke disk dinamis.

```
DISKPART> convert dynamic
```

> **Note**
>
> Jika Anda menerima kesalahan bahwa disk tidak dapat ditulis, Anda dapat menghapus tanda hanya-baca dengan perintah ATTRIBUTE DISK CLEAR READONLY kemudian coba konversi disk dinamis lagi.

d. Gunakan perintah detail disk untuk memeriksa volume yang ada pada disk yang dipilih.

```
DISKPART> detail disk

XENSRC PVDISK SCSI Disk Device
Disk ID: 2D8BF659
Type   : SCSI
Status : Online
Path   : 0
Target : 1
LUN ID : 0
Location Path : PCIROOT(0)#PCI(0300)#SCSI(P00T01L00)
Current Read-only State : No
Read-only  : No
Boot Disk  : No
Pagefile Disk  : No
Hibernation File Disk  : No
Crashdump Disk  : No
Clustered Disk  : No

  Volume ###  Ltr  Label        Fs     Type        Size     Status     Info
  ----------  ---  -----------  -----  ----------  -------  ---------  --------
```

```
  Volume 2     D   NEW VOLUME   FAT32  Simple      8189 MB  Healthy
```

Perhatikan nomor volume pada disk. Dalam contoh ini, jumlah volume adalah 2. Jika tidak ada volume, Anda dapat melewati langkah berikutnya.

e. (Hanya diperlukan jika volume diidentifikasi di langkah sebelumnya) Pilih dan hapus volume yang ada pada disk yang Anda identifikasi di langkah sebelumnya.

> ⚠ Warning
>
> Ini menghancurkan semua data yang ada pada volume.

i. Pilih volume, menggantikan *n* dengan jumlah volume Anda.

```
DISKPART> select volume n
Volume n is the selected volume.
```

ii. Hapus volume.

```
DISKPART> delete volume

DiskPart successfully deleted the volume.
```

iii. Ulangi sublangkah ini untuk setiap volume yang perlu dihapus pada disk yang dipilih.

f. Ulangi Step 6 untuk setiap disk yang ingin Anda gunakan dalam array.

7. Verifikasi bahwa disk yang ingin Anda gunakan sekarang adalah dinamik. Dalam kasus ini, kami menggunakan disk 1 dan 2 untuk volume RAID.

```
DISKPART> list disk

  Disk ###  Status         Size     Free     Dyn  Gpt
  --------  -------------  -------  -------  ---  ---
  Disk 0    Online           30 GB      0 B
  Disk 1    Online            8 GB      0 B   *
  Disk 2    Online            8 GB      0 B   *
```

8. Buat array raid Anda. Pada Windows, volume RAID 0 disebut sebagai volume striped.

Untuk membuat array volume striped pada disk 1 dan 2, gunakan perintah berikut (perhatikan opsi `stripe` untuk membuat strip array):

```
DISKPART> create volume stripe disk=1,2
DiskPart successfully created the volume.
```

9. Verifikasi volume baru Anda.

```
DISKPART> list volume

  DISKPART> list volume

  Volume ###  Ltr  Label        Fs     Type        Size     Status     Info
  ----------  ---  -----------  -----  ----------  -------  ---------  --------
  Volume 0     C                NTFS   Partition     29 GB  Healthy    System
  Volume 1                      RAW    Stripe        15 GB  Healthy
```

Perhatikan bahwa kolom `Type` sekarang menunjukkan bahwa Volume 1 adalah volume `stripe`.

10. Pilih dan format volume Anda sehingga Anda dapat mulai menggunakannya.

    a. Pilih volume yang ingin Anda format, menggantikan *n* dengan jumlah volume Anda.

    ```
    DISKPART> select volume n

    Volume n is the selected volume.
    ```

    b. Format volume.

    > **Note**
    >
    > Untuk melakukan format penuh, hapus opsi `quick`.

    ```
    DISKPART> format quick recommended label="My new volume"

      100 percent completed

    DiskPart successfully formatted the volume.
    ```

    c. Tetapkan huruf drive yang tersedia untuk volume Anda.

    ```
    DISKPART> assign letter f
    ```

```
DiskPart successfully assigned the drive letter or mount point.
```

Volume baru Anda sekarang siap digunakan.

Buat snapshot volume dalam suatu array RAID

Jika Anda ingin mencadangkan data pada volume EBS dalam array RAID menggunakan snapshot, Anda harus memastikan bahwa snapshot konsisten. Ini karena snapshot volume ini dibuat secara independen. Untuk memulihkan volume EBS dalam array RAID dari snapshot yang tidak selaras akan menurunkan integritas dari array.

Untuk membuat satu set snapshot konsisten untuk rangkaian RAID Anda, gunakan Snapshot multivolume EBS. Snapshot multi-volume memungkinkan Anda mengambil point-in-time, mengkoordinasikan data, dan snapshot yang konsisten dengan crash di beberapa volume EBS yang dilampirkan ke instans EC2. Anda tidak perlu menghentikan instans untuk berkoordinasi antar volume guna memastikan konsistensi karena snapshot diambil secara otomatis di berbagai volume EBS. Untuk informasi selengkapnya, lihat langkah-langkah untuk membuat snapshot EBS multivolume di Membuat snapshot Amazon EBS.

Tolok ukur volume EBS

Anda dapat menguji performa volume Amazon EBS dengan menyimulasikan beban kerja I/O. Prosesnya adalah sebagai berikut:

1. Luncurkan instans yang dioptimalkan EBS.
2. Buat volume EBS baru.
3. Lampirkan volume ke instans yang dioptimalkan EBS.
4. Konfigurasikan dan pasang perangkat blok.
5. Pasang alat untuk menetapkan tolok ukur performa I/O.
6. Tolok ukur performa I/O dari volume Anda.
7. Hapus volume Anda dan akhiri instans Anda sehingga Anda tidak terus membebankan biaya.

> **Important**
>
> Beberapa prosedur mengakibatkan penghancuran data yang ada pada volume EBS yang menjadi patokan. Prosedur tolok ukur dimaksudkan untuk digunakan pada volume yang dibuat khusus untuk tujuan pengujian, bukan volume produksi.

Siapkan instans Anda

Untuk mendapatkan performa optimal dari volume EBS, kami menyarankan agar Anda menggunakan instans yang dioptimalkan dengan EBS. Instans yang mengoptimalkan EBS memberikan throughput khusus di antara Amazon EC2 dan Amazon EBS, dengan instans. Instans yang dioptimalkan untuk EBS memberikan bandwidth khusus antara Amazon EC2 dan Amazon EBS, spesifikasi tergantung pada tipe instans.

Untuk membuat instans yang dioptimalkan EBS, pilih Luncurkan sebagai instans yang dioptimalkan EBS saat meluncurkan instans menggunakan konsol Amazon EC2, atau tentukan --ebs-optimized saat menggunakan baris perintah. Pastikan Anda memilih jenis instance yang mendukung opsi ini.

Menyiapkan volume SSD IOPS yang Tersedia atau SSD Tujuan Umum

Untuk menciptakan volume SSD IOPS yang Tersedia (`io1` dan `io2`) atau SSD Tujuan Umum (`gp2` dan `gp3`) menggunakan konsol Amazon EC2, untuk Tipe volume, pilih SSD IOPS yang Tersedia (io1), SSD IOPS yang Tersedia (io2), SSD Tujuan Umum (gp2), atau SSD Tujuan Umum (gp3). Di baris perintah, tentukan`io1`, `io2`, `gp2`, atau `gp3` untuk parameter --volume-type. Untuk volume `io1`, `io2`, dan `gp3`, tentukan jumlah operasi I/O per detik (IOPS) untuk parameter --iops. Untuk informasi lebih lanjut, lihat Tipe volume Amazon EBS dan Buat volume Amazon EBS.

(Hanya instance Linux) Untuk contoh pengujian, kami menyarankan Anda membuat array RAID 0 dengan 6 volume, yang menawarkan kinerja tingkat tinggi. Karena Anda dikenai biaya berdasarkan gigabita yang disediakan (dan jumlah IOPS yang Tersedia untuk volume io1, io2, dan gp3), bukan jumlah volume, tidak ada biaya tambahan untuk membuat beberapa volume yang lebih kecil dan menggunakannya untuk membuat set stripe. Jika Anda menggunakan Oracle Orion untuk mengukur volume Anda, Oracle Orion dapat melakukan simulasi striping dengan cara yang sama seperti yang dilakukan Oracle ASM, jadi sebaiknya biarkan Orion yang melakukan striping. Jika Anda menggunakan alat tolok ukur yang berbeda, Anda perlu membuat volume sendiri.

Untuk informasi selengkapnya tentang cara membuat array RAID 0, lihatBuat array RAID 0.

Siapkan volume HDD Throughput Dioptimalkan (**st1**) atau Cold HDD (**sc1**)

Untuk membuat volume `st1`, pilih HDD Throughput Dioptimalkan saat membuat volume menggunakan konsol Amazon EC2, atau tentukan --type **st1** saat menggunakan baris perintah. Untuk membuat volume `sc1`, pilih HDD Throughput Dioptimalkan saat membuat volume menggunakan konsol Amazon EC2, atau tentukan --type **sc1** saat menggunakan baris perintah. Untuk informasi tentang pembuatan volume EBS, lihat Buat volume Amazon EBS. Untuk informasi tentang memasang volume ini ke instans Anda, lihat Lampirkan volume Amazon EBS ke instans.

(Hanya instance Linux) AWS menyediakan template JSON untuk digunakan AWS CloudFormation yang menyederhanakan prosedur penyiapan ini. Akses template dan simpan sebagai file JSON. AWS CloudFormation memungkinkan Anda mengonfigurasi kunci SSH Anda sendiri dan menawarkan cara yang lebih mudah untuk mengatur lingkungan pengujian kinerja untuk mengevaluasi `st1` volume. Templat membuat instans generasi saat ini dan 2 TiB volume `st1`, dan memasangkan volume ke instans pada `/dev/xvdf`.

(Hanya instance Linux) Untuk membuat volume HDD menggunakan template

1. Buka AWS CloudFormation konsol di https://console.aws.amazon.com/cloudformation.
2. Pilih Buat tumpukan.
3. Pilih Unggah Templat ke Amazon S3 dan pilih templat JSON yang Anda dapatkan sebelumnya.
4. Berikan nama tumpukan Anda seperti “ebs-perf-testing”, dan pilih tipe instans (defaultnya adalah r3.8xlarge) dan kunci SSH.
5. Pilih Selanjutnya dua kali, lalu pilih Buat Tumpukan.
6. Setelah status untuk tumpukan baru Anda berpindah dari CREATE_IN_PROGRESS ke COMPLETE, pilih Output untuk mendapatkan entri DNS publik untuk instans baru Anda, yang akan memiliki volume `st1` 2 TiB yang terlampir padanya.
7. Terhubung menggunakan SSH ke tumpukan baru Anda sebagai pengguna **ec2-user**, dengan nama host yang diperoleh dari entri DNS di langkah sebelumnya.
8. Lanjut ke Pasang alat tolok ukur.

Pasang alat tolok ukur

Tabel berikut mencantumkan beberapa alat yang mungkin dapat Anda gunakan untuk mengukur kinerja volume EBS.

Instans Linux

Alat	Deskripsi

fio	Untuk tolok ukur performa I/O. (Perhatikan bahwa fio memiliki ketergantungan pada `libaio-devel` .)
	
	Untuk menginstal fio di Amazon Linux, jalankan perintah berikut:
	
	`[ec2-user ~]$ sudo yum install -y fio`
	
	Untuk memasang fio di Ubuntu, jalankan perintah berikut:
	
	`sudo apt-get install -y fio`

Alat Kalibrasi Orion Oracle	Untuk mengalibrasi performa sistem penyimpanan I/O yang akan digunakan dalam basis data Oracle.

Instans Windows

Alat	Deskripsi
DiskSpd	DiskSpd adalah alat kinerja penyimpanan dari tim teknik Windows, Windows Server, dan Cloud Server Infrastructure di Microsoft. Tersedia untuk diunduh di https://github.com/Microsoft/diskspd/releases. Setelah Anda mengunduh file `diskspd.exe` yang dapat dijalankan, buka command promt dengan wewenang administratif (dengan memilih "Run as Administrator"), dan kemudian navigasi ke direktori tempat Anda menyalin file `diskspd.exe` . Salin yang diinginkan file `diskspd.exe` yang dapat dieksekusi dari folder executable yang sesuai (`amd64fre`, `armfre` atau `x86fre`) ke jalur yang singkat dan sederhana seperti `C:\DiskSpd` . Dalam kebanyakan kasus, Anda akan menginginkan versi 64-bit DiskSpd dari `amd64fre` folder.

Alat	Deskripsi
	Kode sumber untuk DiskSpd di-host GitHub di: https://github.com/Microsoft/diskspd.
CrystalDiskTandai	CrystalDiskMark adalah perangkat lunak benchmark disk sederhana. Tersedia untuk diunduh di https://crystalmark.info/en/software/crystaldiskmark/.

Alat tolok ukur ini mendukung berbagai macam parameter uji. Anda harus menggunakan perintah yang akan mendukung oleh perkiraan beban kerja volume Anda. Perintah yang diberikan di bawah ini dimaksudkan sebagai contoh untuk membantu Anda memulai.

Pilih panjang antrean volume

Memilih panjang antrean volume terbaik berdasarkan beban kerja dan tipe volume Anda.

Panjang antrean pada volume yang didukung SSD

Untuk menentukan panjang antrean yang optimal untuk beban kerja Anda pada volume yang didukung SSD, kami menyarankan Anda menargetkan panjang antrean 1 untuk setiap 1000 IOPS yang disediakan (acuan untuk volume SSD Tujuan Umum dan jumlah yang disediakan untuk volume SSD IOPS yang Tersedia). Kemudian, Anda dapat memantau performa aplikasi Anda dan menyesuaikan nilai tersebut berdasarkan kebutuhan aplikasi Anda.

Peningkatan panjang antrean akan bermanfaat hingga Anda mencapai IOPS yang tersedia , throughput, atau panjang antrean sistem optimal, yang saat ini ditetapkan ke 32. Misalnya, volume dengan 3.000 IOPS yang Tersedia harus menargetkan panjang antrean 3. Anda harus bereksperimen mengatur nilai-nilai ini ke atas atau ke bawah untuk melihat apa yang terbaik untuk aplikasi Anda.

Panjang antrean pada volume yang didukung HDD

Untuk menentukan panjang antrean yang optimal untuk beban kerja Anda pada volume yang didukung HDD, kami sarankan agar Anda menargetkan panjang antrean minimal 4 sambil melakukan I/O berurutan 1MiB. Kemudian, Anda dapat memantau performa aplikasi Anda dan menyesuaikan nilai tersebut berdasarkan kebutuhan aplikasi Anda. Misalnya, volume `st1` 2 TiB dengan throughput lonjakan sebesar 500 MiB/dtk dan IOPS sebesar 500 harus menargetkan panjang antrean 4, 8, atau 16 saat menjalankan I/O 1.024 KiB, 512 KiB, atau 256 KiB secara berurutan. Anda harus

bereksperimen mengatur nilai-nilai ini ke atas atau ke bawah untuk melihat apa yang terbaik untuk aplikasi Anda.

Nonaktifkan Status C

Sebelum menjalankan benchmarking, Anda harus menonaktifkan prosesor C-states. Inti yang sementara diam di CPU yang mendukung dapat memasuki status C untuk menghemat daya. Ketika inti dipanggil untuk melanjutkan pemrosesan, beberapa waktu berlalu sampai inti beroperasi GApenuh. Latensi ini dapat mengganggu rutinitas tolok ukur prosesor . Untuk informasi selengkapnya tentang status C dan tipe instans EC2 mana yang mendukungnya, lihat Kontrol status processor untuk Instans EC2 Anda.

Instans Linux

Anda dapat menonaktifkan status C di Amazon Linux, RHEL, dan CentOS sebagai berikut:

1. Dapatkan jumlah C-state.

```
$ C:\> cpupower idle-info | grep "Number of idle states:"
```

2. Nonaktifkan status C dari c1 ke cN. Idealnya, inti harus berada dalam keadaan c0.

```
$ C:\> for i in `seq 1 $((N-1))`; do cpupower idle-set -d $i; done
```

Instans Windows

Anda dapat menonaktifkan C-states pada Windows sebagai berikut:

1. Masuk PowerShell, dapatkan skema daya aktif saat ini.

```
$current_scheme = powercfg /getactivescheme
```

2. Dapatkan skema daya GUID.

```
(Get-WmiObject -class Win32_PowerPlan -Namespace "root\cimv2\power" -Filter
 "ElementName='High performance'").InstanceID
```

3. Dapatkan pengaturan daya GUID.

```
(Get-WmiObject -class Win32_PowerSetting -Namespace "root\cimv2\power" -Filter
 "ElementName='Processor idle disable'").InstanceID
```

4. Dapatkan pengaturan daya subgrup GUID.

```
(Get-WmiObject -class Win32_PowerSettingSubgroup -Namespace "root\cimv2\power" -
Filter "ElementName='Processor power management'").InstanceID
```

5. Nonaktifkan status C dengan mengatur nilai indeks ke 1. Nilai 0 menunjukkan bahwa status-C dinonaktifkan.

```
powercfg /
setacvalueindex <power_scheme_guid> <power_setting_subgroup_guid> <power_setting_guid>
 1
```

6. Tetapkan skema aktif untuk memastikan pengaturan disimpan.

```
powercfg /setactive <power_scheme_guid>
```

Lakukan benchmarking

Prosedur berikut menjelaskan perintah tolok ukur untuk berbagai tipe volume EBS.

Jalankan perintah berikut pada instans EBS yang dioptimalkan yang memasang volume EBS. Jika volume EBS dibuat dari snapshot, pastikan untuksebelum menetapkan tolok ukur. Untuk informasi selengkapnya, lihat Inisialisasi volume Amazon EBS.

Setelah selesai menguji volume, lihat topik berikut untuk bantuan pembersihan: Menghapus volume Amazon EBS dan Hentikan instans Anda.

Tolok Ukur Volume SSD IOPS yang Tersedia dan SSD Tujuan Umum

Instans Linux

Jalankan fio pada array RAID 0 yang Anda buat.

Perintah berikut melakukan operasiacak 16 KB.

```
[ec2-user ~]$ sudo fio --directory=/mnt/p_iops_vol0 --ioengine=psync --
name fio_test_file --direct=1 --rw=randwrite --bs=16k --size=1G --numjobs=16 --
time_based --runtime=180 --group_reporting --norandommap
```

Perintah berikut melakukan operasi baca acak 16 KB.

```
[ec2-user ~]$ sudo fio --directory=/mnt/p_iops_vol0 --name fio_test_file --direct=1
 --rw=randread --bs=16k --size=1G --numjobs=16 --time_based --runtime=180 --
group_reporting --norandommap
```

Untuk informasi selengkapnya tentang penafsiran hasil, lihat tutorial ini: Memeriksa performa IO disk dengan fio.

Instans Windows

Jalankan DiskSpd pada volume yang Anda buat.

Perintah berikut akan menjalankan uji I/O acak 30 detik menggunakan file uji 20 GB yang berada di drive `C:`, 25% dan 75% rasio baca, dan 8K ukuran blok. Ini akan menggunakan delapan thread bekerja, masing-masing dengan empat I/O luar biasa, dan benih nilai entropi tulis 1GB. Hasil uji akan disimpan ke file teks yang disebut `DiskSpeedResults.txt`. Parameter ini mensimulasikan beban kerja SQL Server OLTP.

```
diskspd -b8K -d30 -o4 -t8 -h -r -w25 -L -Z1G -c20G C:\iotest.dat > DiskSpeedResults.txt
```

Untuk informasi selengkapnya tentang penafsiran hasil, lihat tutorial ini: Memeriksa performa IO disk dengan fio.

Benchmark **`st1`** dan **`sc1`** volume (instance Linux)

Jalankan fio pada volume `st1` atau `sc1`.

> **Note**
>
> Sebelum menjalankan pengujian ini, atur I/O berpenyangga pada instans Anda seperti yang dijelaskan di Tingkatkan read-ahead untuk throughput tinggi, beban kerja read-heavy pada dan (hanya instance Linux) *`st1 sc1`*.

Perintah berikut melakukan operasi pembacaan berurutan 1 MiB terhadap perangkat blok `st1` terlampir (misalnya, `/dev/xvdf`):

```
[ec2-user ~]$ sudo fio --filename=/dev/<device> --direct=1 --rw=read --randrepeat=0
 --ioengine=libaio --bs=1024k --iodepth=8 --time_based=1 --runtime=180 --
name=fio_direct_read_test
```

Perintah berikut melakukan operasi pembacaan berurutan 1 MiB terhadap perangkat blok `st1` yang terlampir:

```
[ec2-user ~]$ sudo fio --filename=/dev/<device> --direct=1 --rw=write --randrepeat=0
 --ioengine=libaio --bs=1024k --iodepth=8 --time_based=1 --runtime=180 --
name=fio_direct_write_test
```

Beberapa beban kerja melakukan campuran antara baca berurutan danberurutan ke bagian perangkat blok yang berbeda. Untuk mengukur beban kerja tersebut, kami sarankan agar Anda menggunakan pekerjaan fio untuk membaca serta menggunakan opsi fio `offset_increment` untuk menargetkan lokasi perangkat blok yang berbeda untuk setiap pekerjaan.

Menjalankan beban kerja ini adalah yang lebih rumit dibandingkan dengan beban kerja baca-urut atau tulis-urut. Gunakan editor teks untuk membuat file pekerjaan fio, yang disebut `fio_rw_mix.cfg` dalam contoh ini, yang berisi hal berikut:

```
[global]
clocksource=clock_gettime
randrepeat=0
runtime=180

[sequential-write]
bs=1M
ioengine=libaio
direct=1
iodepth=8
filename=/dev/<device>
do_verify=0
rw=write
rwmixread=0
rwmixwrite=100

[sequential-read]
bs=1M
```

```
ioengine=libaio
direct=1
iodepth=8
filename=/dev/<device>
do_verify=0
rw=read
rwmixread=100
rwmixwrite=0
offset=100g
```

Kemudian jalankan perintah berikut:

```
[ec2-user ~]$ sudo fio fio_rw_mix.cfg
```

Untuk informasi selengkapnya tentang penafsiran hasil, lihat tutorial ini: Memeriksa performa IO disk dengan fio.

Banyak pekerjaan fio untuk I/O langsung, meskipun menggunakan operasi baca atau tulis berurutan, dapat menghasilkan throughput yang lebih rendah dari yang diharapkan untuk volume `st1` dan `sc1`. Kami sarankan Anda menggunakan satu pekerjaan langsung I/O dan gunakan parameter `iodepth` untuk mengontrol jumlah operasi I/O bersamaan.

Amazon Data Lifecycle Manager

Anda dapat menggunakan Amazon Data Lifecycle Manager untuk mengotomatisasi pembuatan, retensi, dan penghapusan snapshot EBS dan AMI yang didukung EBS. Ketika Anda mengotomatisasi snapshot dan manajemen AMI, hal ini membantu Anda untuk:

- Lindungi data berharga dengan menerapkan jadwal pencadangan rutin.
- Buat AMI terstandardisasi yang dapat dimuat ulang secara berkala.
- Mempertahankan cadangan sebagaimana diwajibkan oleh auditor atau kepatuhan internal.
- Mengurangi biaya penyimpanan dengan menghapus cadangan yang usang.
- Membuat kebijakan cadangan pemulihan bencana yang membuat cadangan data ke Wilayah atau akun terisolasi.

Jika digabungkan dengan fitur pemantauan Amazon EventBridge dan AWS CloudTrail, Amazon Data Lifecycle Manager menyediakan solusi pencadangan lengkap untuk instans Amazon EC2 dan volume EBS individual tanpa biaya tambahan.

Important

- Amazon Data Lifecycle Manager tidak dapat mengelola snapshot atau AMI yang dibuat dengan cara lain.
- Amazon Data Lifecycle Manager tidak dapat mengotomatisasi pembuatan, retensi, dan penghapusan AMI yang didukung penyimpanan instans.

Daftar Isi

- Memantau siklus hidup snapshot dan AMI
- Pemecahan Masalah

Kuota

AWS Akun Anda memiliki kuota berikut yang terkait dengan Amazon Data Lifecycle Manager:

Deskripsi	Kuota
Kebijakan siklus hidup kustom per Wilayah	100
Kebijakan default untuk snapshot EBS per Wilayah	1
Kebijakan default untuk AMI yang didukung EBS	1
Tanda per sumber daya	45

Cara kerja Amazon Data Lifecycle Manager

Berikut ini adalah elemen utama dari Amazon Data Lifecycle Manager.

Elemen

- Kebijakan
- Jadwal kebijakan (hanya kebijakan kustom)
- Tanda sumber daya target (hanya kebijakan kustom)
- Snapshot
- AMI berdukungan EBS
- Tanda Amazon Data Lifecycle Manager

Kebijakan

Dengan Amazon Data Lifecycle Manager, Anda membuat kebijakan untuk menentukan persyaratan pembuatan dan retensi cadangan. Kebijakan ini biasanya menentukan hal berikut:

- Jenis kebijakan - Menentukan jenis sumber daya cadangan yang dikelola kebijakan (snapshot atau AMI yang didukung EBS).
- Sumber daya target — Menentukan jenis sumber daya yang ditargetkan oleh kebijakan (instans atau volume EBS).
- Frekuensi pembuatan — Menentukan seberapa sering kebijakan berjalan dan membuat snapshot atau AMI.
- Ambang batas retensi — Menentukan berapa lama kebijakan mempertahankan snapshot atau AMI setelah dibuat.
- Tindakan tambahan — Menentukan tindakan tambahan yang harus dilakukan kebijakan, seperti penyalinan, pengarsipan, atau penandaan sumber daya lintas wilayah.

Amazon Data Lifecycle Manager menawarkan kebijakan default dan kebijakan kustom.

Kebijakan default

Kebijakan default mencadangkan semua volume dan instans di Wilayah yang tidak memiliki cadangan terbaru. Anda dapat mengecualikan volume dan instans secara opsional dengan menentukan parameter pengecualian.

Amazon Data Lifecycle Manager mendukung kebijakan default berikut:

- Kebijakan default untuk snapshot EBS — Menargetkan volume dan mengotomatiskan pembuatan, retensi, dan penghapusan snapshot.
- Kebijakan default untuk AMI yang didukung EBS — Menargetkan instans dan mengotomatisasi pembuatan, retensi, dan pembatalan pendaftaran AMI yang didukung EBS.

Anda hanya dapat memiliki satu kebijakan default per jenis sumber daya di setiap akun dan Wilayah AWS .

Kebijakan kustom

Kebijakan kustom menargetkan sumber daya tertentu berdasarkan tanda yang ditetapkan dan mendukung fitur-fitur canggih, seperti pemulihan snapshot cepat, pengarsipan snapshot, penyalinan

lintas akun, serta skrip pra dan pasca. Kebijakan kustom dapat mencakup hingga 4 jadwal, di mana setiap jadwal dapat memiliki frekuensi pembuatan sendiri, ambang retensi, dan konfigurasi fitur lanjutan.

Amazon Data Lifecycle Manager mendukung kebijakan kustom berikut ini:

- Kebijakan default untuk snapshot EBS — Menargetkan volume atau instans dan mengotomatiskan pembuatan, retensi, dan penghapusan snapshot.
- Kebijakan AMI yang didukung EBS — Menargetkan instans dan mengotomatisasi pembuatan, retensi, dan pembatalan pendaftaran AMI yang didukung EBS.
- Kebijakan peristiwa penyalinan lintas akun — Mengotomatiskan tindakan penyalinan lintas-Wilayah untuk snapshot yang dibagikan dengan Anda.

Untuk informasi selengkapnya, lihat Kebijakan default vs kebijakan kustom.

Jadwal kebijakan (hanya kebijakan kustom)

Jadwal kebijakan menentukan kapan snapshot atau AMI yang dibuat oleh kebijakan. Kebijakan dapat memiliki hingga empat jadwal—satu jadwal wajib, dan hingga tiga jadwal opsional.

Menambahkan beberapa jadwal ke satu kebijakan memungkinkan Anda membuat snapshot atau AMI pada frekuensi yang berbeda menggunakan kebijakan yang sama. Misalnya, Anda dapat membuat satu kebijakan yang membuat snapshot harian, mingguan, bulanan, dan tahunan. Hal ini menghilangkan kebutuhan untuk mengelola beberapa kebijakan.

Untuk setiap jadwal, Anda dapat menentukan frekuensi, pengaturan pemulihan snapshot cepat (hanya kebijakan siklus hidup snapshot), aturan salinan lintas-Wilayah, dan tanda. Tanda yang ditetapkan ke jadwal secara otomatis ditetapkan ke snapshot atau AMI yang dibuat saat jadwal dimulai. Selain itu, Amazon Data Lifecycle Manager secara otomatis menetapkan tanda yang dihasilkan sistem berdasarkan frekuensi jadwal ke setiap snapshot atau AMI.

Setiap jadwal dimulai secara individual didasarkan pada frekuensinya. Jika beberapa jadwal dimulai secara bersamaan, Amazon Data Lifecycle Manager hanya membuat satu snapshot atau AMI dan menerapkan pengaturan retensi jadwal yang memiliki periode penyimpanan tertinggi. Tanda semua jadwal yang dimulai akan diterapkan ke snapshot atau AMI.

- (Kebijakan siklus hidup snapshot saja) Jika lebih dari satu jadwal yang dimulai diaktifkan untuk pemulihan snapshot cepat, snapshot akan diaktifkan untuk pemulihan snapshot cepat di semua

Zona Ketersediaan yang ditentukan di semua jadwal yang dimulai. Pengaturan retensi tertinggi untuk jadwal yang dimulai akan digunakan untuk setiap Zona Ketersediaan.

- Jika lebih dari satu jadwal yang dimulai diaktifkan untuk penyalinan lintas-Wilayah, snapshot atau AMI disalin ke semua Wilayah yang ditentukan di semua jadwal yang dimulai. Periode retensi tertinggi dari jadwal yang dimulai diterapkan.

Tanda sumber daya target (hanya kebijakan kustom)

Kebijakan kustom Amazon Data Lifecycle Manager menggunakan tanda sumber daya untuk mengidentifikasi sumber daya yang akan dicadangkan. Saat membuat snapshot atau kebijakan AMI yang didukung EBS, Anda dapat menentukan beberapa tanda sumber daya target. Semua sumber daya dari tipe tertentu (instans atau volume) yang memiliki setidaknya satu tanda sumber daya target yang ditentukan akan ditargetkan oleh kebijakan. Misalnya, jika Anda membuat kebijakan snapshot yang menargetkan volume dan Anda menentukan `purpose=prod`, `costcenter=prod`, dan `environment=live` sebagai tanda sumber daya target, kebijakan tersebut akan menargetkan semua volume yang memiliki salah satu pasangan nilai kunci tanda tersebut.

Jika Anda ingin menjalankan beberapa kebijakan pada sumber daya, Anda dapat menetapkan beberapa tanda ke sumber daya target, lalu membuat kebijakan terpisah yang masing-masing menargetkan tanda sumber daya tertentu.

Anda tidak dapat menggunakan karakter \ atau = dalam kunci tanda. Tanda sumber daya peka huruf besar dan kecil. Untuk informasi selengkapnya, lihat Menandai sumber daya Anda.

Snapshot

Snapshot adalah sarana utama untuk mencadangkan data dari volume EBS Anda. Untuk menghemat biaya penyimpanan, snapshot berikutnya bersifat bertahap, hanya berisi data volume yang berubah sejak snapshot sebelumnya. Ketika Anda menghapus satu snapshot dalam seri snapshot untuk volume, hanya data yang unik untuk snapshot itu yang dihapus. Sisa riwayat volume yang ditangkap dipertahankan. Untuk informasi selengkapnya, lihat Snapshot Amazon EBS.

AMI berdukungan EBS

Amazon Machine Image (AMI) menyediakan informasi yang diperlukan untuk meluncurkan sebuah instans. Anda dapat meluncurkan beberapa instans dari AMI tunggal ketika Anda memerlukan beberapa instans dengan konfigurasi yang sama. Amazon Data Lifecycle Manager mendukung AMI

yang didukung EBS saja. AMI yang didukung EBS mencakup snapshot untuk setiap volume EBS yang dipasang pada instans sumber. Untuk informasi selengkapnya, lihat Gambar Mesin Amazon (AMI).

Tanda Amazon Data Lifecycle Manager

Amazon Data Lifecycle Manager menerapkan tanda berikut pada semua snapshot dan AMI yang dibuat oleh kebijakan, untuk membedakannya dari snapshot dan AMI yang dibuat dengan cara lain:

- `aws:dlm:lifecycle-policy-id`
- `aws:dlm:lifecycle-schedule-name`
- `aws:dlm:expirationTime` — Untuk snapshot yang dibuat oleh jadwal berbasis usia. Menunjukkan kapan snapshot akan dihapus dari tingkat standar.
- `dlm:managed`
- `aws:dlm:archived` — Untuk snapshot yang diarsipkan berdasarkan jadwal.
- `aws:dlm:pre-script` — Untuk snapshot yang dibuat dengan skrip pra.
- `aws:dlm:post-script` — Untuk snapshot yang dibuat dengan skrip pasca.

Anda juga dapat menentukan tanda kustom yang akan diterapkan ke snapshot dan AMI pada saat pembuatan. Anda tidak dapat menggunakan karakter \ atau = dalam kunci tanda.

Tanda target yang digunakan Amazon Data Lifecycle Manager untuk mengaitkan volume dengan kebijakan snapshot dapat secara opsional diterapkan pada snapshot yang dibuat oleh kebijakan. Demikian pula, tanda target yang digunakan untuk mengaitkan instans dengan kebijakan AMI dapat secara opsional diterapkan ke AMI yang dibuat oleh kebijakan.

Kebijakan default vs kebijakan kustom

Bagian ini membandingkan kebijakan default dan kebijakan kustom dan menyoroti persamaan dan perbedaannya.

Topik

- Perbandingan kebijakan snapshot EBS
- Perbandingan kebijakan AMI yang didukung EBS

Perbandingan kebijakan snapshot EBS

Tabel berikut menyoroti perbedaan antara kebijakan default untuk snapshot EBS dan kebijakan snapshot EBS kustom.

Fitur	Kebijakan default untuk snapshot EBS	Kebijakan snapshot EBS kustom
Sumber daya cadangan terkelola	Snapshot EBS	Snapshot EBS
Jenis sumber daya target	Volume	Volume atau instans
Penargetan sumber daya	Menargetkan semua volume di Wilayah yang tidak memiliki snapshot terbaru. Anda dapat menentuka n parameter pengecualian untuk mengecualikan instans tertentu.	Menargetkan hanya volume atau instans yang memiliki tanda tertentu.
Parameter pengecualian	Ya, dapat mengecualikan volume boot, tipe volume tertentu, dan volume dengan tanda tertentu.	Ya, dapat mengecualikan volume dan volume boot dengan tanda tertentu saat menargetkan instans.
Support AWS Outposts	Tidak	Ya
Mendukung beberapa jadwal	Tidak	Ya, hingga 4 jadwal per kebijakan
Jenis retensi yang didukung	Retensi berbasis usia saja	Retensi berbasis usia dan berbasis hitungan
Frekuensi pembuatan snapshot	Setiap 1 hingga 7 hari.	Frekuensi harian, mingguan, bulanan, tahunan, atau kustom menggunakan ekspresi cron.

Fitur	Kebijakan default untuk snapshot EBS	Kebijakan snapshot EBS kustom
Retensi snapshot	2 hingga 14 hari.	Hingga 1000 snapshot (berbasis hitungan) atau hingga 100 tahun (berbasis usia).
Mendukung snapshot yang konsisten dengan aplikasi	Tidak	Ya, menggunakan skrip pra dan pasca
Mendukung pengarsipan snapshot	Tidak	Ya
Mendukung pemulihan snapshot cepat	Tidak	Ya
Mendukung penyalinan lintas Wilayah	Ya, dengan pengaturan default [1]	Ya, dengan pengaturan kustom
Mendukung berbagi lintas akun	Tidak	Ya
Mendukung penghapusan yang diperpanj ang [2]	Ya	Tidak

[1] Untuk kebijakan default:

- Anda tidak dapat menyalin tanda ke salinan lintas wilayah.
- Salinan menggunakan periode retensi yang sama dengan snapshot sumber.

- Salinan mendapatkan status enkripsi yang sama dengan snapshot sumber. Jika Wilayah tujuan diaktifkan untuk enkripsi secara default, salinan selalu dienkripsi, bahkan jika snapshot sumber tidak dienkripsi. Salinan selalu dienkripsi dengan kunci KMS default untuk Wilayah tujuan.

[2] Untuk kebijakan default dan kustom:

- Jika instans atau volume target dihapus, Amazon Data Lifecycle Manager terus menghapus snapshot hingga, tetapi tidak termasuk, snapshot terakhir berdasarkan periode retensi. Untuk kebijakan default, Anda dapat memperpanjang penghapusan untuk menyertakan snapshot terakhir.
- Jika kebijakan dihapus atau memasukkan kesalahan atau status dinonaktifkan, Amazon Data Lifecycle Manager berhenti menghapus snapshot. Untuk kebijakan default, Anda dapat memperpanjang penghapusan terus menghapus snapshot, termasuk snapshot terakhir.

Perbandingan kebijakan AMI yang didukung EBS

Tabel berikut menyoroti perbedaan antara kebijakan default untuk snapshot EBS dan kebijakan snapshot EBS kustom.

Fitur	Kebijakan default untuk AMI yang didukung EBS	Kebijakan AMI yang didukung EBS
Sumber daya cadangan terkelola	AMI berdukungan EBS	AMI berdukungan EBS
Jenis sumber daya target	Instans	Instans
Penargetan sumber daya	Menargetkan semua instans di Wilayah yang tidak memiliki AMI terbaru. Anda dapat menentuka n parameter pengecualian untuk mengecualikan instans tertentu.	Menargetkan hanya instans yang memiliki tanda tertentu.

Fitur	Kebijakan default untuk AMI yang didukung EBS	Kebijakan AMI yang didukung EBS
Boot ulang instans sebelum pembuatan AMI	Tidak	Ya
Parameter pengecualian	Ya, dapat mengecualikan instans dengan tanda tertentu.	Tidak
Mendukung beberapa jadwal	Tidak	Ya, hingga 4 jadwal per kebijakan.
Frekuensi pembuatan AMI	Setiap 1 hingga 7 hari.	Frekuensi harian, mingguan, bulanan, tahunan, atau kustom menggunakan ekspresi cron.
Jenis retensi yang didukung	Retensi berbasis usia saja.	Retensi berbasis usia dan berbasis jumlah.
Retensi AMI	2 hingga 14 hari.	Hingga 1000 snapshot (berbasis hitungan) atau hingga 100 tahun (berbasis usia).
Mendukung AMI penghentian	Tidak	Ya
Mendukung penyalinan lintas Wilayah	Ya, dengan pengaturan default [1]	Ya, dengan pengaturan kustom
Mendukung penghapusan yang diperpanj ang [2]	Ya	Tidak

[1]Untuk kebijakan default:

- Anda tidak dapat menyalin tanda ke salinan lintas wilayah.

- Salinan menggunakan periode retensi yang sama dengan AMI sumber.
- Salinan mendapatkan status enkripsi yang sama dengan AMI sumber. Jika Wilayah tujuan diaktifkan untuk enkripsi secara default, salinan selalu dienkripsi, bahkan jika AMI sumber tidak dienkripsi. Salinan selalu dienkripsi dengan kunci KMS default untuk Wilayah tujuan.

[2] Untuk kebijakan default dan kustom:

- Jika instans sumber diakhiri, Amazon Data Lifecycle Manager terus membatalkan pendaftaran AMI yang dibuat sebelumnya hingga, tetapi tidak termasuk, yang terakhir berdasarkan periode retensi. Untuk kebijakan default, Anda dapat memperpanjang pembatalan pendaftaran untuk menyertakan AMI terakhir.
- Jika kebijakan dihapus atau memasuki status kesalahan atau dinonaktifkan, Amazon Data Lifecycle Manager berhenti membatalkan pendaftaran AMI. Untuk kebijakan default, Anda dapat memperpanjang pembatalan pendaftaran AMI, termasuk AMI terakhir.

Kebijakan default

Untuk membuat AMI yang didukung EBS berkala dari instans, gunakan kebijakan default untuk AMI yang didukung EBS. Untuk membuat snapshot dari semua volume terlepas dari status lampirannya, atau jika Anda ingin mengecualikan volume tertentu, gunakan kebijakan default untuk snapshot EBS.

Bagian ini menjelaskan cara membuat kebijakan default.

Topik

Pertimbangan

Ingatlah hal-hal berikut ini saat bekerja dengan kebijakan default:

- Kebijakan default tidak mencadangkan sumber daya target (instans atau volume) yang memiliki cadangan terbaru (snapshot atau AMI). Frekuensi pembuatan menentukan sumber daya yang

dicadangkan. Volume atau instans dicadangkan hanya jika snapshot atau AMI terakhirnya lebih tua dari frekuensi pembuatan kebijakan. Misalnya, jika Anda menentukan frekuensi pembuatan 3 hari, kebijakan default untuk snapshot EBS akan membuat snapshot volume hanya jika snapshot terakhirnya lebih tua dari 3 hari.

- Secara default, kebijakan default menargetkan semua instans atau volume di Wilayah, kecuali parameter pengecualian ditentukan.
- Kebijakan default akan membuat set snapshot unik minimum. Misalnya, jika Anda mengaktifkan kebijakan AMI yang didukung EBS dan kebijakan snapshot EBS, kebijakan snapshot tidak akan menduplikasi snapshot volume yang sudah didukung oleh kebijakan AMI yang didukung EBS.
- Kebijakan default hanya akan mulai menargetkan sumber daya yang berusia minimal 24 jam.
- Jika Anda menghapus volume atau mengakhiri instans yang ditargetkan oleh kebijakan default, Amazon Data Lifecycle Manager akan terus menghapus cadangan (snapshot atau AMI) yang dibuat sebelumnya sesuai dengan periode retensi hingga, tetapi tidak termasuk, cadangan terakhir. Anda harus menghapus cadangan ini secara manual jika tidak diperlukan.

  Jika ingin Amazon Data Lifecycle Manager menghapus cadangan terakhir, Anda dapat mengaktifkan perpanjang penghapusan.
- Jika kebijakan dihapus atau memasuki status kesalahan atau dinonaktifkan, Amazon Data Lifecycle Manager berhenti menghapus cadangan (snapshot atau AMI) yang dibuat sebelumnya. Jika ingin Amazon Data Lifecycle Manager terus menghapus cadangan, termasuk yang terakhir, Anda harus mengaktifkan perpanjang penghapusan sebelum menghapus kebijakan atau sebelum status kebijakan berubah menjadi dinonaktifkan atau dihapus.
- Saat Anda membuat dan mengaktifkan kebijakan default, Amazon Data Lifecycle Manager secara acak menetapkan sumber daya yang ditargetkan ke jendela waktu empat jam. Sumber daya yang ditargetkan dicadangkan selama jendela yang ditetapkan pada frekuensi pembuatan yang ditentukan. Misalnya, jika kebijakan memiliki frekuensi pembuatan 3 hari, dan sumber daya target ditetapkan ke jendela 12:00 - 16:00, sumber daya tersebut akan dicadangkan antara pukul 12:00 - 16:00 setiap 3 hari.

Kebijakan default untuk snapshot EBS

Prosedur berikut ini menunjukkan cara membuat kebijakan default untuk snapshot EBS.

Console

Untuk membuat kebijakan default untuk snapshot EBS

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Lifecycle Manager lalu pilih Buat kebijakan siklus hidup.
3. Untuk Jenis kebijakan, pilih Kebijakan default, lalu pilih Kebijakan snapshot EBS.
4. Untuk Deskripsi, masukkan deskripsi singkat untuk kebijakan tersebut.
5. Untuk Peran IAM, pilih peran IAM yang memiliki izin untuk mengelola snapshot.

   Sebaiknya pilih Default untuk menggunakan peran IAM default yang disediakan oleh Amazon Data Lifecycle Manager. Namun, Anda juga dapat menggunakan peran IAM kustom yang sebelumnya Anda buat.
6. Untuk Frekuensi Pembuatan, tentukan seberapa sering Anda ingin kebijakan berjalan dan membuat snapshot volume Anda.

   Frekuensi yang Anda tentukan juga menentukan volume yang dicadangkan. Kebijakan hanya akan mencadangkan volume yang belum dicadangkan oleh cara lain dalam frekuensi yang ditentukan. Misalnya, jika Anda menentukan frekuensi pembuatan 3 hari, kebijakan hanya akan membuat snapshot volume yang belum dicadangkan dalam 3 hari terakhir.
7. Untuk Periode retensi, tentukan berapa lama Anda ingin kebijakan mempertahankan snapshot yang dibuatnya. Ketika snapshot mencapai ambang retensi, snapshot akan dihapus secara otomatis. Periode retensi harus lebih besar dari atau sama dengan frekuensi pembuatan.
8. (Opsional) Konfigurasikan Parameter pengecualian untuk mengecualikan volume tertentu dari cadangan terjadwal. Volume yang dikecualikan tidak akan dicadangkan saat kebijakan berjalan.

   a. Untuk mengecualikan volume boot, pilih Kecualikan volume boot. Jika Anda mengecualikan volume boot, hanya volume data (non-boot) yang akan dicadangkan oleh kebijakan. Dengan kata lain, kebijakan tidak akan membuat snapshot volume yang dilampirkan ke instans sebagai volume boot.

   b. Untuk mengecualikan tipe volume tertentu, pilih Kecualikan tipe volume tertentu, lalu pilih tipe volume yang akan dikecualikan. Hanya volume dari tipe yang tersisa yang akan didukung oleh kebijakan.

c. Untuk mengecualikan volume yang memiliki tanda tertentu, pilih Tambahkan tanda, lalu tentukan kunci dan nilai tanda. Kebijakan tidak akan membuat snapshot volume yang memiliki tanda yang ditentukan.

9. (Opsional) Di Pengaturan lanjutan, tentukan tindakan tambahan yang harus dilakukan kebijakan.

   a. Untuk menyalin tanda yang ditetapkan dari volume sumber ke snapshot, pilih Salin tanda dari volume.

   b. Dengan Perpanjang penghapusan dinonaktifkan:

      - Jika instans atau volume target dihapus, Amazon Data Lifecycle Manager terus menghapus snapshot hingga, tetapi tidak termasuk, snapshot terakhir berdasarkan periode penyimpanan. Jika Anda ingin Amazon Data Lifecycle Manager menghapus semua snapshot, termasuk yang terakhir, pilih Perpanjang penghapusan.
      - Jika kebijakan dihapus atau memasuki status `error` atau `disabled`, Amazon Data Lifecycle Manager berhenti menghapus snapshot. Jika Anda ingin Amazon Data Lifecycle Manager agar terus menghapus semua snapshot, termasuk yang terakhir, pilih Perpanjang penghapusan.

      > **Note**
      >
      > Jika Anda mengaktifkan perpanjang penghapusan, Anda menimpa kedua perilaku yang dijelaskan di atas secara bersamaan.

   c. Untuk menyalin snapshot yang dibuat oleh kebijakan ke Wilayah lain, pilih Buat salinan lintas-Wilayah, lalu pilih hingga 3 Wilayah tujuan.

      - Jika snapshot sumber dienkripsi, atau jika enkripsi secara default diaktifkan untuk Wilayah tujuan, snapshot yang disalin dienkripsi menggunakan kunci KMS default untuk enkripsi EBS di Wilayah tujuan.
      - Jika snapshot sumber tidak dienkripsi dan enkripsi secara default dinonaktifkan untuk Wilayah tujuan, snapshot yang disalin tidak dienkripsi.

10. (Opsional) Untuk menambahkan tanda ke kebijakan, pilih Tambahkan tanda lalu tentukan kunci tanda dan pasangan nilai.

11. Pilih Buat kebijakan default.

> **Note**
>
> Jika Anda mendapatkan kesalahan `Role with name AWSDataLifecycleManagerDefaultRole already exists`, lihat Pemecahan Masalah untuk informasi selengkapnya.

AWS CLI

Untuk membuat kebijakan default untuk snapshot EBS

Gunakan perintah create-lifecycle-policy. Anda dapat menentukan parameter permintaan dalam salah satu dari dua metode, bergantung pada kasus penggunaan atau preferensi Anda:

- Metode 1

```
$ aws dlm create-lifecycle-policy \
--state ENABLED | DISABLED \
--description "policy_description" \
--execution-role-arn role_arn \
--default-policy VOLUME \
--create-interval creation_frequency_in_days (1-7) \
--retain-interval retention_period_in_days (2-14) \
--copy-tags | --no-copy-tags \
--extend-deletion | --no-extend-deletion \
--cross-region-copy-targets TargetRegion=destination_region_code \
--exclusions ExcludeBootVolumes=true | false,
 ExcludeTags=[{Key=tag_key,Value=tag_value}], ExcludeVolumeTypes="standard | gp2 |
 gp3 | io1 | io2 | st1 | sc1"
```

Misalnya, untuk membuat kebijakan default untuk snapshot EBS yang menargetkan semua volume di Wilayah, menggunakan peran IAM default, berjalan harian (default), dan mempertahankan snapshot selama 7 hari (default), Anda perlu menentukan parameter berikut:

```
$ aws dlm create-lifecycle-policy \
--state ENABLED \
--description "Daily default snapshot policy" \
--execution-role-arn arn:aws:iam::account_id:role/
AWSDataLifecycleManagerDefaultRole \
--default-policy VOLUME
```

- Metode 2

```
$ aws dlm create-lifecycle-policy \
--state ENABLED | DISABLED \
--description "policy_description" \
--execution-role-arn role_arn \
--default-policy VOLUME \
--policy-details file://policyDetails.json
```

Jika `policyDetails.json` mencakup berikut:

```
{
    "PolicyLanguage": "SIMPLIFIED",
    "PolicyType": "EBS_SNAPSHOT_MANAGEMENT",
    "ResourceType": "VOLUME",
    "CopyTags": true | false,
    "CreateInterval": creation_frequency_in_days (1-7),
    "RetainInterval": retention_period_in_days (2-14),
    "ExtendDeletion": true | false,
    "CrossRegionCopyTargets": [{"TargetRegion":"destination_region_code"}],
    "Exclusions": {
        "ExcludeBootVolume": true | false,
  "ExcludeVolumeTypes": ["standard | gp2 | gp3 | io1 | io2 | st1 | sc1"],
        "ExcludeTags": [{
            "Key": "exclusion_tag_key",
            "Value": "exclusion_tag_value"
        }]
    }
}
```

Kebijakan default untuk AMI yang didukung EBS

Prosedur berikut ini menunjukkan cara membuat kebijakan default untuk AMI yang didukung EBS.

Console

Untuk membuat kebijakan default untuk AMI yang didukung EBS

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Lifecycle Manager lalu pilih Buat kebijakan siklus hidup.

3. Untuk Jenis Kebijakan, pilih Kebijakan default, lalu pilih kebijakan AMI yang didukung EBS.
4. Untuk Deskripsi, masukkan deskripsi singkat untuk kebijakan tersebut.
5. Untuk peran IAM, pilih peran IAM yang memiliki izin untuk mengelola AMI.

   Sebaiknya pilih Default untuk menggunakan peran IAM default yang disediakan oleh Amazon Data Lifecycle Manager. Namun, Anda juga dapat menggunakan peran IAM kustom yang sebelumnya Anda buat.
6. Untuk Frekuensi pembuatan, tentukan seberapa sering Anda ingin kebijakan berjalan dan membuat AMI dari instans Anda.

   Frekuensi yang Anda tentukan juga menentukan instans yang dicadangkan. Kebijakan hanya akan mencadangkan instans yang belum dicadangkan oleh cara lain dalam frekuensi yang ditentukan. Misalnya, jika Anda menentukan frekuensi pembuatan 3 hari, kebijakan hanya akan membuat AMI dari instans yang belum dicadangkan dalam 3 hari terakhir.
7. Untuk Periode retensi, tentukan berapa lama Anda ingin kebijakan mempertahankan AMI yang dibuatnya. Ketika AMI mencapai ambang batas retensi, AMI akan secara otomatis dibatalkan pendaftarannya dan snapshot yang terkait akan dihapus. Periode retensi harus lebih besar dari atau sama dengan frekuensi pembuatan.
8. (Opsional) Konfigurasikan Parameter pengecualian untuk mengecualikan instans tertentu dari cadangan terjadwal. Instans yang dikecualikan tidak akan dicadangkan saat kebijakan berjalan.

   - Untuk mengecualikan instans yang memiliki tanda tertentu, pilih Tambahkan tanda, lalu tentukan kunci dan nilai tanda. Kebijakan tidak akan membuat AMI dari instans yang memiliki tanda yang ditentukan.
9. (Opsional) Di Pengaturan lanjutan, tentukan tindakan tambahan yang harus dilakukan kebijakan.

   a. Untuk menyalin tanda yang ditetapkan dari instans sumber ke AMI, pilih Salin tanda dari instans.
   b. Dengan Perpanjang penghapusan dinonaktifkan:

      - Jika instans sumber diakhiri, Amazon Data Lifecycle Manager terus membatalkan pendaftaran AMI yang dibuat sebelumnya hingga, tetapi tidak termasuk, yang terakhir berdasarkan periode retensi. Jika Anda ingin Amazon Data Lifecycle Manager membatalkan pendaftaran semua AMI, termasuk yang terakhir, pilih Perpanjang penghapusan.

- Jika kebijakan dihapus atau memasuki status `error` atau `disabled`, Amazon Data Lifecycle Manager berhenti membatalkan pendaftaran AMI. Jika Anda ingin Amazon Data Lifecycle Manager terus membatalkan pendaftaran semua AMI, termasuk yang terakhir, pilih Perpanjang penghapusan.

> **Note**
>
> Jika Anda mengaktifkan perpanjangan penghapusan, Anda menimpa kedua perilaku yang dijelaskan di atas secara bersamaan.

c. Untuk menyalin AMI yang dibuat oleh kebijakan ke Wilayah lain, pilih Buat salinan lintas-Wilayah, lalu pilih hingga 3 Wilayah tujuan.

- Jika AMI sumber dienkripsi, atau jika enkripsi secara default diaktifkan untuk Wilayah tujuan, AMI yang disalin dienkripsi menggunakan kunci KMS default untuk enkripsi EBS di Wilayah tujuan.
- Jika AMI sumber tidak dienkripsi dan enkripsi secara default dinonaktifkan untuk Wilayah tujuan, AMI yang disalin tidak dienkripsi.

10. (Opsional) Untuk menambahkan tanda ke kebijakan, pilih Tambahkan tanda lalu tentukan kunci tanda dan pasangan nilai.
11. Pilih Buat kebijakan default.

> **Note**
>
> Jika Anda mendapatkan kesalahan `Role with name AWSDataLifecycleManagerDefaultRoleForAMIManagement already exists`, lihat Pemecahan Masalah untuk informasi selengkapnya.

AWS CLI

Untuk membuat kebijakan default untuk AMI yang didukung EBS

Gunakan perintah create-lifecycle-policy. Anda dapat menentukan parameter permintaan dalam salah satu dari dua metode, bergantung pada kasus penggunaan atau preferensi Anda:

- Metode 1

```
$ aws dlm create-lifecycle-policy \
--state ENABLED | DISABLED \
--description "policy_description" \
--execution-role-arn role_arn \
--default-policy INSTANCE \
--create-interval creation_frequency_in_days (1-7) \
--retain-interval retention_period_in_days (2-14) \
--copy-tags | --no-copy-tags \
--extend-deletion | --no-extend-deletion \
--cross-region-copy-targets TargetRegion=destination_region_code \
--exclusions ExcludeTags=[{Key=tag_key,Value=tag_value}]
```

Misalnya, untuk membuat kebijakan default untuk AMI yang didukung EBS yang menargetkan semua instans di Wilayah, menggunakan peran IAM default, berjalan harian (default), dan mempertahankan AMI selama 7 hari (default), Anda perlu menentukan parameter berikut:

```
$ aws dlm create-lifecycle-policy \
--state ENABLED \
--description "Daily default AMI policy" \
--execution-role-arn arn:aws:iam::account_id:role/
AWSDataLifecycleManagerDefaultRoleForAMIManagement \
--default-policy INSTANCE
```

- Metode 2

```
$ aws dlm create-lifecycle-policy \
--state ENABLED | DISABLED \
--description "policy_description" \
--execution-role-arn role_arn \
--default-policy INSTANCE \
--policy-details file://policyDetails.json
```

Jika `policyDetails.json` mencakup berikut:

```
{
    "PolicyLanguage": "SIMPLIFIED",
    "PolicyType": "IMAGE_MANAGEMENT",
    "ResourceType": "INSTANCE",
    "CopyTags": true | false,
    "CreateInterval": creation_frequency_in_days (1-7),
```

```
    "RetainInterval": retention_period_in_days (2-14),
    "ExtendDeletion": true | false,
 "CrossRegionCopyTargets": [{"TargetRegion":"destination_region_code"}],
    "Exclusions": {
        "ExcludeTags": [{
            "Key": "exclusion_tag_key",
            "Value": "exclusion_tag_value"
        }]
    }
}
```

Aktifkan kebijakan default di seluruh akun dan Wilayah

Dengan menggunakan AWS CloudFormation StackSets, Anda dapat mengaktifkan kebijakan default Amazon Data Lifecycle Manager di beberapa akun dan AWS Wilayah dengan satu operasi.

Anda dapat menggunakan kumpulan tumpukan untuk mengaktifkan kebijakan default dengan salah satu cara berikut:

- Di seluruh AWS organisasi — Memastikan bahwa kebijakan default diaktifkan dan dikonfigurasi secara konsisten di seluruh AWS organisasi atau unit organisasi tertentu dalam suatu organisasi. Ini dilakukan dengan menggunakan izin yang dikelola layanan. AWS CloudFormation StackSets membuat peran IAM yang diperlukan atas nama Anda.
- Di seluruh AWS akun tertentu — Memastikan bahwa kebijakan default diaktifkan dan dikonfigurasi secara konsisten di seluruh akun target tertentu. Ini memerlukan izin yang dikelola sendiri. Anda membuat peran IAM yang diperlukan untuk membangun hubungan kepercayaan antara akun administrator set tumpukan dan akun target.

Untuk informasi selengkapnya, lihat Model izin untuk kumpulan tumpukan di Panduan AWS CloudFormation Pengguna.

Gunakan prosedur berikut untuk mengaktifkan kebijakan default Amazon Data Lifecycle Manager di seluruh AWS organisasi, di seluruh OU tertentu, atau di seluruh akun target tertentu.

Prasyarat

Lakukan salah satu hal berikut, tergantung pada cara Anda mengaktifkan kebijakan default:

- (Di seluruh AWS organisasi) Anda harus mengaktifkan semua fitur di organisasi Anda dan mengaktifkan akses tepercaya AWS Organizations. Anda juga harus menggunakan akun manajemen organisasi atau akun administrator yang didelegasikan.
- (Di seluruh akun target tertentu) Anda harus memberikan izin yang dikelola sendiri dengan membuat peran yang diperlukan untuk membangun hubungan tepercaya antara akun administrator set stack dan akun target.

Console

Untuk mengaktifkan kebijakan default di seluruh AWS organisasi atau di seluruh akun target tertentu

1. Buka AWS CloudFormation konsol di https://console.aws.amazon.com/cloudformation.
2. Di panel navigasi, pilih StackSets, lalu pilih Buat StackSet.
3. Untuk Izin, lakukan salah satu hal berikut, tergantung pada cara Anda mengaktifkan kebijakan default:
   - (Di seluruh AWS organisasi) Pilih Izin yang dikelola layanan.
   - (Di seluruh akun target tertentu) Pilih Izin layanan mandiri. Kemudian, untuk peran admin IAM ARN, pilih peran layanan IAM yang Anda buat untuk akun administrator, dan untuk nama peran eksekusi IAM, masukkan nama peran layanan IAM yang Anda buat di akun target.
4. Untuk Siapkan template, pilih Gunakan contoh template.
5. Untuk contoh template, lakukan salah satu hal berikut:
   - (Kebijakan default untuk snapshot EBS) Pilih Buat kebijakan default Amazon Data Lifecycle Manager untuk EBS Snapshots.
   - (Kebijakan default untuk AMI yang didukung EBS) Pilih Buat kebijakan default Amazon Data Lifecycle Manager untuk AMI yang didukung EBS.
6. Pilih Selanjutnya.
7. Untuk StackSet nama dan StackSet deskripsi, masukkan nama deskriptif dan deskripsi singkat.
8. Di bagian Parameter, konfigurasikan pengaturan kebijakan default sesuai kebutuhan.

> **Note**
>
> Untuk beban kerja kritis, kami sarankan CreateInterval = 1 hari dan RetainInterval = 7 hari.

9. Pilih Selanjutnya.
10. (Opsional) Untuk Tag, tentukan tag untuk membantu Anda mengidentifikasi StackSet dan menumpuk sumber daya.
11. Untuk eksekusi Terkelola, pilih Aktif.
12. Pilih Selanjutnya.
13. Untuk Menambahkan stack ke set stack, pilih Terapkan stack baru.
14. Lakukan salah satu hal berikut, tergantung pada cara Anda mengaktifkan kebijakan default:
    - (Di seluruh AWS organisasi) Untuk target Deployment pilih salah satu opsi berikut:
      - Untuk menyebarkan di seluruh AWS organisasi, pilih Terapkan ke organisasi.
      - Untuk menyebarkan ke unit organisasi tertentu (OU), pilih Menyebarkan ke unit organisasi, dan kemudian untuk ID OU, masukkan ID OU. Untuk menambahkan OU tambahan, pilih Tambahkan OU lain.
    - (Di seluruh akun target tertentu) Untuk Akun, lakukan salah satu hal berikut:
      - Untuk menyebarkan ke akun target tertentu, pilih Menyebarkan tumpukan di akun, lalu untuk nomor Akun, masukkan ID akun target.
      - Untuk menyebarkan ke semua akun di OU tertentu, pilih Menyebarkan tumpukan ke semua akun di unit organisasi, lalu untuk nomor Organisasi, masukkan ID OU target.
15. Untuk penyebaran otomatis, pilih Diaktifkan.
16. Untuk perilaku penghapusan akun, pilih Pertahankan tumpukan.
17. Untuk Menentukan wilayah, pilih Wilayah tertentu untuk mengaktifkan kebijakan default, atau pilih Tambahkan semua Wilayah untuk mengaktifkan kebijakan default di semua Wilayah.
18. Pilih Selanjutnya.
19. Tinjau pengaturan set tumpukan, pilih Saya mengakui yang AWS CloudFormation mungkin membuat sumber daya IAM, lalu pilih Kirim.

AWS CLI

Untuk mengaktifkan kebijakan default di seluruh AWS organisasi

1. Buat set tumpukan. Gunakan perintah create-stack-set.

   Untuk `--permission-model`, tentukan `SERVICE_MANAGED`.

   Untuk`--template-url`, tentukan salah satu URL template berikut:

   - (Kebijakan default untuk AMI yang didukung EBS) `https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/DataLifecycleManagerAMIDefaultPolicy.yaml`
   - (Kebijakan default untuk snapshot EBS) `https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/DataLifecycleManagerEBSSnapshotDefaultPolicy.yaml`

   Untuk`--parameters`, tentukan pengaturan untuk kebijakan default. Untuk parameter yang didukung, deskripsi parameter, dan nilai yang valid, unduh templat menggunakan URL dan kemudian lihat templat menggunakan editor teks.

   Untuk `--auto-deployment`, tentukan `Enabled=true,RetainStacksOnAccountRemoval=true`.

   ```
   $ aws cloudformation create-stack-set \
   --stack-set-name stackset_name \
   --permission-model SERVICE_MANAGED \
   --template-url template_url \
   --parameters "ParameterKey=param_name_1,ParameterValue=param_value_1"
    "ParameterKey=param_name_2,ParameterValue=param_value_2" \
   --auto-deployment "Enabled=true, RetainStacksOnAccountRemoval=true"
   ```

2. Menyebarkan set tumpukan. Gunakan perintah create-stack-instance.

   Untuk`--stack-set-name`, tentukan nama kumpulan tumpukan yang Anda buat pada langkah sebelumnya.

   Untuk`--deployment-targets OrganizationalUnitIds`, tentukan ID root OU yang akan diterapkan ke seluruh organisasi, atau ID OU untuk diterapkan ke OU tertentu di organisasi.

Untuk `--regions`, tentukan AWS Wilayah untuk mengaktifkan kebijakan default.

```
$ aws cloudformation create-stack-instances \
--stack-set-name stackset_name \
--deployment-targets OrganizationalUnitIds='["root_ou_id"]' | '["ou_id_1",
 "ou_id_2]' \
--regions '["region_1", "region_2"]'
```

Untuk mengaktifkan kebijakan default di seluruh akun target tertentu

1. Buat set tumpukan. Gunakan perintah create-stack-set.

   Untuk `--template-url`, tentukan salah satu URL template berikut:

   - (Kebijakan default untuk AMI yang didukung EBS) `https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/DataLifecycleManagerAMIDefaultPolicy.yaml`
   - (Kebijakan default untuk snapshot EBS) `https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/DataLifecycleManagerEBSSnapshotDefaultPolicy.yaml`

   Untuk `--administration-role-arn`, tentukan ARN dari peran layanan IAM yang sebelumnya Anda buat untuk administrator kumpulan tumpukan.

   Untuk `--execution-role-name`, tentukan nama peran layanan IAM yang Anda buat di akun target.

   Untuk `--parameters`, tentukan pengaturan untuk kebijakan default. Untuk parameter yang didukung, deskripsi parameter, dan nilai yang valid, unduh templat menggunakan URL dan kemudian lihat templat menggunakan editor teks.

   Untuk `--auto-deployment`, tentukan `Enabled=true,RetainStacksOnAccountRemoval=true`.

```
$ aws cloudformation create-stack-set \
--stack-set-name stackset_name \
--template-url template_url \
```

```
--parameters "ParameterKey=param_name_1,ParameterValue=param_value_1"
 "ParameterKey=param_name_2,ParameterValue=param_value_2" \
--administration-role-arn administrator_role_arn \
--execution-role-name target_account_role \
--auto-deployment "Enabled=true, RetainStacksOnAccountRemoval=true"
```

2. Menyebarkan set tumpukan. Gunakan perintah create-stack-instance.

   Untuk `--stack-set-name`, tentukan nama kumpulan tumpukan yang Anda buat pada langkah sebelumnya.

   Untuk `--accounts`, tentukan ID AWS akun target.

   Untuk `--regions`, tentukan AWS Wilayah untuk mengaktifkan kebijakan default.

```
$ aws cloudformation create-stack-instances \
--stack-set-name stackset_name \
--accounts '["account_ID_1","account_ID_2"]' \
--regions '["region_1", "region_2"]'
```

Kebijakan kustom

Bagian ini menjelaskan cara membuat snapshot EBS kustom, AMI yang didukung EBS, dan kebijakan peristiwa salinan lintas akun.

Topik

- Mengotomatiskan siklus hidup snapshot
- Mengotomatiskan siklus hidup AMI
- Mengotomatiskan salinan snapshot lintas akun

Mengotomatiskan siklus hidup snapshot

Prosedur berikut ini menunjukkan cara menggunakan Amazon Data Lifecycle Manager untuk mengotomatisasi siklus hidup snapshot Amazon EBS.

Topik

- Membuat kebijakan siklus hidup snapshot
- Pertimbangan untuk kebijakan siklus hidup snapshot

- Sumber daya tambahan
- Persyaratan untuk menggunakan skrip pra dan pasca
- Mengotomatiskan snapshot yang konsisten dengan aplikasi dengan skrip pra dan pasca
- Kasus penggunaan lain untuk skrip pra dan pasca
- Cara kerja skrip pra dan pasca
- Mengidentifikasi snapshot yang dibuat dengan skrip pra dan pasca
- Memantau eksekusi skrip pra dan pasca

Membuat kebijakan siklus hidup snapshot

Gunakan salah satu prosedur berikut ini untuk membuat kebijakan siklus hidup snapshot.

Console

Untuk membuat kebijakan snapshot

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Elastic Block Store, Lifecycle Manager, lalu pilih Buat kebijakan siklus hidup.
3. Pada layar Pilih jenis kebijakan, pilih Kebijakan snapshot EBS, lalu pilih Berikutnya.
4. Di bagian Sumber daya target, lakukan hal berikut ini:

   a. Untuk Jenis sumber daya target, pilih jenis sumber daya untuk pencadangan. Pilih `Volume` untuk membuat snapshot volume individu atau pilih `Instance` untuk membuat snapshot multi-volume dari volume yang dilampirkan ke suatu instans.

   b. (Hanya pelanggan AWS pos terdepan) Tentukan di mana sumber daya target berada.

      Untuk Lokasi sumber daya target, tentukan lokasi sumber daya target.

      - Jika sumber daya target berada di AWS Wilayah, pilih AWS Wilayah. Amazon Data Lifecycle Manager mencadangkan semua sumber daya dari jenis yang ditentukan yang memiliki tanda target yang cocok di Wilayah saat ini saja. Jika sumber daya terletak di Wilayah, snapshot yang dibuat oleh kebijakan akan disimpan di Wilayah yang sama.
      - Jika sumber daya target terletak di Outposts di akun Anda, pilih AWS Outposts. Amazon Data Lifecycle Manager mencadangkan semua sumber daya dari jenis yang

ditentukan yang memiliki tanda target yang cocok di seluruh Outposts di akun Anda. Jika sumber daya terletak di Outposts, snapshot yang dibuat oleh kebijakan dapat disimpan di Wilayah yang sama atau pada Outposts yang sama sebagai sumber daya.

- Jika Anda tidak memiliki Outposts di akun Anda, opsi ini disembunyikan dan AWS Wilayah dipilih untuk Anda.

c. Untuk Tanda sumber daya target, pilih tanda sumber daya yang mengidentifikasi volume atau instans yang akan dicadangkan. Hanya sumber daya yang memiliki pasangan kunci tanda dan nilai yang ditentukan yang dicadangkan oleh kebijakan.

5. Untuk Deskripsi, masukkan deskripsi singkat untuk kebijakan tersebut.
6. Untuk peran IAM, pilih peran IAM yang memiliki izin untuk mengelola snapshot dan untuk mendeskripsikan volume serta instans. Untuk menggunakan peran default yang disediakan oleh Amazon Data Lifecycle Manager, pilih Peran default. Atau, untuk menggunakan peran IAM kustom yang Anda buat sebelumnya, pilih Pilih peran lain, lalu pilih peran yang akan digunakan.
7. Untuk Tanda kebijakan, tambahkan tanda yang akan diterapkan pada kebijakan siklus hidup. Anda dapat menggunakan tanda ini untuk mengidentifikasi dan mengategorikan kebijakan Anda.
8. Untuk Status kebijakan, pilih Aktifkan untuk memulai pelaksanaan kebijakan pada waktu yang dijadwalkan berikutnya, atau Nonaktifkan kebijakan untuk mencegah agar kebijakan tidak berjalan. Jika Anda tidak mengaktifkan kebijakan sekarang, kebijakan tidak akan mulai membuat snapshot sampai Anda mengaktifkannya secara manual setelah pembuatan.
9. (Kebijakan yang hanya menargetkan instans) Kecualikan volume dari set snapshot multi-volume.

   Secara default, Amazon Data Lifecycle Manager akan membuat snapshot dari semua volume yang terlampir ke instans yang ditargetkan. Namun, Anda dapat memilih untuk membuat snapshot dari subset volume yang dilampirkan. Di bagian Parameter, lakukan hal berikut ini:

   - Jika Anda tidak ingin membuat snapshot dari volume root yang dilampirkan ke instans yang ditargetkan, pilih Kecualikan volume root. Jika Anda memilih opsi ini, hanya volume data (non-root) yang dilampirkan ke instans yang ditargetkan yang akan disertakan dalam set snapshot multi-volume.
   - Jika Anda ingin membuat snapshot dari subset volume data (non-root) yang dilampirkan ke instans yang ditargetkan, pilih Kecualikan volume data tertentu, lalu tentukan tanda yang akan digunakan untuk mengidentifikasi volume data yang tidak boleh dibuat snapshot. Amazon Data Lifecycle Manager tidak akan membuat snapshot volume data yang memiliki

tanda yang ditentukan. Amazon Data Lifecycle Manager hanya akan membuat snapshot dari volume data yang tidak memiliki tanda yang ditentukan.

10. Pilih Berikutnya.
11. Pada layar Konfigurasikan jadwal, konfigurasikan jadwal kebijakan. Kebijakan dapat memiliki hingga 4 jadwal. Jadwal 1 bersifat wajib. Jadwal 2, 3, dan 4 bersifat opsional. Untuk setiap jadwal kebijakan yang Anda tambahkan, lakukan hal berikut:

    a. Dalam bagian Detail jadwal, lakukan hal berikut:

        i. Untuk Nama jadwal, tentukan nama deskriptif untuk jadwal.

        ii. Untuk Frekuensi dan bidang terkait, konfigurasikan interval antara kebijakan yang dijalankan.

        Anda dapat mengonfigurasikan kebijakan yang berjalan sesuai jadwal harian, mingguan, bulanan, atau tahunan. Atau, pilih Ekspresi cron kustom untuk menentukan interval hingga satu tahun. Untuk informasi selengkapnya, lihat Ekspresi cron di Panduan Pengguna CloudWatch Acara Amazon.

        > **Note**
        >
        > Jika Anda perlu mengaktifkan pengarsipan snapshot untuk jadwal, Anda harus memilih frekuensi bulanan atau tahunan, atau Anda harus menentukan ekspresi cron dengan frekuensi pembuatan minimal 28 hari. Jika menentukan frekuensi bulanan yang membuat snapshot pada hari tertentu dalam minggu tertentu (misalnya, Kamis kedua setiap bulan), untuk jadwal berbasis hitungan, hitungan retensi untuk tingkat arsip harus 4 atau lebih.

        iii. Untuk Dimulai pada, tentukan waktu pelaksanaan kebijakan dijadwalkan untuk dimulai. Pelaksanaan kebijakan pertama dimulai dalam waktu satu jam setelah waktu yang dijadwalkan. Waktu harus dimasukkan dalam format `hh:mm` UTC.

        iv. Untuk Jenis retensi, tentukan kebijakan retensi untuk snapshot yang dibuat berdasarkan jadwal.

        Anda dapat mempertahankan snapshot berdasarkan total jumlah atau usianya.

        - Retensi berbasis jumlah

- Dengan pengarsipan snapshot dinonaktifkan, rentangnya adalah dari 1 hingga 1000. Saat ambang batas retensi tercapai, snapshot terlama dihapus secara permanen.
- Dengan pengarsipan snapshot diaktifkan, rentangnya adalah dari 0 (diarsipkan segera setelah pembuatan) hingga 1000. Saat ambang batas retensi tercapai, snapshot terlama dikonversi ke snapshot penuh dan dipindahkan ke tingkat arsip.

- Retensi berbasis usia
  - Dengan pengarsipan snapshot dinonaktifkan, rentangnya adalah dari 1 hingga 100 tahun. Saat ambang batas retensi tercapai, snapshot terlama dihapus secara permanen.
  - Dengan pengarsipan snapshot diaktifkan, rentangnya adalah dari 0 hari (diarsipkan segera setelah pembuatan) hingga 100 tahun. Saat ambang batas retensi tercapai, snapshot terlama dikonversi ke snapshot penuh dan dipindahkan ke tingkat arsip.

> **Note**
>
> - Semua jadwal harus memiliki jenis retensi yang sama (berbasis usia atau berbasis hitungan). Anda dapat menentukan jenis retensi hanya untuk Jadwal 1. Jadwal 2, 3, dan 4 mewarisi jenis retensi dari Jadwal 1. Setiap jadwal dapat memiliki jumlah atau periode retensi sendiri.
> - Jika Anda mengaktifkan pemulihan snapshot cepat, salinan lintas Wilayah, atau berbagi snapshot, Anda harus menentukan jumlah retensi 1 atau lebih, atau periode penyimpanan 1 hari atau lebih lama.

v. (hanya AWS Outposts pelanggan) Tentukan tujuan snapshot.

Untuk Tujuan snapshot, tentukan tujuan snapshot yang dibuat oleh kebijakan.

- Jika kebijakan menargetkan sumber daya di Wilayah, snapshot harus dibuat di Wilayah yang sama. AWS Wilayah dipilih untuk Anda.
- Jika kebijakan menargetkan sumber daya di Outposts, Anda dapat memilih untuk membuat snapshot di Outposts yang sama sebagai sumber daya sumber, atau di Wilayah yang terkait dengan Outposts.

- Jika Anda tidak memiliki Outposts di akun Anda, opsi ini disembunyikan dan AWS Wilayah dipilih untuk Anda.

b. Konfigurasikan penandaan untuk snapshot.

Di bagian Penandaan, lakukan hal berikut ini:

i. Untuk menyalin semua tanda yang ditentukan pengguna dari volume sumber ke snapshot yang dibuat oleh jadwal, pilih Salin tanda dari sumber.

ii. Untuk menentukan tanda tambahan yang akan ditetapkan ke snapshot yang dibuat oleh jadwal ini, pilih Tambahkan tanda.

c. Konfigurasikan skrip pra dan pasca untuk snapshot yang konsisten dengan aplikasi.

Untuk informasi selengkapnya, lihat Mengotomatiskan snapshot yang konsisten dengan aplikasi dengan skrip pra dan pasca.

d. (Kebijakan yang hanya menargetkan volume) Konfigurasikan pengarsipan snapshot.

Di bagian Pengarsipan snapshot, lakukan hal berikut:

> **Note**
>
> Anda dapat mengaktifkan pengarsipan snapshot hanya untuk satu jadwal dalam kebijakan.

i. Untuk mengaktifkan pengarsipan snapshot untuk jadwal, pilih Arsipkan snapshot yang dibuat oleh jadwal ini.

> **Note**
>
> Anda dapat mengaktifkan pengarsipan snapshot hanya jika frekuensi pembuatan snapshot bersifat bulanan atau tahunan, atau jika Anda menentukan ekspresi cron dengan frekuensi pembuatan minimal 28 hari.

ii. Tentukan aturan retensi untuk snapshot di tingkat arsip.

- Untuk jadwal berbasis jumlah, tentukan jumlah snapshot yang akan dipertahankan di tingkat arsip. Ketika ambang batas retensi tercapai, snapshot paling lama dihapus secara permanen dari tingkat arsip. Misalnya, jika Anda menentukan

3, jadwal akan mempertahankan maksimal 3 snapshot di tingkat arsip. Ketika snapshot keempat diarsipkan, yang tertua dari tiga snapshot yang ada di tingkat arsip dihapus.

- Untuk jadwal berbasis usia, tentukan periode waktu untuk mempertahankan snapshot di tingkat arsip. Ketika ambang batas retensi tercapai, snapshot paling lama dihapus secara permanen dari tingkat arsip. Misalnya, jika Anda menentukan 120 hari, jadwal akan secara otomatis menghapus snapshot dari tingkat arsip ketika mereka mencapai usia tersebut.

> **Important**
>
> Batas penyimpanan snapshot minimum adalah 90 hari. Anda harus menentukan aturan retensi yang mempertahankan snapshot setidaknya selama 90 hari.

e. Aktifkan pemulihan snapshot cepat.

Untuk mengaktifkan pemulihan snapshot cepat untuk snapshot yang dibuat oleh jadwal, di bagian Pemulihan snapshot cepat, pilih Aktifkan pemulihan snapshot cepat. Jika Anda mengaktifkan pemulihan snapshot cepat, Anda harus memilih Zona Ketersediaan untuk tempat mengaktifkannya. Jika jadwal menggunakan jadwal retensi berbasis usia, Anda harus menentukan periode untuk mengaktifkan pemulihan snapshot cepat untuk setiap snapshot. Jika jadwal menggunakan retensi berbasis jumlah, Anda harus menentukan jumlah maksimal snapshot untuk mengaktifkan pemulihan snapshot cepat.

Jika jadwal membuat snapshot di Outposts, Anda tidak dapat mengaktifkan pemulihan snapshot cepat. Pemulihan snapshot cepat tidak didukung dengan snapshot lokal yang disimpan di Outposts.

> **Note**
>
> Anda dikenai biaya untuk setiap menit pengaktifan pemulihan snapshot cepat untuk snapshot dalam Zona Ketersediaan tertentu. Biaya bersifat pro-rata minimal satu jam.

f. Konfigurasikan salinan lintas Wilayah.

Untuk menyalin snapshot yang dibuat oleh jadwal ke Outposts atau ke Wilayah lain, di bagian Salinan lintas-Wilayah, pilih Aktifkan salinan lintas-Wilayah.

Jika jadwal membuat snapshot di Wilayah, Anda dapat menyalin snapshot hingga tiga Wilayah atau Outposts tambahan di akun Anda. Anda harus menentukan aturan salinan lintas Wilayah terpisah untuk setiap Wilayah atau Outposts tujuan.

Untuk setiap Wilayah atau Outposts, Anda dapat memilih kebijakan retensi yang berbeda dan Anda dapat memilih apakah menyalin semua tanda atau tidak ada tanda. Jika snapshot sumber dienkripsi, atau jika enkripsi secara default diaktifkan, snapshot yang disalin dienkripsi. Jika snapshot sumber tidak dienkripsi, Anda dapat mengaktifkan enkripsi. Jika Anda tidak menentukan kunci KMS, snapshot dienkripsi menggunakan kunci KMS default untuk enkripsi EBS di setiap Wilayah tujuan. Jika Anda menentukan kunci KMS untuk Wilayah tujuan, peran IAM yang dipilih harus memiliki akses ke kunci KMS.

> **Note**
>
> Anda harus memastikan bahwa Anda tidak melebihi jumlah salinan snapshot bersamaan per Wilayah.

Jika kebijakan membuat snapshot di Outposts, Anda tidak dapat menyalin snapshot ke Wilayah atau Outposts lain dan pengaturan salinan lintas Wilayah tidak tersedia.

g. Konfigurasikan berbagi lintas akun.

Dalam berbagi lintas akun, konfigurasikan kebijakan untuk secara otomatis membagikan snapshot yang dibuat oleh jadwal dengan akun lain AWS . Lakukan hal-hal berikut:

i. Untuk mengaktifkan berbagi dengan AWS akun lain, pilih Aktifkan berbagi lintas akun.

ii. Untuk menambahkan akun yang dapat digunakan untuk berbagi snapshot, pilih Tambahkan akun, masukkan 12 digit ID akun AWS , dan pilih Tambah.

iii. Untuk membatalkan berbagi snapshot yang dibagikan secara otomatis setelah periode tertentu, pilih Batalkan pembagian secara otomatis. Jika Anda memilih untuk secara otomatis membatalkan pembagian snapshot yang dinagikan, periode setelah itu untuk secara otomatis membatalkan pembagian snapshot tidak dapat lebih

lama dari periode untuk kebijakan mempertahankan snapshotnya. Misalnya, jika konfigurasi retensi kebijakan mempertahankan snapshot selama 5 hari, Anda dapat mengonfigurasi kebijakan untuk secara otomatis membatalkan pembagian snapshot yang dibagikan setelah periode hingga 4 hari. Hal ini berlaku untuk kebijakan dengan konfigurasi penyimpanan snapshot berbasis usia dan jumlah.

Jika Anda tidak mengaktifkan pembatalan pembagian otomatis, snapshot akan dibagikan hingga dihapus.

> **Note**
>
> Anda hanya dapat berbagi snapshot yang tidak dienkripsi atau yang dienkripsi menggunakan kunci yang dikelola pelanggan. Anda tidak dapat berbagi snapshot yang dienkripsi dengan kunci KMS enkripsi EBS default. Jika Anda berbagi snapshot terenkripsi, kemudian Anda juga harus berbagi kunci KMS yang digunakan untuk mengenkripsi volume sumber dengan akun target. Untuk informasi selengkapnya, lihat Mengizinkan pengguna di akun lain untuk menggunakan kunci KMS di Panduan Developer AWS Key Management Service .

h. Untuk menambahkan jadwal tambahan, pilih Tambahkan jadwal lain, yang terletak di bagian atas layar. Untuk setiap jadwal tambahan, lengkapi bidang seperti yang dijelaskan sebelumnya dalam topik ini.

i. Setelah Anda menambahkan jadwal yang diperlukan, pilih Tinjau kebijakan.

12. Tinjau ringkasan kebijakan, lalu pilih Buat kebijakan.

> **Note**
>
> Jika Anda mendapatkan kesalahan `Role with name AWSDataLifecycleManagerDefaultRole already exists`, lihat Pemecahan Masalah untuk informasi selengkapnya.

Command line

Gunakan perintah create-lifecycle-policy untuk membuat kebijakan siklus hidup snapshot. Untuk `PolicyType`, tentukan `EBS_SNAPSHOT_MANAGEMENT`.

> **Note**
>
> Untuk menyederhanakan sintaksis, contoh berikut menggunakan file JSON, `policyDetails.json`, yang mencakup detail kebijakan.

Contoh 1—Kebijakan siklus hidup snapshot dengan dua jadwal

Contoh ini membuat kebijakan siklus hidup snapshot yang membubat snapshot dari semua volume yang memiliki kunci tanda `costcenter` dengan nilai `115`. Kebijakan tersebut mencakup dua jadwal. Jadwal pertama membuat snapshot setiap hari pada pukul 03.00 UTC. Jadwal kedua membuat snapshot mingguan setiap Jumat pukul 17.00 UTC.

```
aws dlm create-lifecycle-policy \
    --description "My volume policy" \
    --state ENABLED \
    --execution-role-arn
 arn:aws:iam::12345678910:role/AWSDataLifecycleManagerDefaultRole \
    --policy-details file://policyDetails.json
```

Berikut ini adalah contoh file `policyDetails.json`.

```
{
    "PolicyType": "EBS_SNAPSHOT_MANAGEMENT",
    "ResourceTypes": [
        "VOLUME"
    ],
    "TargetTags": [{
        "Key": "costcenter",
        "Value": "115"
    }],
    "Schedules": [{
        "Name": "DailySnapshots",
        "TagsToAdd": [{
            "Key": "type",
            "Value": "myDailySnapshot"
        }],
        "CreateRule": {
            "Interval": 24,
            "IntervalUnit": "HOURS",
            "Times": [
                "03:00"
```

```
            ]
        },
        "RetainRule": {
            "Count": 5
        },
        "CopyTags": false
    },
    {
        "Name": "WeeklySnapshots",
        "TagsToAdd": [{
            "Key": "type",
            "Value": "myWeeklySnapshot"
        }],
        "CreateRule": {
            "CronExpression": "cron(0 17 ? * FRI *)"
        },
        "RetainRule": {
            "Count": 5
        },
        "CopyTags": false
    }
]}
```

Jika permintaan berhasil, perintah mengembalikan ID kebijakan yang baru dibuat. Berikut ini adalah output contoh.

```
{
   "PolicyId": "policy-0123456789abcdef0"
}
```

Contoh 2—Kebijakan siklus hidup snapshot yang menargetkan instans dan membuat snapshot dari subset volume data (non-root)

Contoh ini membuat kebijakan siklus hidup snapshot yang membuat set snapshot multi-volume dari instans yang ditandai dengan `code=production`. Kebijakan ini hanya mencakup satu jadwal. Jadwal tidak membuat snapshot dari volume data yang ditandai dengan `code=temp`.

```
aws dlm create-lifecycle-policy \
    --description "My volume policy" \
    --state ENABLED \
    --execution-role-arn
 arn:aws:iam::12345678910:role/AWSDataLifecycleManagerDefaultRole \
```

```
    --policy-details file://policyDetails.json
```

Berikut ini adalah contoh file `policyDetails.json`.

```
{
    "PolicyType": "EBS_SNAPSHOT_MANAGEMENT",
    "ResourceTypes": [
        "INSTANCE"
    ],
    "TargetTags": [{
        "Key": "code",
        "Value": "production"
    }],
    "Parameters": {
        "ExcludeDataVolumeTags": [{
            "Key": "code",
            "Value": "temp"
        }]
    },
    "Schedules": [{
        "Name": "DailySnapshots",
        "TagsToAdd": [{
            "Key": "type",
            "Value": "myDailySnapshot"
        }],
        "CreateRule": {
            "Interval": 24,
            "IntervalUnit": "HOURS",
            "Times": [
                "03:00"
            ]
        },
        "RetainRule": {
            "Count": 5
        },
        "CopyTags": false
    }
]}
```

Jika permintaan berhasil, perintah mengembalikan ID kebijakan yang baru dibuat. Berikut ini adalah output contoh.

```
{
```

```
    "PolicyId": "policy-0123456789abcdef0"
}
```

Contoh 3—Kebijakan siklus hidup snapshot yang mengotomatisasi snapshot lokal pada sumber daya Outposts

Contoh ini membuat kebijakan siklus hidup snapshot yang membuat snapshot volume yang ditandai dengan `team=dev` di semua Outposts Anda. Kebijakan menciptakan snapshot pada Outposts yang sama sebagai sumber volume. Kebijakan ini menciptakan snapshot setiap 12 jam mulai pukul `00:00` UTC.

```
aws dlm create-lifecycle-policy \
    --description "My local snapshot policy" \
    --state ENABLED \
    --execution-role-arn
 arn:aws:iam::12345678910:role/AWSDataLifecycleManagerDefaultRole \
    --policy-details file://policyDetails.json
```

Berikut ini adalah contoh file `policyDetails.json`.

```
{
    "PolicyType": "EBS_SNAPSHOT_MANAGEMENT",
    "ResourceTypes": "VOLUME",
 "ResourceLocations": "OUTPOST",
    "TargetTags": [{
        "Key": "team",
        "Value": "dev"
    }],
    "Schedules": [{
        "Name": "on-site backup",
        "CreateRule": {
            "Interval": 12,
            "IntervalUnit": "HOURS",
            "Times": [
                "00:00"
            ],
 "Location": [
  "OUTPOST_LOCAL"
 ]
        },
        "RetainRule": {
            "Count": 1
```

```
        },
        "CopyTags": false
    }
]}
```

Contoh 4—Kebijakan siklus hidup snapshot yang membuat snapshot di suatu Wilayah dan menyalinnya ke Outposts

Kebijakan contoh berikut membuat snapshot volume yang ditandai dengan `team=dev`. Snapshot dibuat di Wilayah yang sama dengan volume sumber. Snapshot dibuat setiap 12 jam mulai pukul `00:00` UTC, dan mempertahankan maksimum 1 snapshot. Kebijakan ini juga menyalin snapshot ke Outposts `arn:aws:outposts:us-east-1:123456789012:outpost/op-1234567890abcdef0`, mengenkripsi snapshot yang disalin menggunakan kunci KMS enkripsi default, dan mempertahankan salinan selama 1 bulan.

```
aws dlm create-lifecycle-policy \
    --description "Copy snapshots to Outpost" \
    --state ENABLED \
    --execution-role-arn
 arn:aws:iam::12345678910:role/AWSDataLifecycleManagerDefaultRole \
    --policy-details file://policyDetails.json
```

Berikut ini adalah contoh file `policyDetails.json`.

```
{
    "PolicyType": "EBS_SNAPSHOT_MANAGEMENT",
    "ResourceTypes": "VOLUME",
    "ResourceLocations": "CLOUD",
    "TargetTags": [{
        "Key": "team",
        "Value": "dev"
    }],
    "Schedules": [{
        "Name": "on-site backup",
        "CopyTags": false,
        "CreateRule": {
            "Interval": 12,
            "IntervalUnit": "HOURS",
            "Times": [
                "00:00"
            ],
            "Location": "CLOUD"
```

```
        },
        "RetainRule": {
            "Count": 1
        },
        "CrossRegionCopyRules" : [
        {
            "Target": "arn:aws:outposts:us-east-1:123456789012:outpost/
op-1234567890abcdef0",
            "Encrypted": true,
            "CopyTags": true,
            "RetainRule": {
                "Interval": 1,
                "IntervalUnit": "MONTHS"
            }
        }]
    }
]}
```

Contoh 5—Kebijakan siklus hidup snapshot dengan jadwal berbasis usia dengan pengarsipan aktif

Contoh ini membuat kebijakan siklus hidup snapshot yang menargetkan volume yang ditandai dengan `Name=Prod`. Kebijakan ini memiliki satu jadwal berbasis usia yang membuat snapshot pada hari pertama setiap bulan pada pukul 09:00. Jadwal ini mempertahankan setiap snapshot di tingkat standar selama satu hari, setelah itu memindahkannya ke tingkat arsip. Snapshot disimpan di tingkat arsip selama 90 hari sebelum dihapus.

```
aws dlm create-lifecycle-policy \
    --description "Copy snapshots to Outpost" \
    --state ENABLED \
    --execution-role-arn
 arn:aws:iam::12345678910:role/AWSDataLifecycleManagerDefaultRole \
    --policy-details file://policyDetails.json
```

Berikut ini adalah contoh file `policyDetails.json`.

```
{
    "ResourceTypes": [ "VOLUME"],
    "PolicyType": "EBS_SNAPSHOT_MANAGEMENT",
    "Schedules" : [
      {
        "Name": "sched1",
```

```
        "TagsToAdd": [
          {"Key":"createdby","Value":"dlm"}
        ],
        "CreateRule": {
          "CronExpression": "cron(0 9 1 * ? *)"
        },
        "CopyTags": true,
        "RetainRule":{
          "Interval": 1,
          "IntervalUnit": "DAYS"
        },
        "ArchiveRule": {
            "RetainRule":{
              "RetentionArchiveTier": {
                 "Interval": 90,
                 "IntervalUnit": "DAYS"
              }
            }
        }
      }
    ],
    "TargetTags": [
      {
        "Key": "Name",
        "Value": "Prod"
      }
    ]
}
```

Contoh 6—Kebijakan siklus hidup snapshot dengan jadwal berbasis jumlah dengan pengarsipan aktif

Contoh ini membuat kebijakan siklus hidup snapshot yang menargetkan volume yang ditandai dengan `Purpose=Test`. Kebijakan ini memiliki satu jadwal berbasis jumlah yang membuat snapshot pada hari pertama setiap bulan pada pukul 09:00. Jadwal ini mengarsipkan snapshot segera setelah pembuatan dan mempertahankan maksimal tiga snapshot di tingkat arsip.

```
aws dlm create-lifecycle-policy \
    --description "Copy snapshots to Outpost" \
    --state ENABLED \
    --execution-role-arn
 arn:aws:iam::12345678910:role/AWSDataLifecycleManagerDefaultRole \
```

```
    --policy-details file://policyDetails.json
```

Berikut ini adalah contoh file `policyDetails.json`.

```
{
    "ResourceTypes": [ "VOLUME"],
    "PolicyType": "EBS_SNAPSHOT_MANAGEMENT",
    "Schedules" : [
      {
        "Name": "sched1",
        "TagsToAdd": [
          {"Key":"createdby","Value":"dlm"}
        ],
        "CreateRule": {
          "CronExpression": "cron(0 9 1 * ? *)"
        },
        "CopyTags": true,
        "RetainRule":{
          "Count": 0
        },
        "ArchiveRule": {
            "RetainRule":{
              "RetentionArchiveTier": {
                 "Count": 3
              }
            }
        }
      }
    ],
    "TargetTags": [
      {
        "Key": "Purpose",
        "Value": "Test"
      }
    ]
}
```

Pertimbangan untuk kebijakan siklus hidup snapshot

Pertimbangan umum berikut ini berlaku untuk snapshot kebijakan siklus hidup:

- Kebijakan siklus hidup snapshot hanya menargetkan instans atau volume yang berada di Wilayah yang sama dengan kebijakan.
- Operasi pembuatan snapshot pertama dimulai dalam waktu satu jam setelah waktu mulai yang ditentukan. Operasi pembuatan snapshot selanjutnya dimulai dalam waktu yang dijadwalkan selama satu jam.
- Anda dapat membuat lebih dari satu kebijakan untuk mencadangkan volume atau instans. Misalnya, jika volume memiliki dua tanda, yaitu tanda A adalah target untuk kebijakan A untuk membuat snapshot setiap 12 jam, dan tanda B adalah target untuk kebijakan B untuk membuat snapshot setiap 24 jam, Amazon Data Lifecycle Manager membuat snapshot sesuai jadwal untuk kedua kebijakan. Atau, Anda dapat mencapai hasil yang sama dengan membuat satu kebijakan yang memiliki beberapa jadwal. Misalnya, Anda dapat membuat kebijakan tunggal yang hanya menargetkan tanda A, dan menentukan dua jadwal — satu untuk setiap 12 jam dan satu untuk setiap 24 jam.
- Tanda sumber daya peka huruf besar dan kecil.
- Jika Anda menghapus tanda target dari sumber daya yang ditargetkan oleh kebijakan, Amazon Data Lifecycle Manager tidak lagi mengelola snapshot yang ada di tingkat standar dan tingkat arsip; Anda harus menghapusnya secara manual jika tidak diperlukan lagi.
- Jika Anda membuat kebijakan yang menargetkan instans, dan volume baru dilampirkan ke instans target setelah kebijakan dibuat, volume yang baru ditambahkan disertakan dalam pencadangan pada saat pelaksanaan kebijakan berikutnya. Semua volume yang dilampirkan pada instans saat pelaksanaan kebijakan disertakan.
- Jika Anda membuat kebijakan dengan jadwal berbasis cron kustom yang dikonfigurasi untuk membuat hanya satu snapshot, kebijakan tidak akan secara otomatis menghapus snapshot ketika ambang retensi tercapai. Anda harus menghapus snapshot secara manual jika tidak lagi diperlukan.
- Jika Anda membuat kebijakan berbasis usia dengan periode retensi lebih pendek dari frekuensi pembuatan, Amazon Data Lifecycle Manager akan selalu mempertahankan snapshot terakhir hingga snapshot berikutnya dibuat. Misalnya, jika kebijakan berbasis usia membuat satu snapshot setiap bulan dengan periode retensi tujuh hari, Amazon Data Lifecycle Manager akan mempertahankan setiap snapshot selama satu bulan, meskipun periode retensi adalah tujuh hari.

Pertimbangan berikut berlaku untuk pengarsipan snapshot:

- Anda dapat mengaktifkan pengarsipan snapshot hanya untuk kebijakan snapshot yang menargetkan volume.

- Anda dapat menentukan aturan pengarsipan hanya untuk satu jadwal untuk setiap kebijakan.
- Jika menggunakan konsol, Anda dapat mengaktifkan pengarsipan snapshot hanya jika frekuensi pembuatannya adalah bulanan atau tahunan, atau jika Anda menjadwalkan ekspresi cron dengan frekuensi pembuatan minimal 28 hari.

  Jika Anda menggunakan AWS API AWS CLI, atau AWS SDK, Anda dapat mengaktifkan pengarsipan snapshot hanya jika jadwal memiliki ekspresi cron dengan frekuensi pembuatan minimal 28 hari.
- Periode retensi minimum di tingkat arsip adalah 90 hari.
- Ketika diarsipkan, snapshot dikonversi ke snapshot penuh ketika dipindahkan ke tingkat arsip. Hal ini dapat mengakibatkan biaya penyimpanan snapshot yang lebih tinggi. Untuk informasi selengkapnya, lihat Harga dan penagihan.
- Pemulihan snapshot cepat dan berbagi snapshot dinonaktifkan untuk snapshot saat diarsipkan.
- Jika, dalam kasus tahun kabisat, aturan retensi Anda menghasilkan periode penyimpanan arsip kurang dari 90 hari, Amazon Data Lifecycle Manager memastikan bahwa snapshot dipertahankan untuk periode minimum 90 hari.
- Jika Anda mengarsipkan snapshot yang dibuat oleh Amazon Data Lifecycle Manager secara manual, dan snapshot masih diarsipkan saat ambang retensi jadwal tercapai, Amazon Data Lifecycle Manager tidak lagi mengelola snapshot tersebut. Namun, jika Anda mengembalikan snapshot ke tingkat standar sebelum ambang retensi jadwal tercapai, jadwal akan terus mengelola snapshot sesuai aturan retensi.
- Jika Anda mengarsipkan snapshot yang dibuat oleh Amazon Data Lifecycle Manager secara manual, dan snapshot masih diarsipkan saat ambang penyimpanan jadwal tercapai, Amazon Data Lifecycle Manager tidak lagi mengelola snapshot tersebut. Namun, jika Anda mengarsipkan ulang snapshot sebelum ambang retensi jadwal tercapai, jadwal akan menghapus snapshot saat ambang retensi terpenuhi.
- Snapshot yang diarsipkan oleh Amazon Data Lifecycle Manager dihitung terhadap kuota `Archived snapshots per volume` dan `In-progress snapshot archives per account` Anda.
- Jika jadwal tidak dapat mengarsipkan snapshot setelah mencoba lagi selama 24 jam, snapshot tetap berada di tingkat standar dan dijadwalkan untuk dihapus berdasarkan waktu yang akan dihapus dari tingkat arsip. Misalnya, jika jadwal mengarsipkan snapshot selama 120 hari, snapshot tetap dalam tingkat standar selama 120 hari setelah pengarsipan gagal sebelum dihapus secara permanen. Untuk jadwal berbasis jumlah, snapshot tidak dihitung terhadap jumlah retensi jadwal.

- Snapshot harus diarsipkan di Wilayah yang sama dengan tempat pembuatannya. Jika Anda mengaktifkan salinan lintas Wilayah dan pengarsipan snapshot, Amazon Data Lifecycle Manager tidak mengarsipkan salinan snapshot.
- Snapshot yang diarsipkan oleh Amazon Data Lifecycle Manager ditandai dengan tanda sistem `aws:dlm:archived=true`. Selain itu, snapshot yang dibuat oleh jadwal berbasis usia yang diaktifkan arsip ditandai dengan tanda sistem `aws:dlm:expirationTime`, yang menunjukkan tanggal dan waktu snapshot dijadwalkan untuk diarsipkan.

Pertimbangan berikut berlaku untuk mengecualikan volume root dan volume data (non-root):

- Jika Anda memilih untuk mengecualikan volume boot dan Anda menentukan tag yang akibatnya mengecualikan semua volume data tambahan yang dilampirkan ke instance, maka Amazon Data Lifecycle Manager tidak akan membuat snapshot apa pun untuk instance yang terpengaruh, dan akan mengeluarkan metrik. `SnapshotsCreateFailed` CloudWatch Untuk informasi selengkapnya, lihat Memantau kebijakan Anda menggunakan CloudWatch.

Pertimbangan berikut berlaku untuk menghapus volume atau mengakhiri instans yang ditargetkan oleh kebijakan siklus hidup snapshot:

- Jika Anda menghapus volume atau mengakkhiri instans yang ditargetkan oleh kebijakan dengan jadwal retensi berbasis jumlah, Amazon Data Lifecycle Manager tidak lagi mengelola snapshot di tingkat standar dan tingkat arsip yang dibuat dari volume atau instans yang dihapus. Anda harus menghapus snapshot secara manual jika tidak lagi diperlukan.
- Jika Anda menghapus volume atau mengakhiri instans yang ditargetkan oleh kebijakan dengan jadwal penyimpanan berbasis usia, kebijakan tersebut terus menghapus snapshot dari tingkat standar dan tingkat arsip yang dibuat dari volume atau instans yang dihapus pada jadwal yang ditentukan hingga, tetapi tidak termasuk snapshot terakhir. Anda harus menghapus snapshot secara manual jika tidak lagi diperlukan.

Pertimbangan berikut ini berlaku untuk kebijakan siklus hidup snapshot dan pemulihan snapshot cepat:

- Amazon Data Lifecycle Manager dapat mengaktifkan pemulihan snapshot cepat hanya untuk snapshot dengan ukuran 16 TiB atau kurang. Untuk informasi selengkapnya, lihat Pemulihan snapshot cepat Amazon EBS.

- Snapshot yang diaktifkan untuk pemulihan snapshot cepat tetap aktif meskipun Anda menghapus atau menonaktifkan kebijakan, menonaktifkan pemulihan snapshot cepat untuk kebijakan, atau menonaktifkan pemulihan snapshot cepat untuk Zona Ketersediaan. Anda harus menonaktifkan pemulihan snapshot cepat untuk snapshot ini secara manual.
- Jika Anda mengaktifkan pemulihan snapshot cepat untuk suatu kebijakan dan melebihi jumlah maksimum snapshot yang dapat diaktifkan untuk pemulihan snapshot cepat, Amazon Data Lifecycle Manager membuat snapshot sesuai jadwal, tetapi tidak mengaktifkannya untuk pemulihan snapshot cepat. Setelah snapshot yang diaktifkan untuk pemulihan snapshot cepat dihapus, snapshot berikutnya yang dibuat Amazon Data Lifecycle Manager diaktifkan untuk pemulihan snapshot cepat.
- Ketika pemulihan snapshot cepat diaktifkan untuk snapshot, hal ini memakan waktu 60 menit per TiB untuk mengoptimalkan snapshot. Sebaiknya konfigurasikan jadwal Anda sehingga setiap snapshot sepenuhnya dioptimalkan sebelum Amazon Data Lifecycle Manager membuat snapshot berikutnya.
- Jika Anda mengaktifkan pemulihan snapshot cepat untuk kebijakan yang menargetkan instans, Amazon Data Lifecycle Manager mengaktifkan pemulihan snapshot cepat untuk setiap snapshot dalam snapshot multi-volume yang diatur secara individu. Jika gagal mengaktifkan pemulihan snapshot cepat untuk salah satu snapshot dalam set snapshot multi-volume, Amazon Data Lifecycle Manager masih akan mencoba mengaktifkan pemulihan snapshot cepat untuk snapshot yang tersisa dalam set snapshot.
- Anda dikenai biaya untuk setiap menit pengaktifan pemulihan snapshot cepat untuk snapshot dalam Zona Ketersediaan tertentu. Biaya bersifat pro-rata minimal satu jam. Untuk informasi selengkapnya, lihat Harga dan Penagihan.

> **Note**
>
> Bergantung pada konfigurasi kebijakan siklus hidup, Anda dapat mengaktifkan banyak snapshot untuk pemulihan snapshot cepat di banyak Zona Ketersediaan secara bersamaan.

Pertimbangan berikut ini berlaku untuk kebijakan siklus hidup snapshot dan volume dengan dukungan Multi-Lampiran:

- Saat membuat kebijakan siklus hidup yang menargetkan instans yang memiliki volume Multi-Lampiran aktif, Amazon Data Lifecycle Manager memulai snapshot volume untuk setiap instans

yang dilampirkan. Gunakan tanda stempel waktu untuk mengidentifikasi sejumlah snapshot yang konsisten dengan waktu yang dibuat dari instans yang dilampirkan.

Pertimbangan berikut berlaku untuk berbagi snapshot antar-akun:

- Anda hanya dapat berbagi snapshot yang tidak dienkripsi atau yang dienkripsi menggunakan kunci yang dikelola pelanggan.
- Anda tidak dapat berbagi snapshot yang dienkripsi dengan kunci KMS enkripsi EBS default.
- Jika Anda berbagi snapshot terenkripsi, Anda juga harus berbagi kunci KMS yang digunakan untuk mengenkripsi volume sumber dengan akun target. Untuk informasi selengkapnya, lihat Mengizinkan pengguna di akun lain untuk menggunakan kunci KMS di Panduan Developer AWS Key Management Service .

Pertimbangan berikut berlaku untuk kebijakan snapshot dan pengarsipan snapshot:

- Jika Anda mengarsipkan snapshot yang dibuat oleh kebijakan secara manual, dan snapshot tersebut ada di tingkat arsip saat ambang penyimpanan kebijakan tercapai, Amazon Data Lifecycle Manager tidak akan menghapus snapshot tersebut. Amazon Data Lifecycle Manager tidak mengelola snapshot saat disimpan di tingkat arsip. Jika Anda tidak lagi membutuhkan snapshot yang disimpan di tingkat arsip, Anda harus menghapusnya secara manual.

Pertimbangan berikut berlaku untuk kebijakan snapshot dan Recycle Bin:

- Jika Amazon Data Lifecycle Manager menghapus snapshot dan mengirimkannya ke Keranjang Sampah saat ambang penyimpanan kebijakan tercapai, dan memulihkan snapshot dari Keranjang Sampah secara manual, Anda harus menghapus snapshot tersebut secara manual saat tidak diperlukan lagi. Amazon Data Lifecycle Manager tidak akan lagi mengelola snapshot.
- Jika Anda menghapus snapshot yang dibuat oleh kebijakan secara manual, dan snapshot tersebut ada di Keranjang Sampah saat ambang penyimpanan kebijakan tercapai, Amazon Data Lifecycle Manager tidak akan menghapus snapshot tersebut. Amazon Data Lifecycle Manager tidak mengelola snapshot saat disimpan di Keranjang Sampah.

  Jika snapshot dipulihkan dari Keranjang Sampah sebelum ambang retensi kebijakan tercapai, Amazon Data Lifecycle Manager akan menghapus snapshot tersebut saat ambang retensi kebijakan tercapai.

Jika snapshot dipulihkan dari Keranjang Sampah setelah ambang batas retensi kebijakan tercapai, Amazon Data Lifecycle Manager tidak akan lagi menghapus snapshot tersebut. Anda harus menghapus snapshot secara manual saat tidak lagi diperlukan.

Pertimbangan umum berikut ini berlaku untuk kebijakan siklus hidup snapshot yang berada dalam status kesalahan:

- Untuk kebijakan dengan jadwal retensi berbasis usia, snapshot yang akan kedaluwarsa saat kebijakan berada dalam status `error` akan dipertahankan tanpa batas. Anda harus menghapus snapshot secara manual. Saat Anda mengaktifkan ulang kebijakan, Amazon Data Lifecycle Manager akan melanjutkan penghapusan snapshot karena periode retensinya kedaluwarsa.
- Untuk kebijakan dengan jadwal retensi berbasis jumlah, kebijakan berhenti membuat dan menghapus AMI saat berada dalam status `error`. Saat Anda mengaktifkan kembali kebijakan, Amazon Data Lifecycle Manager akan melanjutkan pembuatan snapshot, dan melanjutkan penghapusan snapshot saat ambang retensi terpenuhi.

Pertimbangan berikut berlaku untuk kebijakan snapshot dan kunci snapshot:

- Jika Anda mengunci snapshot yang dibuat secara manual oleh Amazon Data Lifecycle Manager, dan snapshot tersebut masih terkunci ketika ambang batas retensinya tercapai, Amazon Data Lifecycle Manager tidak lagi mengelola snapshot tersebut. Anda harus menghapus snapshot secara manual jika tidak lagi diperlukan.
- Jika Anda mengunci snapshot secara manual yang dibuat dan diaktifkan untuk pemulihan snapshot cepat oleh Amazon Data Lifecycle Manager, dan snapshot masih terkunci saat ambang batas retensinya tercapai, Amazon Data Lifecycle Manager tidak akan menonaktifkan pemulihan snapshot cepat atau menghapus snapshot. Anda harus menghapus snapshot secara manual jika tidak lagi diperlukan.
- Jika Anda mendaftarkan snapshot yang dibuat secara manual oleh Amazon Data Lifecycle Manager dengan AMI, lalu mengunci snapshot tersebut, dan snapshot tersebut masih terkunci serta dikaitkan dengan AMI saat ambang batas retensinya tercapai, Amazon Data Lifecycle Manager akan terus berusaha menghapus snapshot tersebut. Ketika AMI dibatalkan pendaftarannya dan snapshot tidak terkunci, Amazon Data Lifecycle Manager akan secara otomatis menghapus snapshot tersebut.

Sumber daya tambahan

Untuk informasi selengkapnya, lihat Mengotomatiskan snapshot Amazon EBS dan manajemen AMI menggunakan blog penyimpanan Amazon Data AWS Lifecycle Manager.

Persyaratan untuk menggunakan skrip pra dan pasca

Tabel berikut menguraikan persyaratan untuk menggunakan skrip pra dan pasca dengan Amazon Data Lifecycle Manager.

	Snapshot yang konsisten dengan aplikasi		
Persyaratan	Cadangan VSS	Dokumen SSM Kustom	Kasus penggunaan lainnya
Agen SSM diinstal dan berjalan pada instance target	✓	✓	✓
Persyaratan sistem VSS terpenuhi pada instance target	✓		
Profil instans berkemampuan VSS yang terkait dengan instance target	✓		
Komponen VSS diinstal pada instance target	✓		
Siapkan dokumen SSM dengan perintah skrip pra dan pasca		✓	✓
Mempersiapkan peran IAM Amazon Data Lifecycle Manager	✓	✓	✓

	Snapshot yang konsisten dengan aplikasi		
yang menjalankan skrip pra dan pasca			
Buat kebijakan snapshot yang menargetkan instance dan dikonfigurasi untuk skrip pra dan pasca	✓	✓	✓

Mengotomatiskan snapshot yang konsisten dengan aplikasi dengan skrip pra dan pasca

Anda dapat mengotomatiskan snapshot yang konsisten dengan aplikasi menggunakan Amazon Data Lifecycle Manager dengan mengaktifkan skrip pra dan pasca dalam kebijakan siklus hidup snapshot yang menargetkan instans.

Amazon Data Lifecycle Manager terintegrasi dengan (Systems AWS Systems Manager Manager) untuk mendukung snapshot yang konsisten dengan aplikasi. Amazon Data Lifecycle Manager menggunakan dokumen perintah Systems Manager (SSM) yang menyertakan skrip pra dan pasca untuk mengotomatisasi tindakan yang diperlukan untuk menyelesaikan snapshot yang konsisten dengan aplikasi. Sebelum memulai pembuatan snapshot, Amazon Data Lifecycle Manager menjalankan perintah dalam skrip pra untuk membekukan dan mencairkan I/O. Setelah memulai pembuatan snapshot, Amazon Data Lifecycle Manager menjalankan perintah dalam skrip pasca untuk mencairkan I/O.

Menggunakan Amazon Data Lifecycle Manager, Anda dapat mengotomatisasi snapshot yang konsisten dengan aplikasi berikut ini:

- Aplikasi Windows yang menggunakan Volume Shadow Copy Service (VSS)
- SAP HANA menggunakan dokumen SSDM AWS terkelola. Untuk informasi selengkapnya, lihat Snapshot Amazon EBS untuk SAP HANA.
- Database yang dikelola sendiri, seperti MySQL, PostgreSQL atau IRIS, menggunakan templat dokumen SSM InterSystems

Topik

Memulai snapshot yang konsisten dengan aplikasi

Bagian ini menjelaskan langkah-langkah yang perlu Anda ikuti untuk mengotomatisasi snapshot yang konsisten dengan aplikasi menggunakan Amazon Data Lifecycle Manager.

Langkah 1: Menyiapkan instans target

Anda perlu menyiapkan instans yang ditargetkan untuk snapshot yang konsisten dengan aplikasi menggunakan Amazon Data Lifecycle Manager. Lakukan salah satu langkah berikut sesuai dengan kasus penggunaan Anda.

Prepare for VSS Backups

Untuk mempersiapkan instans target Anda untuk cadangan VSS

1. Instal SSM Agent pada instans target Anda, jika belum diinstal. Jika SSM Agent sudah diinstal pada instans target Anda, lewati langkah ini.

   Untuk informasi selengkapnya, lihat Menginstal Agen SSM secara manual di instans Amazon EC2 untuk Windows.
2. Pastikan SSM Agent berjalan. Untuk informasi selengkapnya, lihat Memeriksa status SSM Agent dan memulai agen.
3. Siapkan Systems Manager untuk instans Amazon EC2. Untuk informasi selengkapnya, lihat Menyiapkan Systems Manager untuk instans Amazon EC2 di Panduan Pengguna AWS Systems Manager .
4. Pastikan persyaratan sistem untuk cadangan VSS terpenuhi.
5. Lampirkan profil instans dengan VSS yang diaktifkan ke instans target.
6. Instal komponen VSS.

Prepare for SAP HANA backups

Untuk mempersiapkan instans target Anda untuk cadangan SAP HANA

1. Siapkan lingkungan SAP HANA pada instans target Anda.

   a. Siapkan instans Anda dengan SAP HANA. Jika Anda belum memiliki lingkungan SAP HANA yang ada, Anda dapat merujuk ke Penyiapan Lingkungan SAP HANA di AWS.

   b. Masuk ke SystemDB sebagai pengguna administrator yang sesuai.

   c. Buat pengguna cadangan basis data untuk digunakan dengan Amazon Data Lifecycle Manager.

   ```
   CREATE USER username PASSWORD password NO FORCE_FIRST_PASSWORD_CHANGE;
   ```

   Misalnya, perintah berikut membuat pengguna bernama `dlm_user` dengan kata sandi `password`.

   ```
   CREATE USER dlm_user PASSWORD password NO FORCE_FIRST_PASSWORD_CHANGE;
   ```

   d. Tetapkan `BACKUP OPERATOR` peran ke pengguna cadangan basis data yang Anda buat di langkah sebelumnya.

   ```
   GRANT BACKUP OPERATOR TO username
   ```

   Misalnya, perintah berikut menetapkan peran untuk pengguna bernama `dlm_user`.

   ```
   GRANT BACKUP OPERATOR TO dlm_user
   ```

   e. Masuk ke sistem operasi sebagai administrator, misalnya *sid*`adm`.

   f. Buat entri `hdbuserstore` untuk menyimpan informasi koneksi sehingga dokumen SSM SAP HANA dapat terhubung ke SAP HANA tanpa pengguna harus memasukkan informasi tersebut.

   ```
   hdbuserstore set DLM_HANADB_SNAPSHOT_USER
    localhost:3hana_instance_number13 username password
   ```

   Misalnya:

```
hdbuserstore set DLM_HANADB_SNAPSHOT_USER localhost:30013 dlm_user password
```

g. Uji koneksi.

```
hdbsql -U DLM_HANADB_SNAPSHOT_USER "select * from dummy"
```

2. Instal SSM Agent pada instans target Anda, jika belum diinstal. Jika SSM Agent sudah diinstal pada instans target Anda, lewati langkah ini.

   Untuk informasi selengkapnya, lihat Menginstal Agen SSM secara manual di instans Amazon EC2 untuk Linux.

3. Pastikan SSM Agent berjalan. Untuk informasi selengkapnya, lihat Memeriksa status SSM Agent dan memulai agen.

4. Siapkan Systems Manager untuk instans Amazon EC2. Untuk informasi selengkapnya, lihat Menyiapkan Systems Manager untuk instans Amazon EC2 di Panduan Pengguna AWS Systems Manager .

Prepare for custom SSM documents

Untuk menyiapkan dokumen SSM kustom instans target Anda

1. Instal SSM Agent pada instans target Anda, jika belum diinstal. Jika SSM Agent sudah diinstal pada instans target Anda, lewati langkah ini.

   - (Instans Linux) Menginstal SSM Agent secara manual di instans Amazon EC2 untuk Linux
   - (Instans Windows) Menginstal SSM Agent secara manual di instans Amazon EC2 untuk Windows

2. Pastikan SSM Agent berjalan. Untuk informasi selengkapnya, lihat Memeriksa status SSM Agent dan memulai agen.

3. Siapkan Systems Manager untuk instans Amazon EC2. Untuk informasi selengkapnya, lihat Menyiapkan Systems Manager untuk instans Amazon EC2 di Panduan Pengguna AWS Systems Manager .

Langkah 2: Siapkan Dokumen SSM

> **Note**
>
> Langkah ini hanya diperlukan untuk dokumen SSM kustom. Hal ini tidak diperlukan untuk Cadangan VSS atau SAP HANA. Untuk Pencadangan VSS dan SAP HANA, Amazon Data Lifecycle Manager menggunakan dokumen SSM terkelola. AWS

Jika Anda mengotomatiskan snapshot yang konsisten aplikasi untuk database yang dikelola sendiri, seperti MySQL, PostgreSQL, atau InterSystems IRIS, Anda harus membuat dokumen perintah SSM yang menyertakan skrip pra untuk membekukan dan menyiram I/O sebelum pembuatan snapshot dimulai, dan skrip posting untuk mencairkan I/O setelah pembuatan snapshot dimulai.

Jika database MySQL, PostgreSQL, atau IRIS menggunakan konfigurasi standar InterSystems , Anda dapat membuat dokumen perintah SSM menggunakan contoh konten dokumen SSM di bawah ini. Jika database MySQL, PostgreSQL, atau IRIS Anda menggunakan konfigurasi non-standar InterSystems , Anda dapat menggunakan konten sampel di bawah ini sebagai titik awal untuk dokumen perintah SSM Anda dan kemudian menyesuaikannya untuk memenuhi kebutuhan Anda. Atau, jika Anda ingin membuat dokumen SSM baru dari awal, Anda dapat menggunakan templat dokumen SSM kosong di bawah ini dan menambahkan praperintah dan pascaperintah Anda di bagian dokumen yang sesuai.

> **Perhatikan hal-hal berikut:**
>
> - Anda bertanggung jawab untuk memastikan bahwa dokumen SSM melakukan tindakan yang benar dan diperlukan untuk konfigurasi basis data Anda.
> - Snapshot dijamin konsisten aplikasi hanya jika skrip pra dan pasca dalam dokumen SSM Anda berhasil membekukan, menyiram, dan mencairkan I/O.
> - Dokumen SSM harus menyertakan bidang wajib untuk `allowedValues`, termasuk, `pre-script`, `post-script`, dan `dry-run`. Amazon Data Lifecycle Manager akan menjalankan perintah pada instans Anda berdasarkan konten bagian tersebut. Jika dokumen SSM Anda tidak memiliki bagian tersebut, Amazon Data Lifecycle Manager akan memperlakukannya sebagai eksekusi yang gagal.

MySQL sample document content

```
###############################################################################
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

# Permission is hereby granted, free of charge, to any person obtaining a copy of
 this
# software and associated documentation files (the "Software"), to deal in the
 Software
# without restriction, including without limitation the rights to use, copy, modify,
# merge, publish, distribute, sublicense, and/or sell copies of the Software, and to
# permit persons to whom the Software is furnished to do so.

# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
 IMPLIED,
# INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
# PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
# HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
# OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
# SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
###############################################################################
schemaVersion: '2.2'
description: Amazon Data Lifecycle Manager Pre/Post script for MySQL databases
parameters:
  executionId:
    type: String
    default: None
    description: (Required) Specifies the unique identifier associated with a pre
 and/or post execution
    allowedPattern: ^(None|[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]
{4}-[a-fA-F0-9]{12})$
  command:
  # Data Lifecycle Manager will trigger the pre-script and post-script actions
 during policy execution.
  # 'dry-run' option is intended for validating the document execution without
 triggering any commands
  # on the instance. The following allowedValues will allow Data Lifecycle Manager
 to successfully
  # trigger pre and post script actions.
    type: String
    default: 'dry-run'
    description: (Required) Specifies whether pre-script and/or post-script should
 be executed.
    allowedValues:
```

```
    - pre-script
    - post-script
    - dry-run

mainSteps:
- action: aws:runShellScript
  description: Run MySQL Database freeze/thaw commands
  name: run_pre_post_scripts
  precondition:
    StringEquals:
    - platformType
    - Linux
  inputs:
    runCommand:
    - |
      #!/bin/bash


 ###============================================================================###
      ### Error Codes

 ###============================================================================###
      # The following Error codes will inform Data Lifecycle Manager of the type of
 error
      # and help guide handling of the error.
      # The Error code will also be emitted via AWS Eventbridge events in the
 'cause' field.
      # 1 Pre-script failed during execution - 201
      # 2 Post-script failed during execution - 202
      # 3 Auto thaw occurred before post-script was initiated - 203
      # 4 Pre-script initiated while post-script was expected - 204
      # 5 Post-script initiated while pre-script was expected - 205
      # 6 Application not ready for pre or post-script initiation - 206

      ###====================================================================###
      ### Global variables
      ###====================================================================###
      START=$(date +%s)
      # For testing this script locally, replace the below with OPERATION=$1.
      OPERATION={{ command }}
      FS_ALREADY_FROZEN_ERROR='freeze failed: Device or resource busy'
      FS_ALREADY_THAWED_ERROR='unfreeze failed: Invalid argument'
      FS_BUSY_ERROR='mount point is busy'
```

```
    # Auto thaw is a fail safe mechanism to automatically unfreeze the application
after the
    # duration specified in the global variable below. Choose the duration based
on your
    # database application's tolerance to freeze.
    export AUTO_THAW_DURATION_SECS="60"

    # Add all pre-script actions to be performed within the function below
    execute_pre_script() {
        echo "INFO: Start execution of pre-script"
        # Check if filesystem is already frozen. No error code indicates that
filesystem
        # is not currently frozen and that the pre-script can proceed with
freezing the filesystem.
        check_fs_freeze
        # Execute the DB commands to flush the DB in preparation for snapshot
        snap_db
        # Freeze the filesystem. No error code indicates that filesystem was
succefully frozen
        freeze_fs

        echo "INFO: Schedule Auto Thaw to execute in ${AUTO_THAW_DURATION_SECS}
seconds."
        $(nohup bash -c execute_schedule_auto_thaw  >/dev/null 2>&1 &)
    }

    # Add all post-script actions to be performed within the function below
    execute_post_script() {
        echo "INFO: Start execution of post-script"
        # Unfreeze the filesystem. No error code indicates that filesystem was
successfully unfrozen.
        unfreeze_fs
        thaw_db
    }

    # Execute Auto Thaw to automatically unfreeze the application after the
duration configured
    # in the AUTO_THAW_DURATION_SECS global variable.
    execute_schedule_auto_thaw() {
        sleep ${AUTO_THAW_DURATION_SECS}
        execute_post_script
    }

    # Disable Auto Thaw if it is still enabled
```

```
    execute_disable_auto_thaw() {
        echo "INFO: Attempting to disable auto thaw if enabled"
        auto_thaw_pgid=$(pgrep -f execute_schedule_auto_thaw | xargs -i ps -hp {}
-o pgid)
        if [ -n "${auto_thaw_pgid}" ]; then
            echo "INFO: execute_schedule_auto_thaw process found with pgid
${auto_thaw_pgid}"
            sudo pkill -g ${auto_thaw_pgid}
            rc=$?
            if [ ${rc} != 0 ]; then
                echo "ERROR: Unable to kill execute_schedule_auto_thaw process.
retval=${rc}"
            else
                echo "INFO: Auto Thaw  has been disabled"
            fi
        fi
    }

    # Iterate over all the mountpoints and check if filesystem is already in
freeze state.
    # Return error code 204 if any of the mount points are already frozen.
    check_fs_freeze() {
        for target in $(lsblk -nlo MOUNTPOINTS)
        do
            # Freeze of the root and boot filesystems is dangerous and pre-script
does not freeze these filesystems.
            # Hence, we will skip the root and boot mountpoints while checking if
filesystem is in freeze state.
            if [ $target == '/' ]; then continue; fi
            if [[ "$target" == *"/boot"* ]]; then continue; fi

            error_message=$(sudo mount -o remount,noatime $target 2>&1)
            # Remount will be a no-op without a error message if the filesystem is
unfrozen.
            # However, if filesystem is already frozen, remount will fail with
busy error message.
            if [ $? -ne 0 ];then
                # If the filesystem is already in frozen, return error code 204
                if [[ "$error_message" == *"$FS_BUSY_ERROR"* ]];then
                    echo "ERROR: Filesystem ${target} already frozen. Return Error
Code: 204"
                    exit 204
                fi
```

```
                # If the check filesystem freeze failed due to any reason other
than the filesystem already frozen, return 201
                echo "ERROR: Failed to check_fs_freeze on mountpoint $target due
to error - $errormessage"
                exit 201
            fi
        done
    }

    # Iterate over all the mountpoints and freeze the filesystem.
    freeze_fs() {
        for target in $(lsblk -nlo MOUNTPOINTS)
        do
            # Freeze of the root and boot filesystems is dangerous. Hence, skip
filesystem freeze
            # operations for root and boot mountpoints.
            if [ $target == '/' ]; then continue; fi
            if [[ "$target" == *"/boot"* ]]; then continue; fi
            echo "INFO: Freezing $target"
            error_message=$(sudo fsfreeze -f $target 2>&1)
            if [ $? -ne 0 ];then
                # If the filesystem is already in frozen, return error code 204
                if [[ "$error_message" == *"$FS_ALREADY_FROZEN_ERROR"* ]]; then
                    echo "ERROR: Filesystem ${target} already frozen. Return Error
Code: 204"
                    sudo mysql -e 'UNLOCK TABLES;'
                    exit 204
                fi
                # If the filesystem freeze failed due to any reason other than the
filesystem already frozen, return 201
                echo "ERROR: Failed to freeze mountpoint $targetdue due to error -
$errormessage"
                thaw_db
                exit 201
            fi
            echo "INFO: Freezing complete on $target"
        done
    }

    # Iterate over all the mountpoints and unfreeze the filesystem.
    unfreeze_fs() {
        for target in $(lsblk -nlo MOUNTPOINTS)
        do
```

```
            # Freeze of the root and boot filesystems is dangerous and pre-script
does not freeze these filesystems.
            # Hence, will skip the root and boot mountpoints during unfreeze as
well.
            if [ $target == '/' ]; then continue; fi
            if [[ "$target" == *"/boot"* ]]; then continue; fi
            echo "INFO: Thawing $target"
            error_message=$(sudo fsfreeze -u $target 2>&1)
            # Check if filesystem is already unfrozen (thawed). Return error code
204 if filesystem is already unfrozen.
            if [ $? -ne 0 ]; then
                if [[ "$error_message" == *"$FS_ALREADY_THAWED_ERROR"* ]]; then
                    echo "ERROR: Filesystem ${target} is already in thaw state.
Return Error Code: 205"
                    exit 205
                fi
                # If the filesystem unfreeze failed due to any reason other than
the filesystem already unfrozen, return 202
                echo "ERROR: Failed to unfreeze mountpoint $targetdue due to error
- $errormessage"
                exit 202
            fi
            echo "INFO: Thaw complete on $target"
        done
    }

    snap_db() {
        # Run the flush command only when MySQL DB service is up and running
        sudo systemctl is-active --quiet mysqld.service
        if [ $? -eq 0 ]; then
            echo "INFO: Execute MySQL Flush and Lock command."
            sudo mysql -e 'FLUSH TABLES WITH READ LOCK;'
            # If the MySQL Flush and Lock command did not succeed, return error
code 201 to indicate pre-script failure
            if [ $? -ne 0 ]; then
                echo "ERROR: MySQL FLUSH TABLES WITH READ LOCK command failed."
                exit 201
            fi
            sync
        else
            echo "INFO: MySQL service is inactive. Skipping execution of MySQL
Flush and Lock command."
        fi
    }
```

```
     thaw_db() {
         # Run the unlock command only when MySQL DB service is up and running
         sudo systemctl is-active --quiet mysqld.service
         if [ $? -eq 0 ]; then
             echo "INFO: Execute MySQL Unlock"
             sudo mysql -e 'UNLOCK TABLES;'
         else
             echo "INFO: MySQL service is inactive. Skipping execution of MySQL
 Unlock command."
         fi
     }

     export -f execute_schedule_auto_thaw
     export -f execute_post_script
     export -f unfreeze_fs
     export -f thaw_db

     # Debug logging for parameters passed to the SSM document
     echo "INFO: ${OPERATION} starting at $(date) with executionId:
 ${EXECUTION_ID}"

     # Based on the command parameter value execute the function that supports
     # pre-script/post-script operation
     case ${OPERATION} in
         pre-script)
             execute_pre_script
             ;;
         post-script)
             execute_post_script
             execute_disable_auto_thaw
             ;;
         dry-run)
             echo "INFO: dry-run option invoked - taking no action"
             ;;
         *)
             echo "ERROR: Invalid command parameter passed. Please use either pre-
script, post-script, dry-run."
             exit 1 # return failure
             ;;
     esac

     END=$(date +%s)
     # Debug Log for profiling the script time
```

```
      echo "INFO: ${OPERATION} completed at $(date). Total runtime: $((${END} -
 ${START})) seconds."
```

PostgreSQL sample document content

```
###############################################################################
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

# Permission is hereby granted, free of charge, to any person obtaining a copy of
 this
# software and associated documentation files (the "Software"), to deal in the
 Software
# without restriction, including without limitation the rights to use, copy, modify,
# merge, publish, distribute, sublicense, and/or sell copies of the Software, and to
# permit persons to whom the Software is furnished to do so.

# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
 IMPLIED,
# INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
# PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
# HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
# OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
# SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
###############################################################################
schemaVersion: '2.2'
description: Amazon Data Lifecycle Manager Pre/Post script for PostgreSQL databases
parameters:
  executionId:
    type: String
    default: None
    description: (Required) Specifies the unique identifier associated with a pre
 and/or post execution
    allowedPattern: ^(None|[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]
{4}-[a-fA-F0-9]{12})$
  command:
  # Data Lifecycle Manager will trigger the pre-script and post-script actions
 during policy execution.
  # 'dry-run' option is intended for validating the document execution without
 triggering any commands
  # on the instance. The following allowedValues will allow Data Lifecycle Manager
 to successfully
  # trigger pre and post script actions.
    type: String
```

```
    default: 'dry-run'
    description: (Required) Specifies whether pre-script and/or post-script should
 be executed.
    allowedValues:
    - pre-script
    - post-script
    - dry-run

mainSteps:
- action: aws:runShellScript
  description: Run PostgreSQL Database freeze/thaw commands
  name: run_pre_post_scripts
  precondition:
    StringEquals:
    - platformType
    - Linux
  inputs:
    runCommand:
    - |
      #!/bin/bash


 ###============================================================================###
      ### Error Codes

 ###============================================================================###
      # The following Error codes will inform Data Lifecycle Manager of the type of
 error
      # and help guide handling of the error.
      # The Error code will also be emitted via AWS Eventbridge events in the
 'cause' field.
      # 1 Pre-script failed during execution - 201
      # 2 Post-script failed during execution - 202
      # 3 Auto thaw occurred before post-script was initiated - 203
      # 4 Pre-script initiated while post-script was expected - 204
      # 5 Post-script initiated while pre-script was expected - 205
      # 6 Application not ready for pre or post-script initiation - 206


 ###============================================================================###
      ### Global variables

 ###============================================================================###
      START=$(date +%s)
```

```
    OPERATION={{ command }}
    FS_ALREADY_FROZEN_ERROR='freeze failed: Device or resource busy'
    FS_ALREADY_THAWED_ERROR='unfreeze failed: Invalid argument'
    FS_BUSY_ERROR='mount point is busy'

    # Auto thaw is a fail safe mechanism to automatically unfreeze the application
after the
    # duration specified in the global variable below. Choose the duration based
on your
    # database application's tolerance to freeze.
    export AUTO_THAW_DURATION_SECS="60"

    # Add all pre-script actions to be performed within the function below
    execute_pre_script() {
        echo "INFO: Start execution of pre-script"
        # Check if filesystem is already frozen. No error code indicates that
filesystem
        # is not currently frozen and that the pre-script can proceed with
freezing the filesystem.
        check_fs_freeze
        # Execute the DB commands to flush the DB in preparation for snapshot
        snap_db
        # Freeze the filesystem. No error code indicates that filesystem was
succefully frozen
        freeze_fs

        echo "INFO: Schedule Auto Thaw to execute in ${AUTO_THAW_DURATION_SECS}
seconds."
        $(nohup bash -c execute_schedule_auto_thaw  >/dev/null 2>&1 &)
    }

    # Add all post-script actions to be performed within the function below
    execute_post_script() {
        echo "INFO: Start execution of post-script"
        # Unfreeze the filesystem. No error code indicates that filesystem was
successfully unfrozen
        unfreeze_fs
    }

    # Execute Auto Thaw to automatically unfreeze the application after the
duration configured
    # in the AUTO_THAW_DURATION_SECS global variable.
    execute_schedule_auto_thaw() {
        sleep ${AUTO_THAW_DURATION_SECS}
```

```
        execute_post_script
    }

    # Disable Auto Thaw if it is still enabled
    execute_disable_auto_thaw() {
        echo "INFO: Attempting to disable auto thaw if enabled"
        auto_thaw_pgid=$(pgrep -f execute_schedule_auto_thaw | xargs -i ps -hp {}
-o pgid)
        if [ -n "${auto_thaw_pgid}" ]; then
            echo "INFO: execute_schedule_auto_thaw process found with pgid
${auto_thaw_pgid}"
            sudo pkill -g ${auto_thaw_pgid}
            rc=$?
            if [ ${rc} != 0 ]; then
                echo "ERROR: Unable to kill execute_schedule_auto_thaw process.
retval=${rc}"
            else
                echo "INFO: Auto Thaw  has been disabled"
            fi
        fi
    }

    # Iterate over all the mountpoints and check if filesystem is already in
freeze state.
    # Return error code 204 if any of the mount points are already frozen.
    check_fs_freeze() {
        for target in $(lsblk -nlo MOUNTPOINTS)
        do
            # Freeze of the root and boot filesystems is dangerous and pre-script
does not freeze these filesystems.
            # Hence, we will skip the root and boot mountpoints while checking if
filesystem is in freeze state.
            if [ $target == '/' ]; then continue; fi
            if [[ "$target" == *"/boot"* ]]; then continue; fi

            error_message=$(sudo mount -o remount,noatime $target 2>&1)
            # Remount will be a no-op without a error message if the filesystem is
unfrozen.
            # However, if filesystem is already frozen, remount will fail with
busy error message.
            if [ $? -ne 0 ];then
                # If the filesystem is already in frozen, return error code 204
                if [[ "$error_message" == *"$FS_BUSY_ERROR"* ]];then
```

```
                    echo "ERROR: Filesystem ${target} already frozen. Return Error
Code: 204"
                    exit 204
                fi
                # If the check filesystem freeze failed due to any reason other
than the filesystem already frozen, return 201
                echo "ERROR: Failed to check_fs_freeze on mountpoint $target due
to error - $errormessage"
                exit 201
            fi
        done
    }

    # Iterate over all the mountpoints and freeze the filesystem.
    freeze_fs() {
        for target in $(lsblk -nlo MOUNTPOINTS)
        do
            # Freeze of the root and boot filesystems is dangerous. Hence, skip
filesystem freeze
            # operations for root and boot mountpoints.
            if [ $target == '/' ]; then continue; fi
            if [[ "$target" == *"/boot"* ]]; then continue; fi
            echo "INFO: Freezing $target"
            error_message=$(sudo fsfreeze -f $target 2>&1)
            if [ $? -ne 0 ];then
                # If the filesystem is already in frozen, return error code 204
                if [[ "$error_message" == *"$FS_ALREADY_FROZEN_ERROR"* ]]; then
                    echo "ERROR: Filesystem ${target} already frozen. Return Error
Code: 204"
                    exit 204
                fi
                # If the filesystem freeze failed due to any reason other than the
filesystem already frozen, return 201
                echo "ERROR: Failed to freeze mountpoint $targetdue due to error -
$errormessage"
                exit 201
            fi
            echo "INFO: Freezing complete on $target"
        done
    }

    # Iterate over all the mountpoints and unfreeze the filesystem.
    unfreeze_fs() {
        for target in $(lsblk -nlo MOUNTPOINTS)
```

```
        do
            # Freeze of the root and boot filesystems is dangerous and pre-script
does not freeze these filesystems.
            # Hence, will skip the root and boot mountpoints during unfreeze as
well.
            if [ $target == '/' ]; then continue; fi
            if [[ "$target" == *"/boot"* ]]; then continue; fi
            echo "INFO: Thawing $target"
            error_message=$(sudo fsfreeze -u $target 2>&1)
            # Check if filesystem is already unfrozen (thawed). Return error code
204 if filesystem is already unfrozen.
            if [ $? -ne 0 ]; then
                if [[ "$error_message" == *"$FS_ALREADY_THAWED_ERROR"* ]]; then
                    echo "ERROR: Filesystem ${target} is already in thaw state.
Return Error Code: 205"
                    exit 205
                fi
                # If the filesystem unfreeze failed due to any reason other than
the filesystem already unfrozen, return 202
                echo "ERROR: Failed to unfreeze mountpoint $targetdue due to error
- $errormessage"
                exit 202
            fi
            echo "INFO: Thaw complete on $target"
        done
    }

    snap_db() {
        # Run the flush command only when PostgreSQL DB service is up and running
        sudo systemctl is-active --quiet postgresql
        if [ $? -eq 0 ]; then
            echo "INFO: Execute Postgres CHECKPOINT"
            # PostgreSQL command to flush the transactions in memory to disk
            sudo -u postgres psql -c 'CHECKPOINT;'
            # If the PostgreSQL Command did not succeed, return error code 201 to
indicate pre-script failure
            if [ $? -ne 0 ]; then
                echo "ERROR: Postgres CHECKPOINT command failed."
                exit 201
            fi
            sync
        else
            echo "INFO: PostgreSQL service is inactive. Skipping execution of
CHECKPOINT command."
```

```
        fi
    }

    export -f execute_schedule_auto_thaw
    export -f execute_post_script
    export -f unfreeze_fs

    # Debug logging for parameters passed to the SSM document
    echo "INFO: ${OPERATION} starting at $(date) with executionId:
 ${EXECUTION_ID}"

    # Based on the command parameter value execute the function that supports
    # pre-script/post-script operation
    case ${OPERATION} in
        pre-script)
            execute_pre_script
            ;;
        post-script)
            execute_post_script
            execute_disable_auto_thaw
            ;;
        dry-run)
            echo "INFO: dry-run option invoked - taking no action"
            ;;
        *)
            echo "ERROR: Invalid command parameter passed. Please use either pre-
script, post-script, dry-run."
            exit 1 # return failure
            ;;
    esac

    END=$(date +%s)
    # Debug Log for profiling the script time
    echo "INFO: ${OPERATION} completed at $(date). Total runtime: $((${END} -
 ${START})) seconds."
```

InterSystems IRIS sample document content

```
###############################################################################
# MIT License
#
# Copyright (c) 2024 InterSystems
#
```

```
# Permission is hereby granted, free of charge, to any person obtaining a copy
# of this software and associated documentation files (the "Software"), to deal
# in the Software without restriction, including without limitation the rights
# to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
# copies of the Software, and to permit persons to whom the Software is
# furnished to do so, subject to the following conditions:
#
# The above copyright notice and this permission notice shall be included in all
# copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
# IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
# FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
# AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
# LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
# OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
# SOFTWARE.
###======================================================================================###
schemaVersion: '2.2'
description: SSM Document Template for Amazon Data Lifecycle Manager Pre/Post script
 feature for InterSystems IRIS.
parameters:
  executionId:
    type: String
    default: None
    description: Specifies the unique identifier associated with a pre and/or post
 execution
    allowedPattern: ^(None|[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]
{4}-[a-fA-F0-9]{12})$
  command:
    type: String
    # Data Lifecycle Manager will trigger the pre-script and post-script actions.
 You can also use this SSM document with 'dry-run' for manual testing purposes.
    default: 'dry-run'
    description: (Required) Specifies whether pre-script and/or post-script should
 be executed.
    #The following allowedValues will allow Data Lifecycle Manager to successfully
 trigger pre and post script actions.
    allowedValues:
    - pre-script
    - post-script
    - dry-run

mainSteps:
```

```
- action: aws:runShellScript
  description: Run InterSystems IRIS Database freeze/thaw commands
  name: run_pre_post_scripts
  precondition:
    StringEquals:
    - platformType
    - Linux
  inputs:
    runCommand:
    - |
      #!/bin/bash

 ###============================================================================###
      ### Global variables

 ###============================================================================###
      DOCKER_NAME=iris
      LOGDIR=./
      EXIT_CODE=0
      OPERATION={{ command }}
      START=$(date +%s)

      # Check if Docker is installed
      # By default if Docker is present, script assumes that InterSystems IRIS is
 running in Docker
      # Leave only the else block DOCKER_EXEC line, if you run InterSystems IRIS
 non-containerised (and Docker is present).
      # Script assumes irissys user has OS auth enabled, change the OS user or
 supply login/password depending on your configuration.
      if command -v docker &> /dev/null
      then
        DOCKER_EXEC="docker exec $DOCKER_NAME"
      else
        DOCKER_EXEC="sudo -i -u irissys"
      fi


      # Add all pre-script actions to be performed within the function below
      execute_pre_script() {
        echo "INFO: Start execution of pre-script"

        # find all iris running instances
        iris_instances=$($DOCKER_EXEC iris qall 2>/dev/null | tail -n +3 | grep
 '^up' | cut -c5-  | awk '{print $1}')
```

```
        echo "`date`: Running iris instances $iris_instances"

        # Only for running instances
        for INST in $iris_instances; do

          echo "`date`: Attempting to freeze $INST"

          # Detailed instances specific log
          LOGFILE=$LOGDIR/$INST-pre_post.log

          #check Freeze status before starting
          $DOCKER_EXEC irissession $INST -U '%SYS'
 "##Class(Backup.General).IsWDSuspendedExt()"
          freeze_status=$?
          if [ $freeze_status -eq 5 ]; then
            echo "`date`:   ERROR: $INST IS already FROZEN"
            EXIT_CODE=204
          else
            echo "`date`:   $INST is not frozen"
            # Freeze
            # Docs: https://docs.intersystems.com/irislatest/csp/documatic/
%25CSP.Documatic.cls?LIBRARY=%25SYS&CLASSNAME=Backup.General#ExternalFreeze
            $DOCKER_EXEC irissession $INST -U '%SYS'
 "##Class(Backup.General).ExternalFreeze(\"$LOGFILE\",,,,,,600,,,300)"
            status=$?

            case $status in
              5) echo "`date`:   $INST IS FROZEN"
                ;;
              3) echo "`date`:   $INST FREEZE FAILED"
                EXIT_CODE=201
                ;;
              *) echo "`date`:   ERROR: Unknown status code: $status"
                EXIT_CODE=201
                ;;
            esac
            echo "`date`:   Completed freeze of $INST"
          fi
        done
        echo "`date`: Pre freeze script finished"
      }

      # Add all post-script actions to be performed within the function below
      execute_post_script() {
```

```
        echo "INFO: Start execution of post-script"

        # find all iris running instances
        iris_instances=$($DOCKER_EXEC iris qall 2>/dev/null | tail -n +3 | grep
 '^up' | cut -c5-  | awk '{print $1}')
        echo "`date`: Running iris instances $iris_instances"

        # Only for running instances
        for INST in $iris_instances; do

          echo "`date`: Attempting to thaw $INST"

          # Detailed instances specific log
          LOGFILE=$LOGDIR/$INST-pre_post.log

          #check Freeze status befor starting
          $DOCKER_EXEC irissession $INST -U '%SYS'
 "##Class(Backup.General).IsWDSuspendedExt()"
          freeze_status=$?
          if [ $freeze_status -eq 5 ]; then
            echo "`date`:  $INST is in frozen state"
            # Thaw
            # Docs: https://docs.intersystems.com/irislatest/csp/documatic/
%25CSP.Documatic.cls?LIBRARY=%25SYS&CLASSNAME=Backup.General#ExternalFreeze
            $DOCKER_EXEC irissession $INST -U%SYS
 "##Class(Backup.General).ExternalThaw(\"$LOGFILE\")"
            status=$?

            case $status in
              5) echo "`date`:   $INST IS THAWED"
                  $DOCKER_EXEC irissession $INST -U%SYS
 "##Class(Backup.General).ExternalSetHistory(\"$LOGFILE\")"
                 ;;
              3) echo "`date`:   $INST THAW FAILED"
                  EXIT_CODE=202
                 ;;
              *) echo "`date`:   ERROR: Unknown status code: $status"
                  EXIT_CODE=202
                 ;;
            esac
            echo "`date`:   Completed thaw of $INST"
          else
            echo "`date`:   ERROR: $INST IS already THAWED"
            EXIT_CODE=205
```

```
          fi
        done
        echo "`date`: Post thaw script finished"
      }

      # Debug logging for parameters passed to the SSM document
        echo "INFO: ${OPERATION} starting at $(date) with executionId:
 ${EXECUTION_ID}"

      # Based on the command parameter value execute the function that supports
      # pre-script/post-script operation
      case ${OPERATION} in
        pre-script)
          execute_pre_script
          ;;
        post-script)
          execute_post_script
            ;;
        dry-run)
          echo "INFO: dry-run option invoked - taking no action"
          ;;
        *)
          echo "ERROR: Invalid command parameter passed. Please use either pre-
script, post-script, dry-run."
          # return failure
          EXIT_CODE=1
          ;;
      esac

      END=$(date +%s)
      # Debug Log for profiling the script time
      echo "INFO: ${OPERATION} completed at $(date). Total runtime: $((${END} -
 ${START})) seconds."
      exit $EXIT_CODE
```

Untuk informasi selengkapnya, lihat GitHub repositori.

Empty document template

```
###============================================================================###
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

# Permission is hereby granted, free of charge, to any person obtaining a copy of
 this
```

```
# software and associated documentation files (the "Software"), to deal in the
 Software
# without restriction, including without limitation the rights to use, copy, modify,
# merge, publish, distribute, sublicense, and/or sell copies of the Software, and to
# permit persons to whom the Software is furnished to do so.

# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
 IMPLIED,
# INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
# PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
# HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
# OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
# SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
###############################################################################
schemaVersion: '2.2'
description: SSM Document Template for Amazon Data Lifecycle Manager Pre/Post script
 feature
parameters:
  executionId:
    type: String
    default: None
    description: (Required) Specifies the unique identifier associated with a pre
 and/or post execution
    allowedPattern: ^(None|[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]
{4}-[a-fA-F0-9]{12})$
  command:
  # Data Lifecycle Manager will trigger the pre-script and post-script actions
 during policy execution.
  # 'dry-run' option is intended for validating the document execution without
 triggering any commands
  # on the instance. The following allowedValues will allow Data Lifecycle Manager
 to successfully
  # trigger pre and post script actions.
    type: String
    default: 'dry-run'
    description: (Required) Specifies whether pre-script and/or post-script should
 be executed.
    allowedValues:
    - pre-script
    - post-script
    - dry-run

mainSteps:
- action: aws:runShellScript
```

```
 description: Run Database freeze/thaw commands
 name: run_pre_post_scripts
 precondition:
   StringEquals:
   - platformType
   - Linux
 inputs:
   runCommand:
   - |
     #!/bin/bash


###############################################################################
     ### Error Codes

###############################################################################
     # The following Error codes will inform Data Lifecycle Manager of the type of
error
     # and help guide handling of the error.
     # The Error code will also be emitted via AWS Eventbridge events in the
'cause' field.
     # 1 Pre-script failed during execution - 201
     # 2 Post-script failed during execution - 202
     # 3 Auto thaw occurred before post-script was initiated - 203
     # 4 Pre-script initiated while post-script was expected - 204
     # 5 Post-script initiated while pre-script was expected - 205
     # 6 Application not ready for pre or post-script initiation - 206


###############################################################################
     ### Global variables

###############################################################################
     START=$(date +%s)
     # For testing this script locally, replace the below with OPERATION=$1.
     OPERATION={{ command }}

     # Add all pre-script actions to be performed within the function below
     execute_pre_script() {
         echo "INFO: Start execution of pre-script"
     }

     # Add all post-script actions to be performed within the function below
     execute_post_script() {
```

```
        echo "INFO: Start execution of post-script"
    }

    # Debug logging for parameters passed to the SSM document
    echo "INFO: ${OPERATION} starting at $(date) with executionId:
 ${EXECUTION_ID}"

    # Based on the command parameter value execute the function that supports
    # pre-script/post-script operation
    case ${OPERATION} in
        pre-script)
            execute_pre_script
            ;;
        post-script)
            execute_post_script
            ;;
        dry-run)
            echo "INFO: dry-run option invoked - taking no action"
            ;;
        *)
            echo "ERROR: Invalid command parameter passed. Please use either pre-
script, post-script, dry-run."
            exit 1 # return failure
            ;;
    esac

    END=$(date +%s)
    # Debug Log for profiling the script time
    echo "INFO: ${OPERATION} completed at $(date). Total runtime: $((${END} -
 ${START})) seconds."
```

Setelah Anda memiliki konten dokumen SSM, gunakan salah satu prosedur berikut untuk membuat dokumen SSM kustom.

Console

Untuk membuat dokumen perintah SSM

1. Buka AWS Systems Manager konsol di https://console.aws.amazon.com//systems-manager/.
2. Di panel navigasi, pilih Dokumen, lalu pilih Buat dokumen, Perintah atau Sesi.
3. Untuk Nama, masukkan nama deskriptif untuk dokumen.

4. Untuk jenis Target, pilih/AWS::EC2::Instance.
5. Untuk Jenis dokumen, pilih Perintah.
6. Di bidang Konten, pilih YAML lalu tempel konten dokumen.
7. Di bagian Tanda dokumen, tambahkan tanda dengan kunci tanda `DLMScriptsAccess`, dan nilai tanda `true`.

> **Important**
>
> `DLMScriptsAccess:true`Tag diperlukan oleh kebijakan AWSDataLifecycleManagerSSMFullAccess AWS terkelola yang digunakan pada Langkah 3: Siapkan peran IAM Amazon Data Lifecycle Manager. Kebijakan menggunakan kunci syarat `aws:ResourceTag` untuk membatasi akses ke dokumen SSM yang memiliki tanda ini.

8. Pilih Buat dokumen.

AWS CLI

Untuk membuat dokumen perintah SSM

Gunakan perintah create-document. Untuk `--name`, tentukan nama deskriptif untuk dokumen. Untuk `--document-type`, tentukan `Command`. Untuk `--content`, tentukan jalur ke file .yaml dengan konten dokumen SSM. Untuk `--tags`, tentukan `"Key=DLMScriptsAccess,Value=true"`.

```
$ aws ssm create-document \
--content file://path/to/file/documentContent.yaml \
--name "document_name" \
--document-type "Command" \
--document-format YAML \
--tags "Key=DLMScriptsAccess,Value=true"
```

Langkah 3: Siapkan peran IAM Amazon Data Lifecycle Manager

> **Note**
>
> Langkah ini diperlukan jika:

- Anda membuat atau memperbarui kebijakan snapshot skrip pra/pasca yang diaktifkan yang menggunakan peran IAM kustom.
- Anda menggunakan baris perintah untuk membuat atau memperbarui kebijakan snapshot skrip pra/pasca yang diaktifkan yang menggunakan default.

Jika Anda menggunakan konsol untuk membuat atau memperbarui kebijakan snapshot berkemampuan skrip pra/posting yang menggunakan peran default untuk mengelola snapshot () AWSDataLifecycleManagerDefaultRole, lewati langkah ini. Dalam hal ini, kami secara otomatis melampirkan AWSDataLifecycleManagerSSMFullAccesskebijakan ke peran itu.

Anda harus memastikan bahwa peran IAM yang Anda gunakan untuk kebijakan memberikan izin kepada Amazon Data Lifecycle Manager untuk melakukan tindakan SSM yang diperlukan untuk menjalankan skrip pra dan pasca pada instans yang ditargetkan oleh kebijakan.

Amazon Data Lifecycle Manager menyediakan kebijakan terkelola (AWSDataLifecycleManagerSSMFullAccess) yang menyertakan izin yang diperlukan. Anda dapat melampirkan kebijakan ini ke peran IAM untuk mengelola snapshot guna memastikan bahwa kebijakan tersebut menyertakan izin.

Important

Kebijakan AWSDataLifecycleManagerSSMFullAccess terkelola menggunakan kunci `aws:ResourceTag` kondisi untuk membatasi akses ke dokumen SSM tertentu saat menggunakan skrip pra dan pasca. Untuk mengizinkan Amazon Data Lifecycle Manager mengakses dokumen SSM, Anda harus memastikan bahwa dokumen SSM Anda ditandai dengan `DLMScriptsAccess:true`.

Atau, Anda dapat membuat kebijakan kustom secara manual atau menetapkan izin yang diperlukan langsung ke peran IAM yang Anda gunakan. Anda dapat menggunakan izin yang sama yang ditentukan dalam kebijakan AWSDataLifecycleManagerSSMFullAccess terkelola, namun, kunci `aws:ResourceTag` kondisi bersifat opsional. Jika Anda memutuskan untuk tidak menyertakan kunci syarat itu, Anda tidak perlu menandai dokumen SSM Anda dengan `DLMScriptsAccess:true`.

Gunakan salah satu metode berikut untuk menambahkan AWSDataLifecycleManagerSSMFullAccesskebijakan ke peran IAM Anda.

Console

Untuk melampirkan kebijakan terkelola ke peran kustom

1. Buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi, pilih Peran.
3. Cari dan pilih peran kustom Anda untuk mengelola snapshot.
4. Pada tab Izin, pilih Tambahkan izin, Lampirkan kebijakan.
5. Cari dan pilih kebijakan AWSDataLifecycleManagerSSMFullAccessterkelola, lalu pilih Tambahkan izin.

AWS CLI

Untuk melampirkan kebijakan terkelola ke peran kustom

Gunakan perintah attach-role-policy. Untuk `---role-name`, tentukan nama peran kustom Anda. Untuk `--policy-arn`, tentukan `arn:aws:iam::aws:policy/AWSDataLifecycleManagerSSMFullAccess`.

```
$ aws iam attach-role-policy \
--policy-arn arn:aws:iam::aws:policy/AWSDataLifecycleManagerSSMFullAccess \
--role-name your_role_name
```

Langkah 4: Membuat kebijakan siklus hidup snapshot

Untuk mengotomatisasi snapshot yang konsisten dengan aplikasi, Anda harus membuat kebijakan siklus hidup snapshot yang menargetkan instans, dan mengonfigurasi skrip pra dan pasca untuk kebijakan tersebut.

Console

Untuk membuat kebijakan siklus hidup snapshot

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.

2. Di panel navigasi, pilih Elastic Block Store, Lifecycle Manager, lalu pilih Buat kebijakan siklus hidup.
3. Pada layar Pilih jenis kebijakan, pilih Kebijakan snapshot EBS, lalu pilih Berikutnya.
4. Di bagian Sumber daya target, lakukan hal berikut ini:

   a. Untuk Jenis sumber daya target, pilih `Instance`.

   b. Untuk Tanda sumber daya target, tentukan tanda sumber daya yang mengidentifikasi instans yang akan dicadangkan. Hanya sumber daya yang memiliki tanda tertentu yang akan dicadangkan.
5. Untuk peran IAM, pilih AWSDataLifecycleManagerDefaultRole(peran default untuk mengelola snapshot), atau pilih peran khusus yang Anda buat dan siapkan untuk skrip pra dan pasca.
6. Konfigurasikan jadwal dan opsi tambahan sesuai kebutuhan. Sebaiknya jadwalkan waktu pembuatan snapshot untuk periode waktu yang sesuai dengan beban kerja Anda, seperti selama jendela pemeliharaan.

   Untuk SAP HANA, kami menyarankan Anda mengaktifkan pemulihan snapshot cepat.

   > **Note**
   >
   > Jika Anda mengaktifkan jadwal untuk Cadangan VSS, Anda tidak dapat mengaktifkan Kecualikan volume data tertentu atau Salin tanda dari sumber.
7. Di bagian Skrip pra dan pasca, pilih Aktifkan skrip pra dan pasca, lalu lakukan hal berikut, bergantung pada beban kerja Anda:

   - Untuk membuat snapshot yang konsisten dengan aplikasi dari aplikasi Windows Anda, pilih Cadangan VSS.
   - Untuk membuat snapshot yang konsisten dengan aplikasi dari beban kerja SAP HANA Anda, pilih SAP HANA.
   - Untuk membuat snapshot yang konsisten dengan aplikasi dari semua database dan beban kerja lainnya, termasuk database MySQL, PostgreSQL, atau IRIS yang dikelola sendiri, menggunakan dokumen SSM kustom, pilih dokumen SSM khusus. InterSystems

     1. Untuk Opsi otomatisasi, pilih Skrip pra dan pasca.
     2. Untuk Dokumen SSM, pilih dokumen SSM yang Anda siapkan.
8. Bergantung pada opsi yang Anda pilih, konfigurasikan opsi tambahan berikut:

- Batas waktu skrip — (Khusus dokumen SSM kustom) Periode batas waktu sebelum Amazon Data Lifecycle Manager menggagalkan upaya menjalankan skrip jika belum selesai. Jika skrip tidak selesai dalam periode batas waktu, Amazon Data Lifecycle Manager menggagalkan upaya tersebut. Periode batas waktu berlaku untuk skrip pra dan pasca secara individual. Periode batas waktu minimum dan default-nya adalah 10 detik. Dan periode batas waktu maksimumnya adalah 120 detik.
- Coba lagi skrip yang gagal — Pilih opsi ini untuk mencoba lagi skrip yang tidak selesai dalam periode batas waktu. Jika skrip pra gagal, Amazon Data Lifecycle Manager akan mencoba ulang seluruh proses pembuatan snapshot, termasuk menjalankan skrip pra dan pasca. Jika skrip pasca gagal, Amazon Data Lifecycle Manager mencoba ulang skrip pasca saja; dalam hal ini, skrip pra akan selesai dan snapshot mungkin telah dibuat.
- Default ke snapshot crash-consistent — Pilih opsi ini ke default ke snapshot crash-consistent jika skrip pra gagal dijalankan. Ini adalah perilaku pembuatan snapshot default untuk Amazon Data Lifecycle Manager jika skrip pra dan pasca tidak diaktifkan. Jika Anda mengaktifkan percobaan ulang, Amazon Data Lifecycle Manager akan default ke snapshot crash-consistent hanya setelah semua upaya percobaan ulang habis. Jika skrip pra gagal dan Anda tidak menetapkan default ke snapshot crash-consistent, Amazon Data Lifecycle Manager tidak akan membuat snapshot untuk instans selama jadwal berjalan.

> **Note**
>
> Jika Anda membuat snapshot untuk SAP HANA, Anda mungkin ingin menonaktifkan opsi ini. Snapshot crash-consistent dari beban kerja SAP HANA tidak dapat dipulihkan dengan cara yang sama.

9. Pilih Buat kebijakan default.

> **Note**
>
> Jika Anda mendapatkan kesalahan `Role with name AWSDataLifecycleManagerDefaultRole already exists`, lihat Pemecahan Masalah untuk informasi selengkapnya.

AWS CLI

Untuk membuat kebijakan siklus hidup snapshot

Gunakan perintah create-lifecycle-policy, dan sertakan parameter `Scripts` dalam `CreateRule`. Untuk informasi selengkapnya tentang parameter, lihat Referensi API Amazon Data Lifecycle Manager.

```
$ aws dlm create-lifecycle-policy \
--description "policy_description" \
--state ENABLED \
--execution-role-arn iam_role_arn \
--policy-details file://policyDetails.json
```

Di mana `policyDetails.json` termasuk salah satu hal berikut, tergantung pada kasus penggunaan Anda:

- Cadangan VSS

```
{
    "PolicyType": "EBS_SNAPSHOT_MANAGEMENT",
    "ResourceTypes": [
        "INSTANCE"
    ],
    "TargetTags": [{
        "Key": "tag_key",
        "Value": "tag_value"
    }],
    "Schedules": [{
        "Name": "schedule_name",
        "CreateRule": {
            "CronExpression": "cron_for_creation_frequency",
            "Scripts": [{
                "ExecutionHandler":"AWS_VSS_BACKUP",
                "ExecuteOperationOnScriptFailure":true|false,
                "MaximumRetryCount":retries (0-3)
            }]
        },
        "RetainRule": {
            "Count": retention_count
        }
    }]
}
```

- Pencadangan SAP HANA

```
{
    "PolicyType": "EBS_SNAPSHOT_MANAGEMENT",
    "ResourceTypes": [
        "INSTANCE"
    ],
    "TargetTags": [{
        "Key": "tag_key",
        "Value": "tag_value"
    }],
    "Schedules": [{
        "Name": "schedule_name",
        "CreateRule": {
            "CronExpression": "cron_for_creation_frequency",
            "Scripts": [{
                "Stages": ["PRE","POST"],
                "ExecutionHandlerService":"AWS_SYSTEMS_MANAGER",
                "ExecutionHandler":"AWSSystemsManagerSAP-
CreateDLMSnapshotForSAPHANA",
                "ExecuteOperationOnScriptFailure":true|false,
                "ExecutionTimeout":timeout_in_seconds (10-120),
                "MaximumRetryCount":retries (0-3)
            }]
        },
        "RetainRule": {
            "Count": retention_count
        }
    }]
}
```

- Dokumen SSM Kustom

```
{
    "PolicyType": "EBS_SNAPSHOT_MANAGEMENT",
    "ResourceTypes": [
        "INSTANCE"
    ],
    "TargetTags": [{
        "Key": "tag_key",
        "Value": "tag_value"
    }],
    "Schedules": [{
        "Name": "schedule_name",
```

```
        "CreateRule": {
            "CronExpression": "cron_for_creation_frequency",
            "Scripts": [{
                "Stages": ["PRE","POST"],
                "ExecutionHandlerService":"AWS_SYSTEMS_MANAGER",
                "ExecutionHandler":"ssm_document_name|arn",
                "ExecuteOperationOnScriptFailure":true|false,
                "ExecutionTimeout":timeout_in_seconds (10-120),
                "MaximumRetryCount":retries (0-3)
            }]
        },
        "RetainRule": {
            "Count": retention_count
        }
    }]
}
```

Pertimbangan untuk Pencadangan VSS dengan Amazon Data Lifecycle Manager

Dengan Amazon Data Lifecycle Manager, Anda dapat mencadangkan dan memulihkan aplikasi Windows berkemampuan VSS (Volume Shadow Copy Service) yang berjalan di instans Amazon EC2. Jika aplikasi memiliki penulis VSS yang terdaftar dengan Windows VSS, Amazon Data Lifecycle Manager membuat snapshot yang akan bersifat konsisten aplikasi untuk aplikasi itu.

> **Note**
>
> Amazon Data Lifecycle Manager saat ini mendukung snapshot sumber daya yang konsisten aplikasi yang berjalan di Amazon EC2 saja, khusus untuk skenario pencadangan di mana data aplikasi dapat dipulihkan dengan mengganti instans yang ada dengan instans baru yang dibuat dari cadangan. Tidak semua tipe instans atau aplikasi didukung untuk pencadangan VSS. Untuk informasi lebih lanjut, lihat Apa itu AWS VSS? di Panduan Pengguna Amazon EC2.

Tipe instans yang didukung

Tipe instans Amazon EC2 berikut tidak didukung untuk pencadangan VSS. Jika kebijakan Anda menargetkan salah satu tipe instans ini, Amazon Data Lifecycle Manager mungkin masih membuat cadangan VSS, tetapi snapshot mungkin tidak ditandai dengan tanda sistem yang diperlukan. Tanpa

tanda ini, snapshot tidak akan dikelola oleh Amazon Data Lifecycle Manager setelah pembuatan. Anda mungkin perlu menghapus snapshot tersebut secara manual.

- T3: | `t3.nano` `t3.micro`
- T3a: | `t3a.nano` `t3a.micro`
- T2: | `t2.nano` `t2.micro`

Tanggung jawab bersama untuk snapshot yang konsisten dengan aplikasi

Anda harus memastikan bahwa:

- Agen SSM diinstal, up-to-date, dan berjalan pada instance target Anda
- Systems Manager memiliki izin untuk melakukan tindakan yang diperlukan pada instans target
- Amazon Data Lifecycle Manager memiliki izin untuk melakukan tindakan Systems Manager yang diperlukan untuk menjalankan skrip pra dan pasca pada instans target.
- Untuk beban kerja kustom, seperti database MySQL, PostgreSQL, atau InterSystems IRIS yang dikelola sendiri, dokumen SSM yang Anda gunakan menyertakan tindakan yang benar dan diperlukan untuk membekukan, membilas, dan mencairkan I/O untuk konfigurasi database Anda.
- Waktu pembuatan snapshot selaras dengan jadwal beban kerja Anda. Misalnya, cobalah untuk menjadwalkan pembuatan snapshot selama jendela pemeliharaan terjadwal.

Amazon Data Lifecycle Manager memastikan bahwa:

- Pembuatan snapshot dimulai dalam waktu 60 menit dari waktu pembuatan snapshot yang dijadwalkan.
- Skrip pra dijalankan sebelum pembuatan snapshot dimulai.
- Skrip pasca berjalan setelah skrip pra berhasil dan pembuatan snapshot telah dimulai. Amazon Data Lifecycle Manager menjalankan skrip pasca hanya jika skrip pra berhasil. Jika skrip pra gagal, Amazon Data Lifecycle Manager tidak akan menjalankan skrip pasca.
- Snapshot ditandai dengan tanda yang sesuai pada pembuatan.
- CloudWatch metrik dan peristiwa dipancarkan ketika skrip dimulai, dan ketika mereka gagal atau berhasil.

Kasus penggunaan lain untuk skrip pra dan pasca

Selain menggunakan skrip pra dan pasca untuk mengotomatiskan snapshot yang konsisten dengan aplikasi, Anda dapat menggunakan skrip pra dan pasca bersama-sama, atau secara individual, untuk mengotomatiskan tugas administratif lainnya sebelum atau sesudah pembuatan snapshot. Misalnya:

- Menggunakan skrip pra untuk menerapkan patch sebelum membuat snapshot. Ini dapat membantu Anda membuat snapshot setelah menerapkan pembaruan perangkat lunak mingguan atau bulanan reguler Anda.

  > **Note**
  >
  > Jika Anda memilih untuk menjalankan skrip pra saja, Tetapkan default ke snapshot crash-consistent diaktifkan secara default.

- Menggunakan skrip pasca untuk menerapkan patch sebelum membuat snapshot. Ini dapat membantu Anda membuat snapshot setelah menerapkan pembaruan perangkat lunak mingguan atau bulanan reguler Anda.

Memulai untuk kasus penggunaan lainnya

Bagian ini menjelaskan langkah-langkah yang perlu Anda lakukan saat menggunakan skrip pra dan/atau pasca untuk kasus penggunaan selain snapshot yang konsisten dengan aplikasi.

Langkah 1: Menyiapkan instans target

Untuk mempersiapkan instans target Anda untuk skrip pra dan/atau pasca

1. Instal SSM Agent pada instans target Anda, jika belum diinstal. Jika SSM Agent sudah diinstal pada instans target Anda, lewati langkah ini.
   - (Instans Linux) Menginstal SSM Agent secara manual di instans Amazon EC2 untuk Linux
   - (Instans Windows) Menginstal SSM Agent secara manual di instans Amazon EC2 untuk Windows
2. Pastikan SSM Agent berjalan. Untuk informasi selengkapnya, lihat Memeriksa status SSM Agent dan memulai agen.
3. Siapkan Systems Manager untuk instans Amazon EC2. Untuk informasi selengkapnya, lihat Menyiapkan Systems Manager untuk instans Amazon EC2 di Panduan Pengguna AWS Systems Manager .

Langkah 2: Siapkan Dokumen SSM

Anda harus membuat dokumen perintah SSM yang menyertakan skrip pra dan/atau pasca dengan perintah yang ingin Anda jalankan.

Anda dapat membuat dokumen SSM menggunakan templat dokumen SSM kosong di bawah ini dan menambahkan perintah pra dan pasca Anda di bagian dokumen yang sesuai.

⚠ Perhatikan hal-hal berikut:

- Anda bertanggung jawab untuk memastikan bahwa dokumen SSM melakukan tindakan yang benar dan diperlukan untuk beban kerja Anda.
- Dokumen SSM harus menyertakan bidang wajib untuk `allowedValues`, termasuk, `pre-script`, `post-script`, dan `dry-run`. Amazon Data Lifecycle Manager akan menjalankan perintah pada instans Anda berdasarkan konten bagian tersebut. Jika dokumen SSM Anda tidak memiliki bagian tersebut, Amazon Data Lifecycle Manager akan memperlakukannya sebagai eksekusi yang gagal.

```
###======================================================================###
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.

# Permission is hereby granted, free of charge, to any person obtaining a copy of this
# software and associated documentation files (the "Software"), to deal in the Software
# without restriction, including without limitation the rights to use, copy, modify,
# merge, publish, distribute, sublicense, and/or sell copies of the Software, and to
# permit persons to whom the Software is furnished to do so.

# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
# INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
# PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
# HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
# OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
# SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
###======================================================================###
schemaVersion: '2.2'
description: SSM Document Template for Amazon Data Lifecycle Manager Pre/Post script
 feature
parameters:
  executionId:
    type: String
```

```
    default: None
    description: (Required) Specifies the unique identifier associated with a pre and/
or post execution
    allowedPattern: ^(None|[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-
[a-fA-F0-9]{12})$
  command:
  # Data Lifecycle Manager will trigger the pre-script and post-script actions during
 policy execution.
  # 'dry-run' option is intended for validating the document execution without
 triggering any commands
  # on the instance. The following allowedValues will allow Data Lifecycle Manager to
 successfully
  # trigger pre and post script actions.
    type: String
    default: 'dry-run'
    description: (Required) Specifies whether pre-script and/or post-script should be
 executed.
    allowedValues:
    - pre-script
    - post-script
    - dry-run

mainSteps:
- action: aws:runShellScript
  description: Run Database freeze/thaw commands
  name: run_pre_post_scripts
  precondition:
    StringEquals:
    - platformType
    - Linux
  inputs:
    runCommand:
    - |
      #!/bin/bash


 ###========================================================================###
      ### Error Codes

 ###========================================================================###
      # The following Error codes will inform Data Lifecycle Manager of the type of
 error
      # and help guide handling of the error.
```

```
    # The Error code will also be emitted via AWS Eventbridge events in the 'cause'
field.
    # 1 Pre-script failed during execution - 201
    # 2 Post-script failed during execution - 202
    # 3 Auto thaw occurred before post-script was initiated - 203
    # 4 Pre-script initiated while post-script was expected - 204
    # 5 Post-script initiated while pre-script was expected - 205
    # 6 Application not ready for pre or post-script initiation - 206


###============================================================================###
    ### Global variables

###============================================================================###
    START=$(date +%s)
    # For testing this script locally, replace the below with OPERATION=$1.
    OPERATION={{ command }}

    # Add all pre-script actions to be performed within the function below
    execute_pre_script() {
        echo "INFO: Start execution of pre-script"
    }

    # Add all post-script actions to be performed within the function below
    execute_post_script() {
        echo "INFO: Start execution of post-script"
    }

    # Debug logging for parameters passed to the SSM document
    echo "INFO: ${OPERATION} starting at $(date) with executionId: ${EXECUTION_ID}"

    # Based on the command parameter value execute the function that supports
    # pre-script/post-script operation
    case ${OPERATION} in
        pre-script)
            execute_pre_script
            ;;
        post-script)
            execute_post_script
            ;;
        dry-run)
            echo "INFO: dry-run option invoked - taking no action"
            ;;
        *)
```

```
              echo "ERROR: Invalid command parameter passed. Please use either pre-
script, post-script, dry-run."
              exit 1 # return failure
              ;;
      esac

      END=$(date +%s)
      # Debug Log for profiling the script time
      echo "INFO: ${OPERATION} completed at $(date). Total runtime: $((${END} -
 ${START})) seconds."
```

Langkah 3: Siapkan peran IAM Amazon Data Lifecycle Manager

> **Note**
>
> Langkah ini diperlukan jika:
>
> - Anda membuat atau memperbarui kebijakan snapshot skrip pra/pasca yang diaktifkan yang menggunakan peran IAM kustom.
> - Anda menggunakan baris perintah untuk membuat atau memperbarui kebijakan snapshot skrip pra/pasca yang diaktifkan yang menggunakan default.
>
> Jika Anda menggunakan konsol untuk membuat atau memperbarui kebijakan snapshot berkemampuan skrip pra/posting yang menggunakan peran default untuk mengelola snapshot () AWSDataLifecycleManagerDefaultRole, lewati langkah ini. Dalam hal ini, kami secara otomatis melampirkan AWSDataLifecycleManagerSSMFullAccesskebijakan ke peran itu.

Anda harus memastikan bahwa peran IAM yang Anda gunakan untuk kebijakan memberikan izin Amazon Data Lifecycle Manager untuk melakukan tindakan SSM yang diperlukan untuk menjalankan skrip pra dan pasca pada instans yang ditargetkan oleh kebijakan.

Amazon Data Lifecycle Manager menyediakan kebijakan terkelola (AWSDataLifecycleManagerSSMFullAccess) yang menyertakan izin yang diperlukan. Anda dapat melampirkan kebijakan ini ke peran IAM untuk mengelola snapshot guna memastikan bahwa kebijakan tersebut menyertakan izin.

> **Important**
>
> Kebijakan AWSDataLifecycleManagerSSMFullAccess terkelola menggunakan kunci `aws:ResourceTag` kondisi untuk membatasi akses ke dokumen SSM tertentu saat menggunakan skrip pra dan pasca. Untuk mengizinkan Amazon Data Lifecycle Manager mengakses dokumen SSM, Anda harus memastikan bahwa dokumen SSM Anda ditandai dengan `DLMScriptsAccess:true`.

Atau, Anda dapat membuat kebijakan kustom secara manual atau menetapkan izin yang diperlukan langsung ke peran IAM yang Anda gunakan. Anda dapat menggunakan izin yang sama yang ditentukan dalam kebijakan AWSDataLifecycleManagerSSMFullAccess terkelola, namun, kunci `aws:ResourceTag` kondisi bersifat opsional. Jika Anda memutuskan untuk tidak menyertakan kunci syarat itu, Anda tidak perlu menandai dokumen SSM Anda. `DLMScriptsAccess:true`

Gunakan salah satu metode berikut untuk menambahkan AWSDataLifecycleManagerSSMFullAccesskebijakan ke peran IAM Anda.

Console

Untuk melampirkan kebijakan terkelola ke peran kustom

1. Buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi, pilih Peran.
3. Cari dan pilih peran kustom Anda untuk mengelola snapshot.
4. Pada tab Izin, pilih Tambahkan izin, Lampirkan kebijakan.
5. Cari dan pilih kebijakan AWSDataLifecycleManagerSSMFullAccessterkelola, lalu pilih Tambahkan izin.

AWS CLI

Untuk melampirkan kebijakan terkelola ke peran kustom

Gunakan perintah attach-role-policy. Untuk `---role-name`, tentukan nama peran kustom Anda. Untuk `--policy-arn`, tentukan `arn:aws:iam::aws:policy/AWSDataLifecycleManagerSSMFullAccess`.

```
$ aws iam attach-role-policy \
```

```
--policy-arn arn:aws:iam::aws:policy/AWSDataLifecycleManagerSSMFullAccess \
--role-name your_role_name
```

Membuat kebijakan siklus hidup snapshot

Console

Untuk membuat kebijakan siklus hidup snapshot

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Elastic Block Store, Lifecycle Manager, lalu pilih Buat kebijakan siklus hidup.
3. Pada layar Pilih jenis kebijakan, pilih Kebijakan snapshot EBS, lalu pilih Berikutnya.
4. Di bagian Sumber daya target, lakukan hal berikut ini:
   a. Untuk Jenis sumber daya target, pilih `Instance`.
   b. Untuk Tanda sumber daya target, tentukan tanda sumber daya yang mengidentifikasi instans yang akan dicadangkan. Hanya sumber daya yang memiliki tanda tertentu yang akan dicadangkan.
5. Untuk peran IAM, pilih AWSDataLifecycleManagerDefaultRole(peran default untuk mengelola snapshot), atau pilih peran khusus yang Anda buat dan siapkan untuk skrip pra dan pasca.
6. Konfigurasikan jadwal dan opsi tambahan sesuai kebutuhan. Sebaiknya jadwalkan waktu pembuatan snapshot untuk periode waktu yang sesuai dengan beban kerja Anda, seperti selama jendela pemeliharaan.
7. Di bagian Skrip pra dan pasca, pilih Aktifkan skrip pra dan pasca, lalu lakukan hal berikut:
   a. Pilih Dokumen SSM Kustom.
   b. Untuk Opsi otomatis, pilih opsi yang cocok dengan skrip yang ingin Anda jalankan.
   c. Untuk Dokumen SSM, pilih dokumen SSM yang Anda siapkan.
8. Konfigurasikan opsi tambahan berikut jika diperlukan:
   - Batas waktu skrip - Periode batas waktu setelah Amazon Data Lifecycle Manager menggagalkan upaya menjalankan skrip jika belum selesai. Jika skrip tidak selesai dalam periode batas waktu, Amazon Data Lifecycle Manager menggagalkan upaya tersebut. Periode batas waktu berlaku untuk skrip pra dan pasca secara individual. Periode batas

waktu minimum dan default-nya adalah 10 detik. Dan periode batas waktu maksimumnya adalah 120 detik.

- Coba lagi skrip yang gagal — Pilih opsi ini untuk mencoba lagi skrip yang tidak selesai dalam periode batas waktu. Jika skrip pra gagal, Amazon Data Lifecycle Manager akan mencoba ulang seluruh proses pembuatan snapshot, termasuk menjalankan skrip pra dan pasca. Jika skrip pasca gagal, Amazon Data Lifecycle Manager mencoba ulang skrip pasca saja; dalam hal ini, skrip pra akan selesai dan snapshot mungkin telah dibuat.
- Default ke snapshot crash-consistent — Pilih opsi ini ke default ke snapshot crash-consistent jika skrip pra gagal dijalankan. Ini adalah perilaku pembuatan snapshot default untuk Amazon Data Lifecycle Manager jika skrip pra dan pasca tidak diaktifkan. Jika Anda mengaktifkan percobaan ulang, Amazon Data Lifecycle Manager akan default ke snapshot crash-consistent hanya setelah semua upaya percobaan ulang habis. Jika skrip pra gagal dan Anda tidak menetapkan default ke snapshot crash-consistent, Amazon Data Lifecycle Manager tidak akan membuat snapshot untuk instans selama jadwal berjalan.

9. Pilih Buat kebijakan default.

> **Note**
>
> Jika Anda mendapatkan kesalahan `Role with name AWSDataLifecycleManagerDefaultRole already exists`, lihat Pemecahan Masalah untuk informasi selengkapnya.

AWS CLI

Untuk membuat kebijakan siklus hidup snapshot

Gunakan perintah create-lifecycle-policy, dan sertakan parameter `Scripts` dalam `CreateRule`. Untuk informasi selengkapnya tentang parameter, lihat Referensi API Amazon Data Lifecycle Manager.

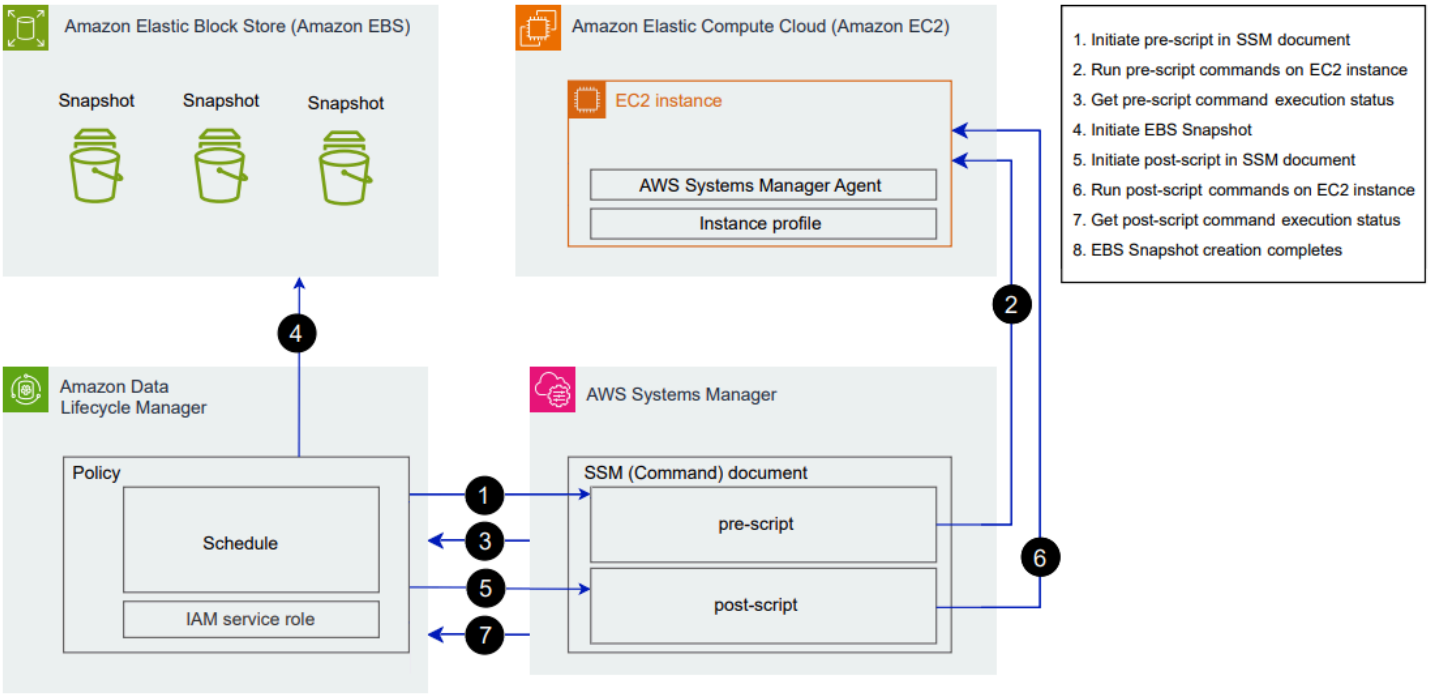
```
$ aws dlm create-lifecycle-policy \
--description "policy_description" \
--state ENABLED \
--execution-role-arn iam_role_arn \
--policy-details file://policyDetails.json
```

Di mana `policyDetails.json` termasuk yang berikut.

```
{
    "PolicyType": "EBS_SNAPSHOT_MANAGEMENT",
    "ResourceTypes": [
        "INSTANCE"
    ],
    "TargetTags": [{
        "Key": "tag_key",
        "Value": "tag_value"
    }],
    "Schedules": [{
        "Name": "schedule_name",
        "CreateRule": {
            "CronExpression": "cron_for_creation_frequency",
            "Scripts": [{
                "Stages": ["PRE" | "POST" | "PRE","POST"],
                "ExecutionHandlerService":"AWS_SYSTEMS_MANAGER",
                "ExecutionHandler":"ssm_document_name|arn",
                "ExecuteOperationOnScriptFailure":true|false,
                "ExecutionTimeout":timeout_in_seconds (10-120),
                "MaximumRetryCount":retries (0-3)
            }]
        },
        "RetainRule": {
            "Count": retention_count
        }
    }]
}
```

Cara kerja skrip pra dan pasca

Gambar berikut menunjukkan alur proses untuk skrip pra dan pasca saat menggunakan dokumen SSM kustom. Hal ini tidak berlaku untuk Pencadangan VSS.



Pada waktu pembuatan snapshot yang dijadwalkan, tindakan berikut dan interaksi lintas layanan terjadi.

1. Amazon Data Lifecycle Manager memulai tindakan skrip pra dengan memanggil dokumen SSM dan meneruskan parameter `pre-script`.

   > **Note**
   >
   > Langkah 1 hingga 3 hanya terjadi jika Anda menjalankan skrip pra. Jika Anda menjalankan skrip pasca saja, langkah 1 hingga 3 dilewati.

2. Systems Manager mengirimkan perintah pra skrip ke SSM Agent yang berjalan pada instans target. SSM Agent menjalankan perintah pada instans, dan mengirimkan informasi status kembali ke Systems Manager.

   Misalnya, jika dokumen SSM digunakan untuk membuat snapshot yang konsisten dengan aplikasi, skrip pra mungkin membekukan dan membersihkan I/O untuk memastikan bahwa semua data buffer ditulis ke volume sebelum snapshot diambil.

3. Systems Manager mengirimkan pembaruan status perintah skrip pra ke Amazon Data Lifecycle Manager. Jika skrip pra gagal, Amazon Data Lifecycle Manager mengambil salah satu tindakan berikut, tergantung pada cara Anda mengonfigurasi opsi skrip pra dan pasca:

Percobaan ulang	Default ke snapshot crash-consistent	Tindakan
Diaktifkan dengan percobaan ulang yang tersisa	Aktif	Coba lagi skrip sampai berhasil atau percobaan ulang habis
Habis tanpa penyelesaian yang berhasil	Aktif	Buat snapshot crash-consistent, dan jangan jalankan skrip pasca.
Diaktifkan dengan percobaan ulang yang tersisa	Nonaktif	Coba lagi skrip sampai berhasil atau percobaan ulang habis
Habis tanpa penyelesaian yang berhasil	Nonaktif	Lewati pembuatan snapshot untuk instans target, dan jangan jalankan skrip pasca.
Nonaktif	Aktif	Buat snapshot crash-consistent, dan jangan jalankan skrip pasca.
Nonaktif	Nonaktif	Lewati pembuatan snapshot untuk instans target, dan jangan jalankan skrip pasca.

4. Amazon Data Lifecycle Manager memulai pembuatan snapshot.
5. Amazon Data Lifecycle Manager memulai tindakan pasca skrip dengan memanggil dokumen SSM dan meneruskan parameter `post-script`.

> **Note**
>
> Langkah 5 hingga 7 hanya terjadi jika Anda menjalankan skrip pra. Jika Anda menjalankan skrip pasca saja, langkah 1 hingga 3 dilewati.

6. Systems Manager mengirimkan perintah post script ke SSM Agent yang berjalan pada instans target. SSM Agent menjalankan perintah pada instans, dan mengirimkan informasi status kembali ke Systems Manager.

   Misalnya, jika dokumen SSM mengaktifkan snapshot yang konsisten dengan aplikasi, skrip pasca ini mungkin mencairkan I/O untuk memastikan bahwa basis data Anda melanjutkan operasi I/O normal setelah snapshot diambil.

7. Jika Anda menjalankan skrip pasca dan Systems Manager menunjukkan bahwa itu selesai dengan sukses, proses selesai.

   Jika skrip pasca gagal, Amazon Data Lifecycle Manager mengambil salah satu tindakan berikut, tergantung pada cara Anda mengonfigurasi opsi skrip pra dan pasca:

Percobaan ulang	Tindakan
Diaktifkan dengan percobaan ulang yang tersisa	Coba lagi skrip sampai berhasil atau percobaan ulang habis
Lelah tanpa sukses	Lewati skrip pasca
Nonaktif	Lewati skrip pasca

   Perlu diingat bahwa jika skrip pasca gagal, skrip pra (jika diaktifkan) akan berhasil diselesaikan, dan snapshot mungkin telah dibuat. Anda mungkin perlu mengambil tindakan lebih lanjut pada instans untuk memastikan bahwa itu beroperasi seperti yang diharapkan. Misalnya jika skrip pra berhenti dan membersihkan I/O, tetapi skrip pasca gagal mencairkan I/O, Anda mungkin perlu mengonfigurasi basis data Anda untuk mencairkan I/O secara otomatis atau Anda perlu mencairkan I/O secara manual.

8. Proses pembuatan snapshot mungkin selesai setelah skrip pasca selesai. Waktu yang dibutuhkan untuk menyelesaikan snapshot tergantung pada ukuran snapshot.

Mengidentifikasi snapshot yang dibuat dengan skrip pra dan pasca

Amazon Data Lifecycle Manager secara otomatis menetapkan tanda sistem berikut ke snapshot yang dibuat dengan skrip pra dan pasca.

- Nilai: `aws:dlm:pre-script`; Kunci: `SUCCESS|FAILED`

  Nilai tanda `SUCCESS` menunjukkan bahwa skrip pra berhasil dieksekusi. Nilai tanda `FAILED` menunjukkan bahwa skrip pra gagal dieksekusi.
- Nilai: `aws:dlm:post-script`; Kunci: `SUCCESS|FAILED`

  Nilai tanda `SUCCESS` menunjukkan bahwa skrip pasca berhasil dieksekusi. Nilai tanda `FAILED` menunjukkan bahwa skrip pasca gagal dieksekusi.

Untuk dokumen SSM kustom dan cadangan SAP HANA, Anda dapat menyimpulkan pembuatan snapshot yang konsisten aplikasi yang berhasil jika snapshot ditandai dengan `aws:dlm:pre-script:SUCCESS` dan `aws:dlm:post-script:SUCCESS`.

Selain itu, snapshot konsisten aplikasi yang dibuat menggunakan cadangan VSS secara otomatis ditandai dengan:

- Nilai: `AppConsistent tag`; Kunci: `true|false`

  Nilai tanda `true` menunjukkan bahwa pencadangan VSS berhasil dan bahwa snapshot bersifat konsisten aplikasi. Nilai tanda `false` menunjukkan bahwa pencadangan VSS gagal dan bahwa snapshot tidak bersifat konsisten aplikasi.

Memantau eksekusi skrip pra dan pasca

CloudWatch Metrik Amazon

Amazon Data Lifecycle Manager menerbitkan CloudWatch metrik berikut saat skrip pra dan pasca gagal dan berhasil dan saat pencadangan VSS gagal dan berhasil.

- `PreScriptStarted`
- `PreScriptCompleted`
- `PreScriptFailed`
- `PostScriptStarted`

- `PostScriptCompleted`
- `PostScriptFailed`
- `VSSBackupStarted`
- `VSSBackupCompleted`
- `VSSBackupFailed`

Untuk informasi selengkapnya, lihat Pantau kebijakan Anda menggunakan Amazon CloudWatch.

Amazon EventBridge

Amazon Data Lifecycle Manager memancarkan peristiwa EventBridge Amazon berikut saat skrip pra atau pasca dimulai, berhasil, atau gagal

- `DLM Pre Post Script Notification`

Untuk informasi selengkapnya, lihat Pantau kebijakan Anda menggunakan CloudWatch Acara.

Mengotomatiskan siklus hidup AMI

Prosedur berikut ini menunjukkan cara menggunakan Amazon Data Lifecycle Manager untuk mengotomatiskan siklus hidup AMI yang didukung EBS.

Topik

- Membuat kebijakan siklus hidup AMI
- Pertimbangan untuk kebijakan siklus hidup AMI
- Sumber daya tambahan

Membuat kebijakan siklus hidup AMI

Gunakan salah satu prosedur berikut ini untuk membuat kebijakan siklus hidup AMI.

Console

Untuk membuat kebijakan AMI

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.

2. Di panel navigasi, pilih Elastic Block Store, Lifecycle Manager, lalu pilih Buat kebijakan siklus hidup.
3. Pada layar Pilih jenis kebijakan, pilih kebijakan AMI yang didukung EBS, lalu pilih Berikutnya.
4. Di bagian Sumber daya target, untuk Tanda sumber daya target, pilih tanda sumber daya yang mengidentifikasi volume atau instans yang akan dicadangkan. Kebijakan hanya mencadangkan sumber daya yang memiliki tanda tertentu kunci dan nilai pasangan.
5. Untuk Deskripsi, masukkan deskripsi singkat untuk kebijakan tersebut.
6. Untuk peran IAM, pilih peran IAM yang memiliki izin untuk mengelola AMI dan snapshot dan untuk mendeskripsikan instans. Untuk menggunakan peran default yang disediakan oleh Amazon Data Lifecycle Manager, pilih Peran default. Atau, untuk menggunakan peran IAM kustom yang Anda buat sebelumnya, pilih Pilih peran lain, lalu pilih peran yang akan digunakan.
7. Untuk Tanda kebijakan, tambahkan tanda yang akan diterapkan pada kebijakan siklus hidup. Anda dapat menggunakan tanda ini untuk mengidentifikasi dan mengategorikan kebijakan Anda.
8. Untuk Status kebijakan setelah pembuatan, pilih Aktifkan kebijakan untuk memulai pelaksanaan kebijakan pada waktu yang dijadwalkan berikutnya, atau Nonaktifkan kebijakan untuk mencegah agar kebijakan tidak berjalan. Jika Anda tidak mengaktifkan kebijakan sekarang, kebijakan tidak akan mulai membuat AMI sampai Anda mengaktifkannya secara manual setelah pembuatan.
9. Di bagian Boot ulang instans, menunjukkan apakah instans harus di-boot ulang sebelum pembuatan AMI. Untuk mencegah instans yang ditargetkan di-boot ulang, pilih Tidak. Memilih Tidak dapat menyebabkan masalah konsistensi data. Untuk melakukan boot ulang instans sebelum pembuatan AMI, pilih Ya. Memilih ini memastikan konsistensi data tetapi dapat menyebabkan beberapa instans yang ditargetkan di-boot ulang secara bersamaan.
10. Pilih Berikutnya.
11. Pada layar Konfigurasikan jadwal, konfigurasikan jadwal kebijakan. Kebijakan dapat memiliki sampai maksimal 4 jadwal. Jadwal 1 bersifat wajib. Jadwal 2, 3, dan 4 bersifat opsional. Untuk setiap jadwal kebijakan yang Anda tambahkan, lakukan hal berikut:

    a. Dalam bagian Detail jadwal, lakukan hal berikut:

        i. Untuk Nama jadwal, tentukan nama deskriptif untuk jadwal.

        ii. Untuk Frekuensi dan bidang terkait, konfigurasikan interval antara kebijakan yang dijalankan.

Anda dapat mengonfigurasikan kebijakan yang berjalan sesuai jadwal harian, mingguan, bulanan, atau tahunan. Atau, pilih Ekspresi cron kustom untuk menentukan interval hingga satu tahun. Untuk informasi selengkapnya, lihat Ekspresi cron di Panduan Pengguna CloudWatch Acara Amazon.

iii. Untuk Dimulai pada, tentukan waktu di mana pelaksanaan kebijakan dijadwalkan untuk dimulai. Pelaksanaan kebijakan pertama dimulai dengan satu jam setelah waktu yang Anda jadwalkan. Waktu harus dimasukkan dalam format `hh:mm` UTC.

iv. Untuk Jenis retensi, tentukan kebijakan penyimpanan untuk AMI yang dibuat berdasarkan jadwal.

Anda dapat mempertahankan snapshot berdasarkan jumlah atau usianya.

Untuk retensi berbasis jumlah, rentangnya adalah 1 sampai `1000`. Setelah jumlah maksimum tercapai, AMI tertua dibatalkan pendaftarannya saat AMI baru dibuat.

Untuk retensi berbasis usia, rentangnya adalah 1 hari ke `100` tahun. Setelah masa retensi masing-masing berakhir, AMI akan dibatalkan pendaftarannya.

> **Note**
>
> Semua jadwal harus memiliki jenis retensi yang sama. Anda dapat menentukan jenis retensi hanya untuk Jadwal 1. Jadwal 2, 3, dan 4 mewarisi jenis retensi dari Jadwal 1. Setiap jadwal dapat memiliki jumlah atau periode retensi sendiri.

b. Konfigurasikan penandaan untuk AMI.

Di bagian Penandaan, lakukan hal berikut ini:

i. Untuk menyalin semua tanda yang ditentukan pengguna dari instans sumber ke AMI yang dibuat oleh jadwal, pilih Salin tanda dari sumber.

ii. Secara default, AMI yang dibuat oleh jadwal secara otomatis ditandai dengan ID dari instans sumber. Untuk mencegah penandaan otomatis ini terjadi, untuk Tanda variabel, hapus petak `instance-id:$(instance-id)`.

iii. Untuk menentukan tanda tambahan untuk ditetapkan ke AMI yang dibuat oleh jadwal ini, pilih Tambahkan tanda.

c. Konfigurasikan penghentian AMI.

Untuk menghentikan AMI ketika seharusnya tidak digunakan lagi, di bagian penghentian AMI, pilih Aktifkan penghentian AMI untuk jadwal ini, lalu tentukan aturan penghentian AMI. Aturan penghentian AMI menentukan kapan AMI akan dihentikan.

Jika jadwal menggunakan retensi AMI berbasis hitungan, Anda harus menentukan jumlah AMI tertua yang akan dihentikan. Jumlah penghentian harus kurang dari atau sama dengan jumlah retensi AMI jadwal, dan tidak boleh lebih dari 1000. Misalnya, jika jadwal dikonfigurasi untuk mempertahankan maksimum 5 AMI, Anda dapat mengonfigurasi jadwal untuk menghentikan hingga 5 AMI tertua yang lama.

Jika jadwal menggunakan retensi AMI berdasarkan usia, Anda harus menentukan periode setelah AMI tidak digunakan lagi. Jumlah penghentian harus kurang dari atau sama dengan periode retensi AMI jadwal, dan tidak boleh lebih dari 10 tahun (120 bulan, 520 minggu, atau 3650 hari). Misalnya, jika jadwal dikonfigurasi untuk mempertahankan AMI selama 10 hari, Anda dapat mengonfigurasi AMI yang dijadwalkan untuk menghentikan AMI setelah periode hingga 10 hari setelah pembuatan.

d. Konfigurasikan salinan lintas-Wilayah.

Untuk menyalin AMI yang dibuat oleh jadwal ke Wilayah lain, di bagian Salinan Lintas-Wilayah , pilih Aktifkan salinan lintas-Wilayah. Anda dapat menyalin AMI hingga tiga Wilayah tambahan di akun Anda. Anda harus menentukan aturan salinan lintas wilayah terpisah untuk setiap Wilayah tujuan.

Untuk setiap tujuan Wilayah, Anda dapat menentukan sebagai berikut:

- Kebijakan penyimpanan untuk salinan AMI. Saat periode retensi berakhir, salinan di Wilayah tujuan secara otomatis dideregistrasi.
- Status enkripsi untuk salinan AMI. Jika snapshot sumber dienkripsi, atau jika enkripsi secara default diaktifkan, snapshot yang disalin dienkripsi. Jika sumber AMI tidak dienkripsi dan enkripsi secara default dinonaktifkan, Anda dapat mengaktifkan enkripsi secara opsional. Jika Anda tidak menentukan kunci KMS, AMI dienkripsi menggunakan kunci KMS default untuk enkripsi EBS di setiap Wilayah tujuan. Jika Anda menentukan kunci KMS untuk Wilayah tujuan, peran IAM yang dipilih harus memiliki akses ke kunci KMS.
- Aturan penghentian untuk salinan AMI. Ketika periode penghentian berakhir, salinan AMI secara otomatis tidak digunakan lagi. Periode penghentian harus kurang dari atau sama dengan periode penyimpanan salinan, dan tidak boleh lebih dari 10 tahun.

- Apakah akan menyalin semua tanda atau tidak ada tanda dari sumber AMI.

> **Note**
>
> Jangan melebihi jumlah salinan AMI bersamaan per Wilayah.

e. Untuk menambahkan jadwal tambahan, pilih Tambahkan jadwal lain, yang terletak di bagian atas layar. Untuk setiap jadwal tambahan, lengkapi bidang seperti yang dijelaskan sebelumnya dalam topik ini.

f. Setelah Anda menambahkan jadwal yang diperlukan, pilih Tinjau kebijakan.

12. Tinjau ringkasan kebijakan, lalu pilih Buat kebijakan.

> **Note**
>
> Jika Anda mendapatkan kesalahan `Role with name AWSDataLifecycleManagerDefaultRoleForAMIManagement already exists`, lihat Pemecahan Masalah untuk informasi selengkapnya.

Command line

Gunakan perintah create-lifecycle-policy untuk membuat kebijakan siklus hidup AMI. Untuk `PolicyType`, tentukan `IMAGE_MANAGEMENT`.

> **Note**
>
> Untuk menyederhanakan sintaksis, contoh berikut menggunakan file JSON, `policyDetails.json`, yang mencakup detail kebijakan.

Contoh 1: Retensi berbasis usia dan penghentian AMI

Contoh ini menciptakan kebijakan siklus hidup AMI yang menciptakan AMI dari semua instans yang memiliki kunci tanda `purpose` dengan nilai `production` tanpa melakukan boot ulang instans yang ditargetkan. Kebijakan ini mencakup satu jadwal yang membuat AMI setiap hari pada pukul `01:00` UTC. Kebijakan mempertahankan AMI selama 2 hari dan menghentikannya setelah 1 hari. Hal ini juga menyalin tanda dari instans sumber ke AMI yang membuatnya.

```
aws dlm create-lifecycle-policy \
    --description "My AMI policy" \
    --state ENABLED \
    --execution-role-arn
 arn:aws:iam::12345678910:role/AWSDataLifecycleManagerDefaultRoleForAMIManagement \
    --policy-details file://policyDetails.json
```

Berikut ini adalah contoh file `policyDetails.json`.

```
{
    "PolicyType": "IMAGE_MANAGEMENT",
    "ResourceTypes": [
        "INSTANCE"
    ],
    "TargetTags": [{
        "Key": "purpose",
        "Value": "production"
    }],
    "Schedules": [{
            "Name": "DailyAMIs",
            "TagsToAdd": [{
                "Key": "type",
                "Value": "myDailyAMI"
            }],
            "CreateRule": {
                "Interval": 24,
                "IntervalUnit": "HOURS",
                "Times": [
                    "01:00"
                ]
            },
            RetainRule":{
                "Interval" : 2,
                "IntervalUnit" : "DAYS"
            },
            DeprecateRule": {
                "Interval" : 1,
                "IntervalUnit" : "DAYS"
            },
            "CopyTags": true
        }
    ],
    "Parameters" : {
```

```
        "NoReboot":true
    }
}
```

Jika permintaan berhasil, perintah mengembalikan ID kebijakan yang baru dibuat. Berikut ini adalah output contoh.

```
{
   "PolicyId": "policy-9876543210abcdef0"
}
```

Contoh 2: Retensi berbasis hitungan dan penghentian AMI dengan salinan Lintas wilayah

Contoh ini menciptakan kebijakan siklus hidup AMI yang membuat AMI dari semua instans yang memiliki kunci tanda `purpose` dengan nilai `production` dan melakukan boot ulang instans target. Kebijakan ini mencakup satu jadwal yang membuat AMI setiap 6 jam mulai pukul `17:30` UTC. Kebijakan mempertahankan 3 AMI dan secara otomatis menghentikan 2 AMI tertua. Kebijakan ini juga memiliki aturan menyalin lintas wilayah yang menyalin AMI ke `us-east-1`, mempertahankan 2 salinan AMI, dan secara otomatis menghentikan AMI yang tertua.

```
aws dlm create-lifecycle-policy \
    --description "My AMI policy" \
    --state ENABLED \
    --execution-role-arn
 arn:aws:iam::12345678910:role/AWSDataLifecycleManagerDefaultRoleForAMIManagement \
    --policy-details file://policyDetails.json
```

Berikut ini adalah contoh file `policyDetails.json`.

```
{
    "PolicyType": "IMAGE_MANAGEMENT",
    "ResourceTypes" : [
        "INSTANCE"
    ],
    "TargetTags": [{
        "Key":"purpose",
        "Value":"production"
    }],
    "Parameters" : {
          "NoReboot": true
    },
```

```
    "Schedules" : [{
        "Name" : "Schedule1",
        "CopyTags": true,
        "CreateRule" : {
            "Interval": 6,
            "IntervalUnit": "HOURS",
            "Times" : ["17:30"]
        },
        "RetainRule":{
            "Count" : 3
        },
        "DeprecateRule":{
            "Count" : 2
        },
        "CrossRegionCopyRules": [{
            "TargetRegion": "us-east-1",
            "Encrypted": true,
            "RetainRule":{
                "IntervalUnit": "DAYS",
                "Interval": 2
            },
            "DeprecateRule":{
                "IntervalUnit": "DAYS",
                "Interval": 1
            },
            "CopyTags": true
        }]
    }]
}
```

Pertimbangan untuk kebijakan siklus hidup AMI

Pertimbangan umum berikut ini berlaku untuk pembuatan kebijakan siklus hidup AMI:

- Kebijakan siklus hidup AMI hanya menargetkan instans atau volume yang berada di Wilayah yang sama dengan kebijakan.
- Operasi pembuatan AMI pertama dimulai dalam waktu satu jam setelah waktu mulai yang ditentukan. Operasi pembuatan AMI selanjutnya dimulai dalam waktu yang dijadwalkan selama satu jam.
- Saat Amazon Data Lifecycle Manager membatalkan pendaftaran AMI, secara otomatis akan menghapusnya untuk mendukung snapshot.

- Tanda sumber daya peka huruf besar dan kecil.
- Jika Anda menghapus tanda target dari instans yang ditargetkan oleh kebijakan, Amazon Data Lifecycle Manager tidak lagi mengelola AMI yang ada dalam standar; Anda harus menghapusnya secara manual jika tidak diperlukan lagi.
- Anda dapat membuat beberapa kebijakan untuk mencadangkan volume atau instans. Misalnya, jika instans memiliki dua tanda, tandai A adalah target kebijakan A untuk membuat AMI setiap 12 jam, dan tanda B adalah target kebijakan B untuk membuat AMI setiap 24 jam, Amazon Data Lifecycle Manager menciptakan AMI sesuai jadwal untuk kedua kebijakan. Atau, Anda dapat mencapai hasil yang sama dengan membuat satu kebijakan yang memiliki beberapa jadwal. Misalnya, Anda dapat membuat kebijakan tunggal yang hanya menargetkan tanda A, dan menentukan dua jadwal — satu untuk setiap 12 jam dan satu untuk setiap 24 jam.
- Volume baru yang dipasang ke instans target setelah kebijakan dibuat secara otomatis disertakan dalam pencadangan saat pelaksanaan kebijakan berikutnya. Semua volume yang dilampirkan pada instans saat pelaksanaan kebijakan disertakan.
- Jika Anda membuat kebijakan dengan jadwal berbasis kron kustom yang dikonfigurasi untuk membuat hanya satu snapshot, kebijakan tidak akan secara otomatis menghapus AMI ketika ambang retensi tercapai. Anda harus menghapus AMI secara manual jika tidak lagi diperlukan.
- Jika Anda membuat kebijakan berbasis usia dengan periode penyimpanan lebih pendek dari frekuensi pembuatan, Amazon Data Lifecycle Manager akan selalu mempertahankan AMI terakhir hingga snapshot berikutnya dibuat. Misalnya, jika kebijakan berbasis usia membuat satu AMI setiap bulan dengan periode retensi tujuh hari, Amazon Data Lifecycle Manager akan mempertahankan setiap AMI selama satu bulan, meskipun periode retensi adalah tujuh hari.
- Untuk kebijakan berbasis jumlah, Amazon Data Lifecycle Manager selalu membuat AMI sesuai dengan frekuensi pembuatan sebelum mencoba membatalkan pendaftaran AMI tertua sesuai dengan kebijakan retensi.
- Diperlukan beberapa jam untuk berhasil membatalkan pendaftaran AMI dan menghapus snapshot dukungan terkait. Jika Amazon Data Lifecycle Manager membuat AMI berikutnya sebelum AMI yang dibuat sebelumnya berhasil dihapus, Anda dapat mempertahankan sementara jumlah AMI yang lebih besar dari jumlah retensi Anda.

Pertimbangan berikut berlaku untuk mengakhiri instans yang ditargetkan oleh kebijakan:

- Jika Anda menghentikan instans yang ditargetkan oleh kebijakan dengan jadwal penyimpanan berbasis hitungan, kebijakan tersebut tidak lagi mengelola AMI yang sebelumnya dibuat dari instans yang dihentikan. Anda harus menghapus snapshot secara manual jika tidak lagi diperlukan.

- Jika Anda menghentikan instans yang ditargetkan oleh kebijakan dengan jadwal penyimpanan berbasis usia, kebijakan tersebut terus membatalkan pendaftaran sampai AMI yang sebelumnya dibuat dari instans yang dihentikan pada jadwal yang ditentukan, hingga, tetapi tidak termasuk, AMI terakhir. Anda harus menghapus AMI secara manual jika tidak lagi diperlukan.

Pertimbangan berikut berlaku untuk kebijakan AMI dan penghentian AMI:

- Jika Anda meningkatkan jumlah penghentian AMI untuk jadwal dengan retensi berbasis hitungan, perubahan akan diterapkan ke semua AMI (yang ada dan baru) yang dibuat oleh jadwal.
- Jika Anda meningkatkan periode penghentian AMI untuk jadwal dengan retensi berbasis usia, perubahan hanya akan diterapkan pada AMI baru. AMI yang ada tidak terpengaruh.
- Jika Anda menghapus aturan penghentian AMI dari jadwal, Amazon Data Lifecycle Manager tidak akan membatalkan penghentian AMI yang sebelumnya tidak digunakan lagi oleh jadwal tersebut.
- Jika mengurangi jumlah atau periode jadwal penghentian AMI, Amazon Data Lifecycle Manager tidak akan membatalkan penghentian AMI yang sebelumnya tidak digunakan lagi oleh jadwal tersebut.
- Jika Anda menghentikan AMI yang dibuat oleh kebijakan AMI secara manual, Amazon Data Lifecycle Manager tidak akan mengganti penghentian tersebut.
- Jika Anda membatalkan penghentian AMI yang sebelumnya dihentikan oleh kebijakan AMI secara manual, Amazon Data Lifecycle Manager tidak akan mengganti penghentian tersebut.
- Jika AMI dibuat oleh beberapa jadwal yang bertentangan, dan satu atau beberapa jadwal tersebut tidak memiliki aturan penghentian AMI, Amazon Data Lifecycle Manager tidak akan menghentikan AMI tersebut.
- Jika AMI dibuat oleh beberapa jadwal yang bertentangan, dan satu atau beberapa jadwal tersebut tidak memiliki aturan penghentian AMI, Amazon Data Lifecycle Manager akan menggunakan aturan penghentian yang menghasilkan tanggal penghentian terbaru.

Pertimbangan berikut berlaku untuk kebijakan AMI dan Recycle Bin:

- Jika Amazon Data Lifecycle Manager membatalkan pendaftaran AMI dan mengirimkannya ke Keranjang Sampah saat ambang retensi kebijakan tercapai, dan Anda memulihkan AMI dari Keranjang Sampah secara manual, Anda harus membatalkan pendaftaran AMI tersebut secara manual saat tidak diperlukan lagi. Amazon Data Lifecycle Manager tidak akan lagi mengelola AMI.
- Jika Anda membatalkan pendaftaran AMI yang dibuat oleh kebijakan secara manual, dan AMI berada di Keranjang Sampah saat ambang retensi kebijakan tercapai, Amazon Data Lifecycle

Manager tidak akan membatalkan pendaftaran AMI. Amazon Data Lifecycle Manager tidak mengelola AMI saat berada di Keranjang Sampah.

Jika snapshot dipulihkan dari Keranjang Sampah sebelum ambang retensi kebijakan tercapai, Amazon Data Lifecycle Manager akan membatalkan pendaftaran AMI tersebut saat ambang retensi kebijakan tercapai.

Jika AMI dipulihkan dari Keranjang Sampah setelah ambang batas retensi kebijakan tercapai, Amazon Data Lifecycle Manager tidak akan lagi membatalkan pendaftaran AMI tersebut. Anda harus menghapus snapshot AMI secara manual saat tidak lagi diperlukan.

Pertimbangan umum berikut ini berlaku untuk kebijakan AMI dalam status kesalahan:

- Untuk kebijakan dengan jadwal retensi berbasis usia, AMI yang ditetapkan berakhir saat kebijakan berada dalam status `error` akan dipertahankan tanpa batas. Anda harus membatalkan pendaftaran AMI secara manual. Saat Anda mengaktifkan ulang kebijakan, Amazon Data Lifecycle Manager akan melanjutkan pembatalan pendaftaran AMI karena periode retensinya berakhir.
- Untuk kebijakan dengan jadwal retensi berbasis jumlah, kebijakan berhenti membuat dan membatalkan pendaftaran AMI saat berada dalam status `error`. Saat Anda mengaktifkan kembali kebijakan, Amazon Data Lifecycle Manager akan melanjutkan pembuatan AMI, dan melanjutkan pembatalan pendaftaran AMI saat ambang retensi terpenuhi.

Pertimbangan berikut berlaku untuk kebijakan AMI dan menonaktifkan AMI:

- Jika Anda menonaktifkan AMI yang dibuat oleh Amazon Data Lifecycle Manager, dan AMI dinonaktifkan saat ambang retensi tercapai, Amazon Data Lifecycle Manager akan membatalkan pendaftaran AMI dan menghapus snapshot terkait.
- Jika Anda menonaktifkan AMI yang dibuat oleh Amazon Data Lifecycle Manager dan mengarsipkan snapshot terkait secara manual, dan snapshot tersebut diarsipkan saat ambang retensi terpenuhi, Amazon Data Lifecycle Manager tidak akan menghapus snapshot tersebut dan tidak akan lagi mengelolanya.

Pertimbangan berikut berlaku untuk kebijakan AMI dan perlindungan deregistrasi AMI:

- Jika Anda mengaktifkan perlindungan deregistrasi secara manual untuk AMI yang dibuat oleh Amazon Data Lifecycle Manager, dan masih diaktifkan saat ambang retensi AMI tercapai, Amazon

Data Lifecycle Manager tidak lagi mengelola AMI tersebut. Anda harus membatalkan pendaftaran AMI secara manual dan menghapus snapshot yang mendasarinya jika tidak lagi diperlukan.

Sumber daya tambahan

Untuk informasi selengkapnya, lihat Mengotomatiskan snapshot Amazon EBS dan manajemen AMI menggunakan blog penyimpanan Amazon Data AWS Lifecycle Manager.

Mengotomatiskan salinan snapshot lintas akun

Mengotomatisasi salinan snapshot lintas akun memungkinkan Anda untuk menyalin snapshot Amazon EBS Anda ke Wilayah tertentu dalam akun terisolasi dan mengenkripsi snapshot tersebut dengan kunci enkripsi. Hal ini memungkinkan Anda melindungi diri dari kehilangan data jika akun Anda disusupi.

Mengotomatisasi salinan snapshot lintas akun melibatkan dua akun:

- Akun sumber—Akun sumber adalah akun yang membuat dan berbagi snapshot dengan akun target. Di akun ini, Anda harus membuat kebijakan snapshot EBS yang membuat snapshot pada interval yang ditetapkan dan kemudian membagikannya dengan akun lain. AWS
- Akun target—Akun target adalah akun dengan akun tujuan tempat snapshot dibagikan, dan itu adalah akun yang membuat salinan dari snapshot bersama. Dalam akun ini, Anda harus membuat salinan lintas akun kebijakan peristiwa yang secara otomatis menyalin snapshot yang dibagi dengan snapshot tersebut oleh satu atau beberapa akun sumber tertentu.

Topik

Membuat kebijakan salinan snapshot lintas akun

Untuk mempersiapkan akun sumber dan target untuk menyalin snapshot lintas akun, Anda perlu melakukan langkah-langkah berikut:

Langkah 1: Membuat kebijakan snapshot EBS (Akun sumber)

Di akun sumber, buat kebijakan snapshot EBS yang akan membuat snapshot dan membagikannya dengan akun target yang diperlukan.

Saat membuat kebijakan, pastikan Anda mengaktifkan berbagi lintas akun dan menentukan AWS akun target untuk berbagi snapshot. Ini adalah akun yang akan digunakan untuk membagikan snapshot. Jika Anda berbagi snapshot terenkripsi, Anda harus memberikan izin akun target yang dipilih untuk menggunakan kunci KMS yang digunakan untuk mengenkripsi volume sumber. Untuk informasi selengkapnya, lihat Langkah 2: Bagikan kunci yang dikelola pelanggan (Akun sumber).

> **Note**
>
> Anda hanya dapat berbagi snapshot yang tidak dienkripsi atau yang dienkripsi menggunakan kunci yang dikelola pelanggan. Anda tidak dapat berbagi snapshot yang dienkripsi dengan kunci KMS enkripsi EBS default. Jika Anda berbagi snapshot terenkripsi, kemudian Anda juga harus berbagi kunci KMS yang digunakan untuk mengenkripsi volume sumber dengan akun target. Untuk informasi selengkapnya, lihat Mengizinkan pengguna di akun lain untuk menggunakan kunci KMS di Panduan Developer AWS Key Management Service .

Untuk informasi selengkapnya tentang cara membuat kebijakan snapshot EBS, lihat Mengotomatiskan siklus hidup snapshot.

Gunakan salah satu metode berikut untuk membuat kebijakan snapshot EBS.

Langkah 2: Bagikan kunci yang dikelola pelanggan (Akun sumber)

Jika Anda berbagi snapshot terenkripsi, Anda harus memberikan izin kepada peran IAM dan akun AWS target (yang Anda pilih di langkah sebelumnya) untuk menggunakan kunci yang dikelola pelanggan yang digunakan untuk mengenkripsi volume sumber.

> **Note**
>
> Lakukan langkah ini hanya jika Anda berbagi snapshot terenkripsi. Jika Anda berbagi snapshot yang tidak dienkripsi, lewati langkah ini.

Console

1. Buka AWS KMS konsol di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Di panel navigasi, pilih Kunci yang dikelola pelanggan, lalu pilih kunci KMS yang Anda butuhkan untuk berbagi dengan akun target.

   Catat ARN kunci KMS, Anda akan membutuhkan ini nanti.
4. Pada tab Kebijakan kunci, gulir ke bawah ke bagian Pengguna kunci. Pilih Tambahkan, masukkan nama peran IAM yang Anda pilih di langkah sebelumnya, kemudian pilih Tambahkan.
5. Pada tab Kebijakan kunci, gulir ke bawah ke bagian Akun AWS lainnya. Pilih Tambahkan AWS akun lain, lalu tambahkan semua AWS akun target yang Anda pilih untuk berbagi snapshot pada langkah sebelumnya.
6. Pilih Simpan perubahan.

Command line

Gunakan perintah get-key-policy untuk mengambil kebijakan kunci yang saat ini dilampirkan pada kunci KMS.

Misalnya, perintah berikut mengambil kebijakan kunci untuk kunci KMS dengan ID `9d5e2b3d-e410-4a27-a958-19e220d83a1e` dan menulisnya ke sebuah file bernama `snapshotKey.json`.

```
$ aws kms get-key-policy \
    --policy-name default \
    --key-id 9d5e2b3d-e410-4a27-a958-19e220d83a1e \
    --query Policy \
    --output text > snapshotKey.json
```

Buka kebijakan kunci menggunakan editor teks pilihan Anda. Menambahkan ARN peran IAM yang Anda tentukan ketika Anda membuat kebijakan snapshot dan ARN akun target yang digunakan untuk berbagi kunci KMS.

Sebagai contoh, dalam kebijakan berikut, kami menambahkan ARN peran IAM default, dan ARN akun root untuk akun target `222222222222`.

> **Tip**
>
> Untuk mengikuti prinsip hak akses paling rendah, jangan biarkan akses penuh ke `kms:CreateGrant`. Sebagai gantinya, gunakan tombol `kms:GrantIsForAWSResource` kondisi untuk memungkinkan pengguna membuat hibah pada kunci KMS hanya ketika hibah dibuat atas nama pengguna oleh AWS layanan, seperti yang ditunjukkan pada contoh berikut.

```
{
    "Sid" : "Allow use of the key",
    "Effect" : "Allow",
    "Principal" : {
        "AWS" : [
            "arn:aws:iam::111111111111:role/service-role/
AWSDataLifecycleManagerDefaultRole",
            "arn:aws:iam::222222222222:root"
        ]
    },
    "Action" : [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
    ],
    "Resource" : "*"
},
{
    "Sid" : "Allow attachment of persistent resources",
    "Effect" : "Allow",
    "Principal" : {
        "AWS" : [
            "arn:aws:iam::111111111111:role/service-role/
AWSDataLifecycleManagerDefaultRole",
            "arn:aws:iam::222222222222:root"
        ]
    },
    "Action" : [
        "kms:CreateGrant",
        "kms:ListGrants",
        "kms:RevokeGrant"
```

```
    ],
    "Resource" : "*",
    "Condition" : {
        "Bool" : {
          "kms:GrantIsForAWSResource" : "true"
        }
    }
}
```

Simpan dan tutup file . Kemudian gunakan perintah put-key-policy untuk melampirkan kebijakan kunci yang diperbarui untuk bukti kunci KMS.

```
$ aws kms put-key-policy \
    --policy-name default \
    --key-id 9d5e2b3d-e410-4a27-a958-19e220d83a1e \
    --policy file://snapshotKey.json
```

Langkah 3: Buat kebijakan peristiwa penyalinan lintas akun (Akun target)

Di akun target, Anda harus membuat kebijakan peristiwa penyalinan lintas akun yang secara otomatis akan menyalin snapshot yang dibagikan oleh akun sumber yang diperlukan.

Kebijakan ini hanya berjalan di akun target ketika salah satu akun sumber tertentu berbagi snapshot dengan akun.

Gunakan salah satu metode berikut untuk membuat peristiwa penyalinan lintas akun.

Console

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Elastic Block Store, Lifecycle Manager, lalu pilih Buat kebijakan siklus hidup.
3. Pada layar Pilih jenis kebijakan, pilih Kebijakan peristiwa penyalinan lintas akun, lalu pilih Berikutnya.
4. Untuk Deskripsi kebijakan, masukkan deskripsi singkat untuk kebijakan tersebut.
5. Untuk Tanda kebijakan, tambahkan tanda untuk diterapkan ke kebijakan siklus hidup. Anda dapat menggunakan tanda ini untuk mengidentifikasi dan mengategorikan kebijakan Anda.

6. Di bagian Pengaturan peristiwa, tentukan peristiwa berbagi snapshot yang akan menyebabkan kebijakan berjalan. Lakukan hal berikut:

   a. Untuk Berbagi akun, tentukan AWS akun sumber tempat Anda ingin menyalin snapshot bersama. Pilih Tambah akun, masukkan ID AWS akun 12 digit, lalu pilih Tambah.

   b. Untuk Filter berdasarkan deskripsi, masukkan deskripsi snapshot yang diperlukan menggunakan ekspresi reguler. Hanya snapshot yang dibagikan oleh akun sumber tertentu dan memiliki deskripsi yang cocok dengan filter tertentu yang disalin oleh kebijakan. Untuk informasi selengkapnya, lihat Tentukan filter deskripsi snapshot.

7. Untuk peran IAM, pilih peran IAM yang memiliki izin untuk melakukan tindakan penyalinan snapshot. Untuk menggunakan peran default yang disediakan oleh Amazon Data Lifecycle Manager, pilih Peran default. Atau, untuk menggunakan peran IAM kustom yang Anda buat sebelumnya, pilih Pilih peran lain, lalu pilih peran yang akan digunakan.

   Jika Anda menyalin snapshot terenkripsi, Anda harus memberikan izin peran IAM yang dipilih untuk menggunakan kunci enkripsi KMS yang digunakan untuk mengenkripsi volume sumber. Demikian pula, jika Anda mengenkripsi snapshot di Wilayah tujuan menggunakan kunci KMS yang berbeda, Anda harus memberikan izin kepada peran IAM untuk menggunakan kunci KMS tujuan. Untuk informasi selengkapnya, lihat Langkah 4: Memungkinkan peran IAM untuk menggunakan kunci KMS yang diperlukan (Akun target).

8. Di bagian Salin tindakan, tentukan tindakan penyalinan snapshot yang harus dilakukan kebijakan saat diaktifkan. Kebijakan ini dapat menyalin snapshot hingga ke tiga Wilayah. Anda harus menentukan aturan salinan lintas wilayah terpisah untuk setiap Wilayah tujuan. Untuk setiap jadwal kebijakan yang Anda tambahkan, lakukan hal berikut:

   a. Untuk Nama, masukkan nama deskriptif untuk tindakan penyalinan.

   b. Untuk Wilayah target, pilih Wilayah untuk menyalin snapshot.

   c. Untuk Kedaluwarsa, tentukan berapa lama untuk mempertahankan salinan snapshot di Wilayah target setelah pembuatan.

   d. Untuk mengenkripsi salinan snapshot, untuk Enkripsi, pilih Aktifkan enkripsi. Jika snapshot sumber dienkripsi, atau jika enkripsi secara default diaktifkan untuk akun Anda, salinan snapshot selalu dienkripsi, meskipun Anda mengaktifkan enkripsi di sini. Jika snapshot sumber tidak dienkripsi dan enkripsi secara default tidak diaktifkan untuk akun Anda, Anda dapat memilih untuk mengaktifkan atau menonaktifkan enkripsi. Jika Anda tidak mengaktifkan enkripsi, tetapi tidak menentukan kunci KMS, snapshot dienkripsi menggunakan kunci KMS enkripsi default untuk setiap Wilayah tujuan. Jika

Anda menentukan kunci KMS untuk Wilayah tujuan, Anda harus memiliki akses ke kunci KMS.

9. Untuk menambahkan tindakan penyalinan snapshot tambahan, pilih Tambahkan Wilayah baru.
10. Untuk Status kebijakan setelah pembuatan, pilih Aktifkan kebijakan untuk memulai pelaksanaan kebijakan pada waktu yang dijadwalkan berikutnya, atau Nonaktifkan kebijakan untuk mencegah agar kebijakan tidak berjalan. Jika Anda tidak mengaktifkan kebijakan sekarang, kebijakan tidak akan mulai membuat snapshot sampai Anda mengaktifkannya secara manual setelah pembuatan.
11. Pilih Buat kebijakan.

Command line

Gunakan perintah create-lifecycle-policy untuk membuat kebijakan. Untuk membuat kebijakan peristiwa salinan lintas akun, untuk `PolicyType`, tentukan `EVENT_BASED_POLICY`.

Sebagai contoh, perintah berikut membuat kebijakan menyalin peristiwa lintas akun di akun target `222222222222`. Kebijakan menyalin snapshot yang dibagi oleh akun sumber `111111111111`. Kebijakan menyalin snapshot ke `sa-east-1` dan `eu-west-2`. Snapshot yang disalin ke `sa-east-1` tidak dienkripsi dan dipertahankan selama 3 hari. Snapshot yang disalin ke `eu-west-2` dienkripsi menggunakan kunci KMS `8af79514-350d-4c52-bac8-8985e84171c7` dan dipertahankan selama 1 bulan. Kebijakan menggunakan peran IAM default.

```
$ aws dlm create-lifecycle-policy \
    --description "Copy policy" \
    --state ENABLED \
    --execution-role-arn arn:aws:iam::222222222222:role/service-role/
AWSDataLifecycleManagerDefaultRole \
    --policy-details file://policyDetails.json
```

Berikut ini menunjukkan isi file `policyDetails.json`.

```
{
    "PolicyType" : "EVENT_BASED_POLICY",
    "EventSource" : {
        "Type" : "MANAGED_CWE",
        "Parameters": {
            "EventType" : "shareSnapshot",
            "SnapshotOwner": ["111111111111"]
```

```
        }
    },
    "Actions" : [{
        "Name" :"Copy Snapshot to Sao Paulo and London",
        "CrossRegionCopy" : [{
            "Target" : "sa-east-1",
             "EncryptionConfiguration" : {
                 "Encrypted" : false
             },
             "RetainRule" : {
             "Interval" : 3,
            "IntervalUnit" : "DAYS"
            }
        },
        {
            "Target" : "eu-west-2",
            "EncryptionConfiguration" : {
                 "Encrypted" : true,
                 "CmkArn" : "arn:aws:kms:eu-
west-2:222222222222:key/8af79514-350d-4c52-bac8-8985e84171c7"
            },
            "RetainRule" : {
                "Interval" : 1,
                "IntervalUnit" : "MONTHS"
            }
        }]
    }]
}
```

Jika permintaan berhasil, perintah mengembalikan ID kebijakan yang baru dibuat. Berikut ini adalah output contoh.

```
{
    "PolicyId": "policy-9876543210abcdef0"
}
```

Langkah 4: Memungkinkan peran IAM untuk menggunakan kunci KMS yang diperlukan (Akun target)

Jika Anda menyalin snapshot terenkripsi, Anda harus memberikan izin kepada peran IAM (yang Anda pilih di langkah sebelumnya) untuk menggunakan kunci yang dikelola pelanggan yang digunakan untuk mengenkripsi volume sumber.

> **Note**
>
> Hanya lakukan langkah ini jika Anda menyalin snapshot terenkripsi. Jika Anda menyalin snapshot yang tidak dienkripsi, lewati langkah ini.

Gunakan salah satu metode berikut untuk menambahkan kebijakan yang diperlukan peran IAM.

Console

1. Buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi, pilih Peran. Cari dan pilih peran IAM yang Anda pilih saat Anda membuat kebijakan peristiwa salinan lintas akun pada langkah sebelumnya. Jika Anda memilih untuk menggunakan peran default, peran tersebut diberi nama AWSDataLifecycleManagerDefaultRole.
3. Pilih Tambahkan kebijakan inline kemudian pilih tab JSON.
4. Ganti kebijakan yang ada dengan yang berikut ini, dan tentukan ARN kunci KMS yang digunakan untuk mengenkripsi volume sumber dan yang dibagikan dengan Anda oleh akun sumber di Langkah 2.

   > **Note**
   >
   > Jika Anda menyalin dari beberapa akun sumber, Anda harus menentukan ARN kunci KMS yang sesuai dari setiap akun sumber.

   Pada contoh berikut, kebijakan memberikan izin peran IAM untuk menggunakan kunci KMS `1234abcd-12ab-34cd-56ef-1234567890ab`, yang dibagikan oleh akun sumber `111111111111`, dan kunci KMS`4567dcba-23ab-34cd-56ef-0987654321yz`, yang ada di akun target `222222222222`.

   > **Tip**
   >
   > Untuk mengikuti prinsip hak akses paling rendah, jangan biarkan akses penuh ke `kms:CreateGrant`. Sebagai gantinya, gunakan tombol `kms:GrantIsForAWSResource` kondisi untuk memungkinkan pengguna membuat

hibah pada kunci KMS hanya ketika hibah dibuat atas nama pengguna oleh AWS layanan, seperti yang ditunjukkan pada contoh berikut.

```
{
   "Version": "2012-10-17",
   "Statement": [
       {
           "Effect": "Allow",
           "Action": [
               "kms:RevokeGrant",
               "kms:CreateGrant",
               "kms:ListGrants"
           ],
           "Resource": [
               "arn:aws:kms:us-
east-1:111111111111:key/1234abcd-12ab-34cd-56ef-1234567890ab",
               "arn:aws:kms:us-
east-1:222222222222:key/4567dcba-23ab-34cd-56ef-0987654321yz"
           ],
           "Condition": {
               "Bool": {
                   "kms:GrantIsForAWSResource": "true"
               }
           }
       },
       {
           "Effect": "Allow",
           "Action": [
               "kms:Encrypt",
               "kms:Decrypt",
               "kms:ReEncrypt*",
               "kms:GenerateDataKey*",
               "kms:DescribeKey"
           ],
           "Resource": [
               "arn:aws:kms:us-
east-1:111111111111:key/1234abcd-12ab-34cd-56ef-1234567890ab",
               "arn:aws:kms:us-
east-1:222222222222:key/4567dcba-23ab-34cd-56ef-0987654321yz"
           ]
       }
```

```
    ]
}
```

5. Pilih Tinjau kebijakan
6. Untuk Nama, masukkan nama deskriptif untuk kebijakan, lalu pilih Buat kebijakan.

Command line

Menggunakan editor teks pilihan Anda, buat file JSON baru bernama `policyDetails.json`. Ganti kebijakan yang ada dengan yang berikut ini, dan tentukan ARN kunci KMS yang digunakan untuk mengenkripsi volume sumber dan yang dibagikan dengan Anda oleh akun sumber di Langkah 2.

> **Note**
>
> Jika Anda menyalin dari beberapa akun sumber, Anda harus menentukan ARN kunci KMS yang sesuai dari setiap akun sumber.

Pada contoh berikut, kebijakan memberikan izin peran IAM untuk menggunakan kunci KMS `1234abcd-12ab-34cd-56ef-1234567890ab`, yang dibagikan oleh akun sumber `111111111111`, dan kunci KMS`4567dcba-23ab-34cd-56ef-0987654321yz`, yang ada di akun target `222222222222`.

> **Tip**
>
> Untuk mengikuti prinsip hak akses paling rendah, jangan biarkan akses penuh ke `kms:CreateGrant`. Sebagai gantinya, gunakan tombol `kms:GrantIsForAWSResource` kondisi untuk memungkinkan pengguna membuat hibah pada kunci KMS hanya ketika hibah dibuat atas nama pengguna oleh AWS layanan, seperti yang ditunjukkan pada contoh berikut.

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

```
            "Action": [
                "kms:RevokeGrant",
                "kms:CreateGrant",
                "kms:ListGrants"
            ],
            "Resource": [
                "arn:aws:kms:us-
east-1:111111111111:key/1234abcd-12ab-34cd-56ef-1234567890ab",
                "arn:aws:kms:us-
east-1:222222222222:key/4567dcba-23ab-34cd-56ef-0987654321yz"
            ],
            "Condition": {
                "Bool": {
                    "kms:GrantIsForAWSResource": "true"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:DescribeKey"
            ],
            "Resource": [
                "arn:aws:kms:us-
east-1:111111111111:key/1234abcd-12ab-34cd-56ef-1234567890ab",
                "arn:aws:kms:us-
east-1:222222222222:key/4567dcba-23ab-34cd-56ef-0987654321yz"
            ]
        }
    ]
}
```

Simpan dan tutup file. Kemudian, gunakan perintah put-role-policy untuk menambahkan kebijakan terkait peran IAM.

Sebagai contoh

```
$ aws iam put-role-policy \
    --role-name AWSDataLifecycleManagerDefaultRole \
```

```
    --policy-name CopyPolicy \
    --policy-document file://AdminPolicy.json
```

Tentukan filter deskripsi snapshot

Ketika Anda membuat kebijakan penyalinan snapshot di akun target, Anda harus menentukan filter deskripsi snapshot. Snapshot deskripsi filter memungkinkan Anda untuk menentukan tingkat tambahan pemfilteran yang memungkinkan Anda mengontrol snapshot yang disalin oleh kebijakan. Artinya, snapshot hanya disalin oleh kebijakan jika dibagikan oleh salah satu akun sumber tertentu, dan memiliki deskripsi snapshot yang cocok dengan filter yang ditentukan. Dengan kata lain, jika snapshot dibagi oleh salah satu akun kursus yang ditentukan, tetapi tidak memiliki deskripsi yang cocok dengan filter yang ditentukan, itu tidak disalin oleh kebijakan.

Deskripsi filter snapshot harus ditentukan menggunakan ekspresi reguler. Ini adalah bidang wajib saat membuat kebijakan peristiwa penyalinan lintas akun menggunakan konsol dan baris perintah. Berikut ini adalah contoh ekspresi reguler yang dapat digunakan:

- `.*`—Filter ini cocok dengan semua deskripsi snapshot. Jika Anda menggunakan ekspresi ini kebijakan akan menyalin semua snapshot yang dibagi oleh salah satu akun sumber tertentu.
- `Created for policy: policy-0123456789abcdef0.*`—Filter ini hanya cocok dengan snapshot yang dibuat oleh kebijakan dengan ID `policy-0123456789abcdef0`. Jika Anda menggunakan ekspresi seperti ini, hanya snapshot yang dibagikan dengan akun Anda oleh salah satu akun sumber tertentu, dan yang telah dibuat oleh kebijakan dengan ID tertentu yang akan disalin oleh kebijakan tersebut.
- `.*production.*`—Filter ini cocok dengan setiap snapshot yang memiliki kata `production` di mana saja dalam deskripsi. Jika Anda menggunakan ekspresi ini kebijakan akan menyalin semua snapshot yang dibagi oleh salah satu akun sumber tertentu dan yang memiliki teks tertentu dalam deskripsi mereka.

Pertimbangan untuk kebijakan penyalinan snapshot lintas akun

Pertimbangan berikut berlaku untuk kebijakan peristiwa penyalinan lintas akun:

- Anda hanya dapat menyalin snapshot yang tidak dienkripsi atau yang dienkripsi menggunakan kunci yang dikelola pelanggan.
- Anda dapat membuat kebijakan peristiwa penyalinan lintas akun untuk menyalin snapshot yang dibagi di luar Amazon Data Lifecycle Manager.

- Jika Anda ingin mengenkripsi snapshot di akun target, peran IAM yang dipilih untuk kebijakan peristiwa salinan lintas akun harus memiliki izin untuk menggunakan kunci KMS yang diperlukan.

Sumber daya tambahan

Untuk informasi selengkapnya, lihat Mengotomatisasi menyalin snapshot AWS Amazon EBS terenkripsi di seluruh blog penyimpanan akun. AWS

Melihat, memodifikasi, dan menghapus kebijakan siklus hidup

Gunakan prosedur berikut untuk melihat, mengubah, dan menghapus kebijakan siklus hidup yang ada.

Topik

Lihat kebijakan siklus hidup

Gunakan salah satu prosedur berikut ini untuk melihat kebijakan siklus hidup.

Console

Untuk melihat kebijakan siklus hidup

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Elastic Block Store, Lifecycle Manager.
3. Pilih ID kebijakan siklus hidup dari daftar.

Command line

Untuk mendapatkan informasi ringkasan tentang kebijakan siklus hidup Anda

Gunakan perintah get-lifecycle-policies.

```
aws dlm get-lifecycle-policies
```

Untuk menampilkan informasi tentang kebijakan siklus hidup tertentu

Gunakan perintah get-lifecycle-policy. Untuk `--policy-id`, tentukan ID kebijakan guna dapat melihat.

```
aws dlm get-lifecycle-policy --policy-id policy-0123456789abcdef0
```

Modifikasi kebijakan siklus hidup

Pertimbangan untuk memodifikasi kebijakan

- Jika Anda memodifikasi kebijakan AMI atau snapshot dengan menghapus tanda target, volume atau instans tanda tersebut tidak lagi dikelola oleh kebijakan.
- Jika Anda memodifikasi nama jadwal, snapshot atau AMI yang dibuat dengan nama jadwal lama tidak lagi dikelola oleh kebijakan.
- Jika Anda memodifikasi jadwal retensi berbasis usia untuk menggunakan interval waktu baru, interval baru hanya digunakan untuk snapshot baru atau AMI yang dibuat setelah perubahan. Jadwal baru tidak memengaruhi jadwal retensi snapshot atau AMI yang dibuat sebelum perubahan.
- Anda tidak dapat mengubah jadwal penyimpanan kebijakan dari berdasarkan hitungan menjadi berbasis usia setelah pembuatan. Untuk melakukan perubahan ini, Anda harus membuat kebijakan baru.
- Jika Anda menonaktifkan kebijakan dengan jadwal retensi berbasis usia, snapshot, atau AMI yang ditetapkan berakhir saat kebijakan dinonaktifkan akan dipertahankan tanpa batas. Anda harus menghapus snapshot atau membatalkan pendaftaran AMI secara manual. Saat Anda mengaktifkan ulang kebijakan, Amazon Data Lifecycle Manager akan melanjutkan penghapusan snapshot karena periode retensinya berakhir.
- Jika Anda menonaktifkan kebijakan dengan jadwal retensi berbasis jumlah, kebijakan akan berhenti membuat dan menghapus snapshot atau AMI. Saat Anda mengaktifkan kembali kebijakan, Amazon Data Lifecycle Manager akan melanjutkan pembuatan snapshot, dan melanjutkan penghapusan snapshot saat ambang retensi terpenuhi.
- Jika Anda menonaktifkan kebijakan yang memiliki kebijakan yang mengaktifkan pengarsipan snapshot, snapshot yang berada di tingkat arsip pada saat penonaktifan kebijakan tidak lagi dikelola oleh Amazon Data Lifecycle Manager. Anda harus menghapus snapshot secara manual jika tidak lagi diperlukan.

- Jika Anda mengaktifkan pengarsipan snapshot pada jadwal berbasis jumlah, aturan pengarsipan berlaku untuk semua snapshot baru yang dibuat dan diarsipkan berdasarkan jadwal, dan juga berlaku untuk snapshot yang ada yang sebelumnya dibuat dan diarsipkan berdasarkan jadwal.
- Jika Anda mengaktifkan pengarsipan snapshot pada jadwal berbasis usia, aturan pengarsipan hanya berlaku untuk snapshot baru yang dibuat setelah mengaktifkan pengarsipan snapshot. Snapshot yang ada yang dibuat sebelum pengaktifan pengarsipan snapshot terus dihapus dari tingkatan penyimpanan masing-masing, sesuai dengan jadwal yang ditetapkan saat snapshot tersebut awalnya dibuat dan diarsipkan.
- Jika Anda menonaktifkan pengarsipan snapshot untuk jadwal berbasis hitungan, jadwal akan segera berhenti mengarsipkan snapshot. Snapshot yang sebelumnya diarsipkan berdasarkan jadwal tetap berada di tingkat arsip dan tidak akan dihapus oleh Amazon Data Lifecycle Manager.
- Jika Anda menonaktifkan pengarsipan snapshot untuk jadwal berdasarkan usia, snapshot yang dibuat oleh kebijakan dan yang dijadwalkan untuk diarsipkan akan dihapus secara permanen pada tanggal dan waktu arsip terjadwal, seperti yang ditunjukkan oleh tanda sistem `aws:dlm:expirationTime`.
- Jika Anda menonaktifkan pengarsipan snapshot untuk jadwal berbasis jumlah, jadwal akan segera berhenti mengarsipkan snapshot. Snapshot yang sebelumnya diarsipkan berdasarkan jadwal tetap berada di tingkat arsip dan tidak akan dihapus oleh Amazon Data Lifecycle Manager.
- Jika Anda mengubah jumlah retensi arsip untuk jadwal berbasis jumlah, jumlah retensi baru menyertakan snapshot yang sudah ada yang sebelumnya diarsipkan oleh jadwal.
- Jika Anda mengubah periode retensi arsip untuk jadwal berdasarkan usia, periode retensi baru hanya berlaku untuk snapshot yang diarsipkan setelah mengubah aturan retensi.

Gunakan salah satu prosedur berikut ini untuk memodifikasi kebijakan siklus hidup.

Console

Untuk mengubah kebijakan siklus hidup

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Elastic Block Store, Lifecycle Manager.
3. Pilih kebijakan siklus hidup dari daftar.
4. Pilih Tindakan, Modifikasi kebijakan siklus hidup.
5. Ubah pengaturan kebijakan sesuai kebutuhan. Misalnya, Anda dapat mengubah jadwal, menambahkan atau menghapus tanda, atau mengaktifkan atau menonaktifkan kebijakan.

6. Pilih Modifikasi kebijakan.

Command line

Gunakan perintah update-lifecycle-policy untuk mengubah informasi dalam kebijakan siklus hidup. Untuk menyederhanakan sintaksis, contoh ini mengacu pada file JSON, `policyDetailsUpdated.json`, yang mencakup detail kebijakan.

```
aws dlm update-lifecycle-policy \
    --state DISABLED \
    --execution-role-arn
 arn:aws:iam::12345678910:role/AWSDataLifecycleManagerDefaultRole" \
    --policy-details file://policyDetailsUpdated.json
```

Berikut ini adalah contoh file `policyDetailsUpdated.json`.

```
{
   "ResourceTypes":[
      "VOLUME"
   ],
   "TargetTags":[
      {
         "Key": "costcenter",
         "Value": "120"
      }
   ],
   "Schedules":[
      {
         "Name": "DailySnapshots",
         "TagsToAdd": [
            {
               "Key": "type",
               "Value": "myDailySnapshot"
            }
         ],
         "CreateRule": {
            "Interval": 12,
            "IntervalUnit": "HOURS",
            "Times": [
               "15:00"
            ]
         },
```

```
            "RetainRule": {
               "Count" :5
            },
            "CopyTags": false
         }
      ]
}
```

Untuk melihat kebijakan yang diperbarui, gunakan perintah `get-lifecycle-policy`. Anda dapat melihat bahwa status, nilai tanda, interval snapshot, dan waktu mulai snapshot diubah.

Hapus kebijakan siklus hidup

Pertimbangan untuk memodifikasi kebijakan

- Jika Anda menghapus kebijakan, snapshot atau AMI yang dibuat oleh kebijakan tersebut tidak dihapus secara otomatis. Jika Anda tidak lagi membutuhkan snapshot atau AMI, Anda harus menghapusnya secara manual.
- Jika Anda menghapus kebijakan yang mengaktifkan kebijakan pengarsipan snapshot, snapshot yang berada di tingkat arsip pada saat penghapusan kebijakan tidak lagi dikelola oleh Amazon Data Lifecycle Manager. Anda harus menghapus snapshot secara manual jika tidak lagi diperlukan.
- Jika Anda menghapus kebijakan dengan jadwal berbasis usia yang diaktifkan pengarsipan, snapshot yang dibuat oleh kebijakan dan yang dijadwalkan untuk diarsipkan akan dihapus secara permanen pada tanggal dan waktu arsip terjadwal, seperti yang ditunjukkan oleh tanda sistem `aws:dlm:expiretiontime`.

Gunakan salah satu prosedur berikut ini untuk menghapus kebijakan siklus hidup.

Console

Untuk menghapus kebijakan siklus hidup

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Elastic Block Store, Lifecycle Manager.
3. Pilih kebijakan siklus hidup dari daftar.
4. Pilih Tindakan, Hapus kebijakan siklus hidup.

5. Jika diminta untuk mengonfirmasi, pilih Hapus kebijakan.

Command line

Gunakan perintah delete-lifecycle-policy untuk menghapus kebijakan siklus hidup dan mengosongkan tanda target yang ditentukan dalam kebijakan untuk digunakan kembali.

> **Note**
>
> Anda dapat menghapus snapshot yang dibuat hanya oleh Amazon Data Lifecycle Manager.

```
aws dlm delete-lifecycle-policy --policy-id policy-0123456789abcdef0
```

Referensi API Amazon Data Lifecycle Manager memberikan deskripsi dan sintaksis untuk setiap tindakan dan jenis data untuk API Kueri Amazon Data Lifecycle Manager.

Atau, Anda dapat menggunakan salah satu AWS SDK untuk mengakses API dengan cara yang disesuaikan dengan bahasa pemrograman atau platform yang Anda gunakan. Untuk informasi selengkapnya, lihat AWS SDK.

AWS Identity and Access Management

Akses ke Amazon Data Lifecycle Manager memerlukan krekredensial. Kredensial tersebut harus memiliki izin untuk mengakses sumber daya AWS , seperti instans, volume, snapshot, dan AMI. Bagian berikut memberikan rincian tentang bagaimana Anda dapat menggunakan AWS Identity and Access Management (IAM), dan membantu mengamankan akses ke sumber daya Anda.

Topik

- AWS kebijakan terkelola
- Peran layanan IAM
- Izin untuk pengguna
- Izin untuk enkripsi

AWS kebijakan terkelola

Kebijakan AWS terkelola adalah kebijakan mandiri yang dibuat dan dikelola oleh AWS. AWS kebijakan terkelola dirancang untuk memberikan izin untuk banyak kasus penggunaan umum. AWS Kebijakan terkelola membuatnya lebih efisien bagi Anda untuk menetapkan izin yang sesuai kepada pengguna, grup, dan peran, daripada jika Anda harus menulis kebijakan sendiri.

Namun, Anda tidak dapat mengubah izin yang ditentukan dalam kebijakan AWS terkelola. AWS sesekali memperbarui izin yang ditentukan dalam kebijakan AWS terkelola. Ketika ini terjadi, pembaruan memengaruhi semua entitas pengguna utama (pengguna, grup, dan peran) yang terkait dengan kebijakan tersebut.

Amazon Data Lifecycle Manager menyediakan kebijakan AWS terkelola untuk kasus penggunaan umum. Kebijakan ini membuatnya lebih efisien untuk menentukan izin yang sesuai dan mengontrol akses ke sumber daya Anda. Kebijakan AWS terkelola yang disediakan oleh Amazon Data Lifecycle Manager dirancang untuk dilampirkan ke peran yang diteruskan ke Amazon Data Lifecycle Manager.

Topik

- AWSDataLifecycleManagerServiceRole
- AWSDataLifecycleManagerServiceRoleForAMIManagement
- AWSDataLifecycleManagerSSMFullAccess
- AWS pembaruan kebijakan terkelola

AWSDataLifecycleManagerServiceRole

AWSDataLifecycleManagerServiceRoleKebijakan ini memberikan izin yang sesuai kepada Amazon Data Lifecycle Manager untuk membuat dan mengelola kebijakan snapshot Amazon EBS dan kebijakan peristiwa salinan lintas akun.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateSnapshot",
                "ec2:CreateSnapshots",
                "ec2:DeleteSnapshot",
```

```
                "ec2:DescribeInstances",
                "ec2:DescribeVolumes",
                "ec2:DescribeSnapshots",
                "ec2:EnableFastSnapshotRestores",
                "ec2:DescribeFastSnapshotRestores",
                "ec2:DisableFastSnapshotRestores",
                "ec2:CopySnapshot",
                "ec2:ModifySnapshotAttribute",
                "ec2:DescribeSnapshotAttribute",
                "ec2:ModifySnapshotTier",
                "ec2:DescribeSnapshotTierStatus"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:*::snapshot/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "events:PutRule",
                "events:DeleteRule",
                "events:DescribeRule",
                "events:EnableRule",
                "events:DisableRule",
                "events:ListTargetsByRule",
                "events:PutTargets",
                "events:RemoveTargets"
            ],
            "Resource": "arn:aws:events:*:*:rule/AwsDataLifecycleRule.managed-cwe.*"
        }
    ]
}
```

AWSDataLifecycleManagerServiceRoleForAMIManagement

AWSDataLifecycleManagerServiceRoleForAMIManagementKebijakan ini memberikan izin yang sesuai kepada Amazon Data Lifecycle Manager untuk membuat dan mengelola kebijakan AMI yang didukung Amazon EBS-backed.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": [
                "arn:aws:ec2:*::snapshot/*",
                "arn:aws:ec2:*::image/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeImages",
                "ec2:DescribeInstances",
                "ec2:DescribeImageAttribute",
                "ec2:DescribeVolumes",
                "ec2:DescribeSnapshots"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:DeleteSnapshot",
            "Resource": "arn:aws:ec2:*::snapshot/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:ResetImageAttribute",
                "ec2:DeregisterImage",
                "ec2:CreateImage",
                "ec2:CopyImage",
                "ec2:ModifyImageAttribute"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:EnableImageDeprecation",
                "ec2:DisableImageDeprecation"
```

```
            ],
            "Resource": "arn:aws:ec2:*::image/*"
        }
    ]
}
```

AWSDataLifecycleManagerSSMFullAccess

Memberikan izin kepada Amazon Data Lifecycle Manager untuk melakukan tindakan Systems Manager yang diperlukan untuk menjalankan skrip pra dan pasca di semua instans Amazon EC2.

> **Important**
>
> Kebijakan menggunakan kunci syarat `aws:ResourceTag` untuk membatasi akses ke dokumen SSM tertentu saat menggunakan skrip pra dan pasca. Untuk mengizinkan Amazon Data Lifecycle Manager mengakses dokumen SSM, Anda harus memastikan bahwa dokumen SSM Anda ditandai dengan `DLMScriptsAccess:true`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowSSMReadOnlyAccess",
            "Effect": "Allow",
            "Action": [
                "ssm:GetCommandInvocation",
                "ssm:ListCommands",
                "ssm:DescribeInstanceInformation"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowTaggedSSMDocumentsOnly",
            "Effect": "Allow",
            "Action": [
                "ssm:SendCommand",
                "ssm:DescribeDocument",
                "ssm:GetDocument"
            ],
            "Resource": [
```

```
                "arn:aws:ssm:*:*:document/*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/DLMScriptsAccess": "true"
                }
            }
        },
        {
            "Sid": "AllowSpecificAWSOwnedSSMDocuments",
            "Effect": "Allow",
            "Action": [
                "ssm:SendCommand",
                "ssm:DescribeDocument",
                "ssm:GetDocument"
            ],
            "Resource": [
                "arn:aws:ssm:*:*:document/AWSEC2-CreateVssSnapshot",
                "arn:aws:ssm:*:*:document/AWSSystemsManagerSAP-
CreateDLMSnapshotForSAPHANA"
            ]
        },
        {
            "Sid": "AllowAllEC2Instances",
            "Effect": "Allow",
            "Action": [
                "ssm:SendCommand"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:instance/*"
            ]
        }
    ]
}
```

AWS pembaruan kebijakan terkelola

AWS layanan memelihara dan memperbarui kebijakan AWS terkelola. Anda tidak dapat mengubah izin dalam kebijakan AWS terkelola. Layanan terkadang menambahkan izin tambahan ke kebijakan AWS terkelola untuk mendukung fitur baru. Jenis pembaruan ini akan memengaruhi semua identitas (pengguna, grup, dan peran) di mana kebijakan tersebut dilampirkan. Layanan kemungkinan besar akan memperbarui kebijakan AWS terkelola saat fitur baru diluncurkan atau saat operasi

baru tersedia. Layanan tidak menghapus izin dari kebijakan AWS terkelola, sehingga pembaruan kebijakan tidak akan merusak izin yang ada.

Tabel berikut memberikan detail tentang pembaruan kebijakan AWS terkelola untuk Amazon Data Lifecycle Manager sejak layanan ini mulai melacak perubahan ini. Untuk peringatan otomatis tentang perubahan pada halaman ini, berlangganan ke umpan RSS pada halaman Riwayat dokumen untuk Panduan Pengguna Amazon EBS.

Perubahan	Deskripsi	Tanggal
AWSDataLifecycleManagerSSMFullAccess—Memperbarui izin kebijakan.	Memperbarui kebijakan untuk mendukung snapshot yang konsisten dengan aplikasi untuk SAP HANA menggunakan dokumen SSM `AWSSystemsManagerSAP-CreateDLMSnapshotForSAPHANA` .	17 November 2023
AWSDataLifecycleManagerSSMFullAccess—Menambahkan kebijakan AWS terkelola baru.	Amazon Data Lifecycle Manager menambahkan kebijakan terkelola. AWSDataLifecycleManagerSSMFullAccess AWS	7 November 2023

Perubahan	Deskripsi	Tanggal
AWSDataLifecycleManagerServiceRole—Menambahkan izin untuk mendukung pengarsipan snapshot.	Amazon Data Lifecycle Manager menambahkan tindakan `ec2:ModifySnapshotTier` dan `ec2:DescribeSnapshotTierStatus` untuk memberikan izin kebijakan snapshot untuk mengarsipkan snapshot dan untuk memeriksa status pengarsipan untuk snapshot.	30 September 2022

Perubahan	Deskripsi	Tanggal
AWSDataLifecycleManagerServiceRoleForAMIManagement—Menambahkan izin untuk mendukung penghentian AMI.	Amazon Data Lifecycle Manager menambahkan tindakan `ec2:EnableImageDeprecation` dan `ec2:DisableImageDeprecation` untuk memberikan izin kebijakan AMI yang didukung EBS untuk mengaktifkan dan menonaktifkan penghentian AMI.	23 Agustus 2021
Amazon Data Lifecycle Manager mulai melacak perubahan	Amazon Data Lifecycle Manager mulai melacak perubahan untuk kebijakan yang dikelola. AWS	23 Agustus 2021

Peran layanan IAM

Peran AWS Identity and Access Management (IAM) mirip dengan pengguna, karena itu adalah AWS identitas dengan kebijakan izin yang menentukan apa yang dapat dan tidak dapat dilakukan identitas. AWS Namun, alih-alih secara unik terkait dengan satu orang, peran dimaksudkan untuk menjadi

dapat diambil oleh siapa pun yang membutuhkannya. Peran layanan adalah peran yang diasumsikan AWS layanan untuk melakukan tindakan atas nama Anda. Sebagai layanan yang melakukan operasi pencadangan atas nama Anda, Amazon Data Lifecycle Manager mengharuskan Anda meneruskan peran yang harus diambil saat melakukan operasi kebijakan atas nama Anda. Untuk informasi selengkapnya tentang peran IAM, lihat Peran IAM dalam Panduan Pengguna IAM.

Peran yang diteruskan ke Amazon Data Lifecycle Manager harus memiliki kebijakan IAM dengan izin yang memungkinkan Amazon Data Lifecycle Manager untuk melakukan tindakan yang terkait dengan operasi kebijakan, seperti membuat snapshot dan AMI, menyalin snapshot dan AMI, menghapus snapshot, dan membatalkan pendaftaran AMI. Izin yang berbeda diperlukan untuk setiap jenis kebijakan Amazon Data Lifecycle Manager. Peran ini juga harus memiliki Amazon Data Lifecycle Manager terdaftar sebagai entitas tepercaya, yang memungkinkan Amazon Data Lifecycle Manager untuk mengambil peran.

Topik

Peran layanan default untuk Amazon Data Lifecycle Manager

Amazon Data Lifecycle Manager menggunakan peran layanan default berikut:

- AWSDataLifecycleManagerDefaultRole—peran default untuk mengelola snapshot. Peran ini hanya memercayai layanan `dlm.amazonaws.com` untuk mengambil peran dan memungkinkan Amazon Data Lifecycle Manager untuk melakukan tindakan yang diperlukan oleh kebijakan penyalinan snapshot lintas akun dan snapshot atas nama Anda. Peran ini menggunakan kebijakan `AWSDataLifecycleManagerServiceRole` AWS terkelola.

> **Note**
>
> Format ARN peran berbeda tergantung pada apakah itu dibuat menggunakan konsol atau. AWS CLI Jika peran dibuat menggunakan konsol, format ARN adalah `arn:aws:iam::account_id:role/service-role/AWSDataLifecycleManagerDefaultRole`. Jika peran dibuat menggunakan AWS CLI, format ARN adalah. `arn:aws:iam::account_id:role/AWSDataLifecycleManagerDefaultRole`

- AWSDataLifecycleManagerDefaultRoleForAMIManagement—peran default untuk mengelola AMI. Layanan hanya mempercayai layanan `dlm.amazonaws.com` untuk mengambil peran tersebut dan memungkinkan Amazon Data Lifecycle Manager untuk melakukan tindakan yang diperlukan oleh kebijakan AMI yang didukung EBS atas nama Anda. Peran ini menggunakan kebijakan `AWSDataLifecycleManagerServiceRoleForAMIManagement` AWS terkelola.

Jika Anda menggunakan konsol Amazon Data Lifecycle Manager, Amazon Data Lifecycle Manager secara otomatis AWSDataLifecycleManagerDefaultRolemembuat peran layanan saat pertama kali Anda membuat kebijakan penyalinan snapshot atau snapshot lintas akun, dan secara otomatis membuat peran layanan saat pertama kali AWSDataLifecycleManagerDefaultRoleForAMIManagementmembuat kebijakan AMI yang didukung EBS.

Jika Anda tidak menggunakan konsol, Anda dapat membuat peran layanan secara manual menggunakan perintah create-default-role. Untuk`--resource-type`, tentukan `snapshot` untuk membuat AWSDataLifecycleManagerDefaultRole, atau `image` membuat AWSDataLifecycleManagerDefaultRoleForAMIManagement.

```
$ aws dlm create-default-role --resource-type snapshot|image
```

Jika Anda menghapus peran layanan default, kemudian ingin membuatnya lagi, Anda dapat menggunakan proses yang sama untuk membuatnya ulang di akun Anda.

Peran layanan kustom untuk Amazon Data Lifecycle Manager

Sebagai alternatif untuk menggunakan peran layanan default, Anda dapat membuat peran IAM kustom dengan izin yang diperlukan lalu memilihnya saat Anda membuat kebijakan siklus hidup.

Untuk membuat peran IAM kustom

1. Buat peran dengan izin sebagai berikut.

   - Izin diperlukan untuk mengelola kebijakan siklus hidup snapshot

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

```
                "ec2:CreateSnapshot",
                "ec2:CreateSnapshots",
                "ec2:DeleteSnapshot",
                "ec2:DescribeInstances",
                "ec2:DescribeVolumes",
                "ec2:DescribeSnapshots",
                "ec2:EnableFastSnapshotRestores",
                "ec2:DescribeFastSnapshotRestores",
                "ec2:DisableFastSnapshotRestores",
                "ec2:CopySnapshot",
                "ec2:ModifySnapshotAttribute",
                "ec2:DescribeSnapshotAttribute",
                "ec2:ModifySnapshotTier",
                "ec2:DescribeSnapshotTierStatus"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:*::snapshot/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "events:PutRule",
                "events:DeleteRule",
                "events:DescribeRule",
                "events:EnableRule",
                "events:DisableRule",
                "events:ListTargetsByRule",
                "events:PutTargets",
                "events:RemoveTargets"
            ],
            "Resource": "arn:aws:events:*:*:rule/AwsDataLifecycleRule.managed-
cwe.*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetCommandInvocation",
                "ssm:ListCommands",
```

```
                "ssm:DescribeInstanceInformation"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:SendCommand",
                "ssm:DescribeDocument",
                "ssm:GetDocument"
            ],
            "Resource": [
                "arn:aws:ssm:*:*:document/*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/DLMScriptsAccess": "true"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:SendCommand",
                "ssm:DescribeDocument",
                "ssm:GetDocument"
            ],
            "Resource": [
                "arn:aws:ssm:*::document/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:SendCommand"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:instance/*"
            ],
            "Condition": {
                "StringNotLike": {
                    "aws:ResourceTag/DLMScriptsAccess": "false"
                }
            }
```

```
        }
    ]
}
```

- Izin diperlukan untuk mengelola kebijakan siklus hidup AMI

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": [
                "arn:aws:ec2:*::snapshot/*",
                "arn:aws:ec2:*::image/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeImages",
                "ec2:DescribeInstances",
                "ec2:DescribeImageAttribute",
                "ec2:DescribeVolumes",
                "ec2:DescribeSnapshots"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:DeleteSnapshot",
            "Resource": "arn:aws:ec2:*::snapshot/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:ResetImageAttribute",
                "ec2:DeregisterImage",
                "ec2:CreateImage",
                "ec2:CopyImage",
                "ec2:ModifyImageAttribute"
            ],
            "Resource": "*"
        },
```

```
        {
            "Effect": "Allow",
            "Action": [
                "ec2:EnableImageDeprecation",
                "ec2:DisableImageDeprecation"
            ],
            "Resource": "arn:aws:ec2:*::image/*"
        }
    ]
}
```

Untuk informasi selengkapnya, lihat Membuat Peran dalam Panduan Pengguna IAM.

2. Tambahkan hubungan kepercayaan ke peran tersebut.

   a. Di konsol IAM, pilih Peran.

   b. Pilih peran yang Anda buat, lalu pilih Hubungan kepercayaan.

   c. Pilih Edit Hubungan Kepercayaan, tambahkan kebijakan berikut, lalu pilih Perbarui Kebijakan Kepercayaan.

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

   Kami menyarankan Anda menggunakan kunci syarat `aws:SourceAccount` dan `aws:SourceArn` untuk melindungi diri Anda dari masalah wakil yang membingungkan. Misalnya, Anda dapat menambahkan blok kondisi berikut ke kebijakan kepercayaan sebelumnya. `aws:SourceAccount` adalah pemilik kebijakan siklus hidup dan `aws:SourceArn` adalah ARN dari kebijakan siklus hidup. Jika Anda tidak mengetahui ID kebijakan siklus hidup, Anda dapat mengganti bagian ARN tersebut dengan wildcard (*), lalu memperbarui kebijakan trust setelah Anda membuat kebijakan siklus hidup.

```
"Condition": {
```

```
        "StringEquals": {
            "aws:SourceAccount": "account_id"
        },
        "ArnLike": {
            "aws:SourceArn": "arn:partition:dlm:region:account_id:policy/policy_id"
        }
}
```

Izin untuk pengguna

Pengguna harus memiliki izin berikut untuk menggunakan Amazon Data Lifecycle Manager.

> **Note**
>
> - Izin `ec2:DescribeAvailabilityZones`, `ec2:DescribeRegions`, `kms:ListAliases`, dan `kms:DescribeKey` diperlukan hanya untuk pengguna konsol. Jika akses konsol tidak diperlukan, Anda dapat menghapus izin.
> - Format ARN AWSDataLifecycleManagerDefaultRoleperan berbeda tergantung pada apakah itu dibuat menggunakan konsol atau. AWS CLI Jika peran dibuat menggunakan konsol, format ARN adalah `arn:aws:iam::account_id:role/service-role/AWSDataLifecycleManagerDefaultRole`. Jika peran dibuat menggunakan AWS CLI, format ARN adalah Kebijakan `arn:aws:iam::account_id:role/AWSDataLifecycleManagerDefaultRole` berikut mengasumsikan peran dibuat menggunakan. AWS CLI

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "dlm:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": [
```

```
                "arn:aws:iam::account_id:role/service-role/
AWSDataLifecycleManagerDefaultRole",
                "arn:aws:iam::account_id:role/service-role/
AWSDataLifecycleManagerDefaultRoleForAMIManagement"
                ]
        },
        {
            "Effect": "Allow",
            "Action": "iam:ListRoles",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeAvailabilityZones",
                "ec2:DescribeRegions",
                "kms:ListAliases",
                "kms:DescribeKey"
            ],
            "Resource": "*"
        }
    ]
}
```

Untuk informasi selengkapnya, lihat Mengubah izin untuk pengguna pada Panduan Pengguna IAM.

Izin untuk enkripsi

Pertimbangkan hal berikut saat bekerja dengan Amazon Data Lifecycle Manager dan sumber daya terenkripsi.

- Jika volume sumber dienkripsi, pastikan bahwa AWSDataLifecycleManagerDefaultRoleperan default Amazon Data Lifecycle Manager AWSDataLifecycleManagerDefaultRoleForAMIManagement(dan) memiliki izin untuk menggunakan kunci KMS yang digunakan untuk mengenkripsi volume.
- Jika Anda mengaktifkan Salinan lintas Wilayah untuk snapshot yang tidak dienkripsi atau AMI yang didukung oleh snapshot yang tidak dienkripsi, dan memilih untuk mengaktifkan enkripsi di Wilayah tujuan, pastikan bahwa peran default memiliki izin untuk menggunakan kunci KMS yang diperlukan untuk melakukan enkripsi di Wilayah tujuan.

- Jika Anda mengaktifkan Salinan lintas Wilayah untuk snapshot terenkripsi atau AMI yang didukung oleh snapshot terenkripsi, pastikan bahwa peran default memiliki izin untuk menggunakan kunci KMS sumber dan tujuan.
- Jika Anda mengaktifkan pengarsipan snapshot untuk snapshot terenkripsi, pastikan peran default Amazon Data Lifecycle Manager AWSDataLifecycleManagerDefaultRole(memiliki izin untuk menggunakan kunci KMS yang digunakan untuk mengenkripsi snapshot.

Untuk informasi selengkapnya, lihat Mengizinkan pengguna di akun lain untuk menggunakan kunci KMS di Panduan Developer AWS Key Management Service .

Memantau siklus hidup snapshot dan AMI

Anda dapat menggunakan fitur berikut untuk siklus hidup snapshot dan AMI.

Fitur

- Konsol dan AWS CLI
- AWS CloudTrail
- Pantau kebijakan Anda menggunakan CloudWatch Acara
- Pantau kebijakan Anda menggunakan Amazon CloudWatch

Konsol dan AWS CLI

Anda dapat melihat kebijakan siklus hidup Anda menggunakan konsol Amazon EC2 atau AWS CLI. Setiap snapshot dan AMI yang dibuat oleh kebijakan memiliki stempel waktu dan tanda terkait kebijakan. Anda dapat memfilter snapshot dan AMI menggunakan tanda ini untuk memverifikasi bahwa cadangan Anda sedang dibuat seperti yang Anda inginkan. Untuk informasi tentang kebijakan siklus tampilan menggunakan konsol, lihat Lihat kebijakan siklus hidup.

AWS CloudTrail

Dengan AWS CloudTrail, Anda dapat melacak aktivitas pengguna dan penggunaan API untuk menunjukkan kepatuhan terhadap kebijakan internal dan standar peraturan. Untuk informasi selengkapnya, silakan lihat Panduan Pengguna AWS CloudTrail.

Pantau kebijakan Anda menggunakan CloudWatch Acara

Amazon EBS dan Amazon Data Lifecycle Manager memancarkan peristiwa terkait tindakan kebijakan siklus hidup. Anda dapat menggunakan AWS Lambda dan CloudWatch Acara Amazon untuk menangani pemberitahuan acara secara terprogram. Peristiwa dipancarkan atas dasar upaya terbaik. Untuk informasi selengkapnya, lihat Panduan Pengguna CloudWatch Acara Amazon.

Peristiwa berikut ini tersedia:

> **Note**
>
> Tidak ada peristiwa yang dipancarkan untuk tindakan kebijakan siklus hidup AMI.

- `createSnapshot` — Peristiwa Amazon EBS yang dipancarkan saat tindakan `CreateSnapshot` berhasil atau gagal. Untuk informasi selengkapnya, lihat Amazon EventBridge untuk Amazon EBS.
- `DLM Policy State Change` — Peristiwa Amazon Data Lifecycle Manager dipancarkan ketika kebijakan siklus hidup memasuki status kesalahan. Peristiwa ini berisi deskripsi penyebab kesalahan.

  Berikut ini adalah contoh peristiwa ketika izin yang diberikan oleh peran IAM tidak memadai.

```
{
    "version": "0",
    "id": "01234567-0123-0123-0123-0123456789ab",
    "detail-type": "DLM Policy State Change",
    source": "aws.dlm",
    "account": "123456789012",
    "time": "2018-05-25T13:12:22Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:dlm:us-east-1:123456789012:policy/policy-0123456789abcdef"
    ],
    "detail": {
        "state": "ERROR",
        "cause": "Role provided does not have sufficient permissions",
        "policy_id": "arn:aws:dlm:us-east-1:123456789012:policy/
policy-0123456789abcdef"
    }
}
```

Berikut ini adalah contoh peristiwa saat batas terlampaui.

```
{
    "version": "0",
    "id": "01234567-0123-0123-0123-0123456789ab",
    "detail-type": "DLM Policy State Change",
    "source": "aws.dlm",
    "account": "123456789012",
    "time": "2018-05-25T13:12:22Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:dlm:us-east-1:123456789012:policy/policy-0123456789abcdef"
    ],
    "detail":{
        "state": "ERROR",
        "cause": "Maximum allowed active snapshot limit exceeded",
        "policy_id": "arn:aws:dlm:us-east-1:123456789012:policy/
policy-0123456789abcdef"
    }
}
```

- `DLM Pre Post Script Notification` — Peristiwa yang dipancarkan ketika skrip pra atau pasca dimulai, berhasil, atau gagal.

  Berikut ini contoh peristiwa saat cadangan VSS berhasil.

```
{
    "version": "0",
    "id": "12345678-1234-1234-1234-123456789012",
    "detail-type": "DLM Pre Post Script Notification",
    "source": "aws.dlm",
    "account": "123456789012",
    "time": "2023-10-27T22:04:52Z",
    "region": "us-east-1",
    "resources": ["arn:aws:dlm:us-east-1:123456789012:policy/
policy-01234567890abcdef"],
    "detail": {
        "script_stage": "",
        "result": "success",
        "cause": "",
        "policy_id": "arn:aws:dlm:us-east-1:123456789012:policy/
policy-01234567890abcdef",
```

```
        "execution_handler": "AWS_VSS_BACKUP",
        "source": "arn:aws:ec2:us-east-1:123456789012:instance/i-01234567890abcdef",
        "resource_type": "EBS_SNAPSHOT",
        "resources": [{
            "status": "pending",
            "resource_id": "arn:aws:ec2:us-east-1::snapshot/snap-01234567890abcdef",
            "source": "arn:aws:ec2:us-east-1:123456789012:volume/
vol-01234567890abcdef"
        }],
        "request_id": "a1b2c3d4-a1b2-a1b2-a1b2-a1b2c3d4e5f6",
        "start_time": "2023-10-27T22:03:29.370Z",
        "end_time": "2023-10-27T22:04:51.370Z",
        "timeout_time": ""
    }
}
```

Pantau kebijakan Anda menggunakan Amazon CloudWatch

Anda dapat memantau kebijakan siklus hidup Amazon Data Lifecycle Manager menggunakan CloudWatch, yang mengumpulkan data mentah dan memprosesnya menjadi metrik hampir real-time yang dapat dibaca. Anda dapat menggunakan metrik ini untuk melihat dengan tepat banyaknya snapshot Amazon EBS serta AMI yang didukung EBS yang dibuat, dihapus, dan disalin oleh kebijakan Anda dari waktu ke waktu. Anda juga dapat mengatur alarm yang memperhatikan ambang batas tertentu dan mengirim notifikasi atau mengambil tindakan saat ambang batas tersebut terpenuhi.

Metrik disimpan selama periode 15 bulan, sehingga Anda dapat mengakses informasi historis dan mendapatkan pemahaman yang lebih baik tentang bagaimana performa kebijakan siklus hidup Anda selama periode yang lama.

Untuk informasi selengkapnya tentang Amazon CloudWatch, lihat Panduan CloudWatch Pengguna Amazon.

Topik

- Metrik yang didukung
- Melihat CloudWatch metrik untuk kebijakan Anda
- Grafik metrik untuk kebijakan Anda
- Buat CloudWatch alarm untuk kebijakan

- Contoh kasus penggunaan
- Mengelola kebijakan yang melaporkan tindakan gagal

Metrik yang didukung

Namespace `Data Lifecycle Manager` menyertakan metrik berikut untuk kebijakan siklus hidup Amazon Data Lifecycle Manager. Metrik yang didukung berbeda menurut jenis kebijakan.

Semua metrik dapat diukur pada `DLMPolicyId` dimensi. Statistik yang paling berguna adalah `sum` dan `average`, dan satuan ukurannya adalah `count`.

Pilih tab untuk melihat metrik yang didukung oleh jenis kebijakan tersebut.

EBS snapshot policies

Metrik	Deskripsi
`ResourcesTargeted`	Jumlah sumber daya yang ditargetkan oleh tanda yang ditentukan dalam snapshot atau kebijakan AMI yang didukung EBS.
`SnapshotsCreateStarted`	Jumlah tindakan pembuatan snapshot yang diinisiasi oleh kebijakan snapshot. Setiap tindakan direkam hanya sekali, bahkan jika ada banyak percobaan ulang berikutnya. Jika tindakan pembuatan snapshot gagal, Amazon Data Lifecycle Manager mengirimkan metrik `SnapshotsCreateFailed` .
`SnapshotsCreateCompleted`	Jumlah tindakan pembuatan snapshot yang diinisiasi oleh kebijakan snapshot. Ini termasuk percobaan ulang yang berhasil dalam waktu 60 menit dari waktu yang dijadwalkan.
`SnapshotsCreateFailed`	Jumlah snapshot yang tidak dapat dibuat oleh kebijakan snapshot. Ini termasuk percobaan ulang yang tidak berhasil dalam waktu 60 menit dari waktu yang dijadwalkan.
`SnapshotsSharedCompleted`	Jumlah tindakan pembuatan snapshot yang dibagikan di seluruh akun oleh kebijakan snapshot.

Metrik	Deskripsi
`SnapshotsDeleteCompleted`	Jumlah snapshot yang dihapus oleh snapshot atau kebijakan AMI yang didukung EBS. Metrik ini hanya berlaku untuk snapshot yang dibuat oleh kebijakan. Hal ini tidak berlaku untuk salinan snapshot lintas Wilayah yang dibuat oleh kebijakan. Metrik ini mencakup snapshot yang dihapus ketika kebijakan AMI yang didukung EBS membatalkan pendaftaran AMI.
`SnapshotsDeleteFailed`	Jumlah snapshot yang tidak dapat dihapus oleh snapshot atau kebijakan AMI yang didukung EBS. Metrik ini hanya berlaku untuk snapshot yang dibuat oleh kebijakan. Hal ini tidak berlaku untuk salinan snapshot lintas Wilayah yang dibuat oleh kebijakan. Metrik ini mencakup snapshot yang dihapus ketika kebijakan AMI yang didukung EBS membatalkan pendaftaran AMI.
`SnapshotsCopiedRegionStarted`	Jumlah tindakan penyalinan snapshot lintas wilayah yang diinisiasi oleh kebijakan snapshot.
`SnapshotsCopiedRegionCompleted`	Jumlah tindakan penyalinan snapshot lintas wilayah yang dibuat oleh kebijakan snapshot. Ini termasuk percobaan ulang yang berhasil dalam waktu 24 jam dari waktu yang dijadwalkan.
`SnapshotsCopiedRegionFailed`	Jumlah tindakan penyalinan snapshot lintas wilayah yang tidak dapat dibuat oleh kebijakan snapshot. Ini termasuk percobaan ulang yang berhasil dalam waktu 24 jam dari waktu yang dijadwalkan.
`SnapshotsCopiedRegionDeleteCompleted`	Jumlah salinan snapshot lintas Wilayah yang dihapus, sebagaimana ditetapkan oleh aturan retensi, oleh kebijakan snapshot.

Metrik	Deskripsi
SnapshotsCopiedRegionDeleteFailed	Jumlah salinan snapshot lintas Wilayah yang tidak dapat dihapus, sebagaimana ditetapkan oleh aturan retensi, oleh kebijakan snapshot.
snapshotsArchiveDeletionFailed	Jumlah snapshot yang diarsipkan yang tidak dapat dihapus dari tingkat arsip oleh kebijakan snapshot.
snapshotsArchiveScheduled	Jumlah snapshot yang dijadwalkan untuk diarsipkan oleh kebijakan snapshot.
snapshotsArchiveCompleted	Jumlah snapshot yang berhasil diarsipkan oleh kebijakan snapshot.
snapshotsArchiveFailed	Jumlah snapshot yang tidak dapat diarsipkan oleh kebijakan snapshot.
snapshotsArchiveDeletionCompleted	Jumlah snapshot yang diarsipkan yang berhasil dihapus dari tingkat arsip oleh kebijakan snapshot.
PreScriptStarted	Jumlah instans saat skrip pra berhasil dimulai. Jika percobaan ulang skrip diaktifkan, metrik ini dapat dipancarkan beberapa kali per kebijakan yang dijalankan.
PreScriptCompleted	Jumlah instans saat skrip pra berhasil diselesaikan. Metrik dipancarkan meskipun skrip pra selesai di luar periode batas waktu yang ditentukan. Jika percobaan ulang skrip diaktifkan, metrik ini dapat dipancarkan beberapa kali per kebijakan yang dijalankan.

Metrik	Deskripsi
`PreScriptFailed`	Jumlah instans saat skrip pra gagal diselesaikan dengan sukses. Metrik dipancarkan meskipun skrip pra selesai di luar periode batas waktu yang ditentukan. Jika percobaan ulang skrip diaktifkan, metrik ini dapat dipancarkan beberapa kali per kebijakan yang dijalankan.
`PostScriptStarted`	Jumlah instans saat skrip pasca berhasil dimulai. Jika percobaan ulang skrip diaktifkan, metrik ini dapat dipancarkan beberapa kali per kebijakan yang dijalankan.
PostScriptSelesai	Jumlah instans saat skrip pasca berhasil diselesaikan. Metrik dipancarkan bahkan jika skrip pasca selesai di luar periode batas waktu yang ditentukan. Jika percobaan ulang skrip diaktifkan, metrik ini dapat dipancarkan beberapa kali per kebijakan yang dijalankan.
PostScriptGagal	Jumlah peristiwa saat skrip pasca gagal diselesaikan dengan sukses. Metrik dipancarkan bahkan jika skrip pasca selesai di luar periode batas waktu yang ditentukan. Jika percobaan ulang skrip diaktifkan, metrik ini dapat dipancarkan beberapa kali per kebijakan yang dijalankan.
`VSSBackupStarted`	Jumlah instans saat cadangan VSS berhasil dimulai. Jika percobaan ulang skrip diaktifkan, metrik ini dapat dipancarkan beberapa kali per kebijakan yang dijalankan.
`VSSBackupCompleted`	Jumlah instans saat cadangan VSS berhasil diselesaikan. Metrik dipancarkan bahkan jika cadangan VSS selesai di luar periode batas waktu. Jika percobaan ulang skrip diaktifkan, metrik ini dapat dipancarkan beberapa kali per kebijakan yang dijalankan.

Metrik	Deskripsi
`VSSBackup Failed`	Jumlah instans saat pencadangan VSS gagal diselesaikan dengan sukses. Metrik dipancarkan bahkan jika cadangan VSS selesai di luar periode batas waktu. Jika percobaan ulang skrip diaktifkan, metrik ini dapat dipancarkan beberapa kali per kebijakan yang dijalankan.

EBS-backed AMI policies

Metrik berikut dapat digunakan dengan kebijakan AMI yang didukung EBS:

Metrik	Deskripsi
`Resources Targeted`	Jumlah sumber daya yang ditargetkan oleh tanda yang ditentukan dalam snapshot atau kebijakan AMI yang didukung EBS.
`Snapshots DeleteCompleted`	Jumlah snapshot yang dihapus oleh snapshot atau kebijakan AMI yang didukung EBS. Metrik ini hanya berlaku untuk snapshot yang dibuat oleh kebijakan. Hal ini tidak berlaku untuk salinan snapshot lintas Wilayah yang dibuat oleh kebijakan. Metrik ini mencakup snapshot yang dihapus ketika kebijakan AMI yang didukung EBS membatalkan pendaftaran AMI.
`Snapshots DeleteFailed`	Jumlah snapshot yang tidak dapat dihapus oleh snapshot atau kebijakan AMI yang didukung EBS. Metrik ini hanya berlaku untuk snapshot yang dibuat oleh kebijakan. Hal ini tidak berlaku untuk salinan snapshot lintas Wilayah yang dibuat oleh kebijakan. Metrik ini mencakup snapshot yang dihapus ketika kebijakan AMI yang didukung EBS membatalkan pendaftaran AMI.
`Snapshots CopiedReg`	Jumlah salinan snapshot lintas Wilayah yang dihapus, sebagaimana ditetapkan oleh aturan retensi, oleh kebijakan snapshot.

Metrik	Deskripsi
ionDelete Completed	
Snapshots CopiedReg ionDelete Failed	Jumlah salinan snapshot lintas Wilayah yang tidak dapat dihapus, sebagaimana ditetapkan oleh aturan retensi, oleh kebijakan snapshot.
ImagesCre ateStarted	Jumlah CreateImagetindakan yang diprakarsai oleh kebijakan AMI yang didukung EBS.
ImagesCre ateCompleted	Jumlah AMI yang dibuat oleh kebijakan AMI yang didukung EBS.
ImagesCre ateFailed	Jumlah AMI yang dibuat oleh kebijakan AMI yang didukung EBS.
ImagesDer egisterCo mpleted	Jumlah AMI yang dibuat oleh kebijakan AMI yang didukung EBS.
ImagesDer egisterFailed	Jumlah AMI yang dibuat oleh kebijakan AMI yang didukung EBS.
ImagesCop iedRegion Started	Jumlah tindakan penyalinan lintas Wilayah yang diprakarsai oleh kebijakan AMI yang didukung oleh EBS.
ImagesCop iedRegion Completed	Jumlah salinan AMI lintas wilayah yang dibuat oleh kebijakan AMI yang didukung EBS.

Metrik	Deskripsi
`ImagesCopiedRegionFailed`	Jumlah salinan AMI lintas wilayah yang dibuat oleh kebijakan AMI yang didukung EBS.
`ImagesCopiedRegionDeregisterCompleted`	Jumlah salinan AMI lintas Wilayah yang dibatalkan pendaftarannya, sebagaimana ditetapkan oleh aturan retensi, oleh kebijakan AMI yang didukung EBS.
`ImagesCopiedRegionDeregisteredFailed`	Jumlah salinan AMI lintas Wilayah yang tidak dapat dibatalkan pendaftarannya, sebagaimana ditetapkan oleh aturan retensi, oleh kebijakan AMI yang didukung EBS.
`EnableImageDeprecationCompleted`	Jumlah AMI yang ditandai untuk dihentikan oleh kebijakan AMI yang didukung EBS.
`EnableImageDeprecationFailed`	Jumlah AMI yang ditandai untuk dihentikan oleh kebijakan AMI yang didukung EBS.
`EnableCopiedImageDeprecationCompleted`	Jumlah AMI yang ditandai untuk dihentikan oleh kebijakan AMI yang didukung EBS.
`EnableCopiedImageDeprecationFailed`	Jumlah AMI yang ditandai untuk dihentikan oleh kebijakan AMI yang didukung EBS.

Cross-account copy event policies

Metrik berikut ini dapat digunakan dengan kebijakan peristiwa penyalinan lintas akun:

Metrik	Deskripsi
`SnapshotsCopiedAccountStarted`	Jumlah tindakan menyalin snapshot lintas akun yang diinisiasi oleh kebijakan peristiwa penyalinan lintas akun.
`SnapshotsCopiedAccountCompleted`	Jumlah snapshot yang disalin dari akun lain oleh kebijakan peristiwa penyalinan lintas akun. Ini termasuk percobaan ulang yang berhasil dalam waktu 24 jam dari waktu yang dijadwalkan.
`SnapshotsCopiedAccountFailed`	Jumlah snapshot yang tidak dapat disalin dari akun lain oleh kebijakan peristiwa penyalinan lintas akun. Ini termasuk percobaan ulang yang berhasil dalam waktu 24 jam dari waktu yang dijadwalkan.
`SnapshotsCopiedAccountDeleteCompleted`	Jumlah salinan snapshot lintas Wilayah yang dihapus, sebagaimana ditetapkan oleh aturan retensi, berdasarkan kebijakan peristiwa penyalinan lintas akun.
`SnapshotsCopiedAccountDeleteFailed`	Jumlah salinan snapshot lintas Wilayah yang tidak dapat dihapus, sebagaimana ditetapkan oleh aturan retensi, berdasarkan kebijakan peristiwa penyalinan lintas akun.

Melihat CloudWatch metrik untuk kebijakan Anda

Anda dapat menggunakan AWS Management Console atau alat baris perintah untuk membuat daftar metrik yang dikirimkan oleh Amazon Data Lifecycle Manager ke Amazon. CloudWatch

Amazon EC2 console

Untuk melihat metrik menggunakan konsol Amazon EC2

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Lifecycle Manager.
3. Pilih kebijakan di grid lalu pilih tab Pemantauan.

CloudWatch console

Untuk melihat metrik menggunakan konsol Amazon CloudWatch

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Di panel navigasi, pilih Metrik.
3. Pilih namespace EBS, kemudian pilih Metrik Data Lifecycle Manager.

AWS CLI

Untuk mencantumkan semua metrik yang tersedia untuk Amazon Data Lifecycle Manager

Gunakan perintah list-metrics.

```
$ C:\> aws cloudwatch list-metrics \
    --namespace AWS/EBS
```

Membuat daftar semua metrik untuk kebijakan tertentu

Gunakan perintah list-metrics dan tentukan dimensi `DLMPolicyId`.

```
$ C:\> aws cloudwatch list-metrics \
    --namespace AWS/EBS \
    --dimensions Name=DLMPolicyId,Value=policy-abcdef01234567890
```

Untuk mencantumkan satu metrik di semua kebijakan

Gunakan perintah list-metrics dan tentukan opsi `--metric-name`.

```
$ C:\> aws cloudwatch list-metrics \
    --namespace AWS/EBS \
```

```
    --metric-name SnapshotsCreateCompleted
```

Grafik metrik untuk kebijakan Anda

Setelah Anda membuat kebijakan, Anda dapat membuka konsol Amazon EC2 dan melihat grafik pemantauan untuk kebijakan tersebut di tab Pemantauan. Setiap grafik didasarkan pada salah satu metrik Amazon EC2 yang tersedia.

Berikut adalah grafik yang tersedia:

- Sumber daya yang ditargetkan (berdasarkan `ResourcesTargeted`)
- Pembuatan snapshot dimulai (berdasarkan `SnapshotsCreateStarted`)
- Pembuatan snapshot selesai (berdasarkan `SnapshotsCreateCompleted`)
- Pembuatan snapshot gagal (berdasarkan `SnapshotsCreateFailed`)
- Pembagian snapshot selesai (berdasarkan `SnapshotsSharedCompleted`)
- Penghapusan snapshot selesai (berdasarkan) `SnapshotsDeleteCompleted`
- Penghapusan snapshot gagal (berdasarkan `SnapshotsDeleteFailed`)
- Penyalinan snapshot lintas Wilayah dimulai (berdasarkan `SnapshotsCopiedRegionStarted`)
- Penyalinan snapshot lintas Wilayah selesai (berdasarkan `SnapshotsCopiedRegionCompleted`)
- Penyalinan snapshot lintas Wilayah gagal (berdasarkan `SnapshotsCopiedRegionFailed`)
- Penghapusan salinan snapshot lintas Wilayah selesai (berdasarkan `SnapshotsCopiedRegionDeleteCompleted`)
- Penghapusan salinan snapshot lintas Wilayah gagal (berdasarkan `SnapshotsCopiedRegionDeleteFailed`)
- Penyalinan snapshot lintas akun dimulai (berdasarkan `SnapshotsCopiedAccountStarted`)
- Salinan snapshot lintas akun selesai (berdasarkan `SnapshotsCopiedAccountCompleted`)
- Penyalinan snapshot lintas akun gagal (berdasarkan `SnapshotsCopiedAccountFailed`)
- Penghapusan salinan lintas akun snapshot selesai (berdasarkan `SnapshotsCopiedAccountDeleteCompleted`)
- Penghapusan salinan snapshot lintas akun gagal (berdasarkan `SnapshotsCopiedAccountDeleteFailed`)
- Pembuatan AMI dimulai (berdasarkan `ImagesCreateStarted`)
- Pembuatan AMI selesai (berdasarkan `ImagesCreateCompleted`)

- Pembuatan AMI gagal (berdasarkan `ImagesCreateFailed`)
- Pembatalan pendaftaran AMI selesai (berdasarkan `ImagesDeregisterCompleted`)
- Pembatalan pendaftaran AMI gagal (berdasarkan `ImagesDeregisterFailed`)
- Penyalinan lintas wilayah AMI dimulai (berdasarkan `ImagesCopiedRegionStarted`)
- Penyalinan lintas wilayah AMI selesai (berdasarkan `ImagesCopiedRegionCompleted`)
- Penyalinan lintas wilayah AMI gagal (berdasarkan `ImagesCopiedRegionFailed`)
- Pembatalan pendaftaran salinan lintas wilayah AMI selesai (berdasarkan `ImagesCopiedRegionDeregisterCompleted`)
- Pembatalan pendaftaran salinan lintas wilayah AMI gagal (berdasarkan `ImagesCopiedRegionDeregisteredFailed`)
- AMI mengaktifkan penghentian selesai (berdasarkan) `EnableImageDeprecationCompleted`
- AMI mengaktifkan penghentian gagal (berdasarkan) `EnableImageDeprecationFailed`
- Salinan lintas wilayah AMI mengaktifkan penghentian selesai (berdasarkan) `EnableCopiedImageDeprecationCompleted`
- AMI Salinan lintas wilayah mengaktifkan penghentian gagal (berdasarkan) `EnableCopiedImageDeprecationFailed`

Buat CloudWatch alarm untuk kebijakan

Anda dapat membuat CloudWatch alarm yang memantau CloudWatch metrik untuk kebijakan Anda. CloudWatch akan secara otomatis mengirimi Anda pemberitahuan ketika metrik mencapai ambang batas yang Anda tentukan. Anda dapat membuat CloudWatch alarm menggunakan CloudWatch konsol.

Untuk informasi selengkapnya tentang membuat alarm menggunakan CloudWatch konsol, lihat topik berikut di Panduan CloudWatch Pengguna Amazon.

- Buat CloudWatch Alarm Berdasarkan Ambang Statis
- Buat CloudWatch Alarm Berdasarkan Deteksi Anomali

Contoh kasus penggunaan

Berikut adalah contoh kasus penggunaan.

Topik

Contoh 1: ResourcesTargeted metrik

Anda dapat menggunakan `ResourcesTargeted` metrik untuk memantau jumlah total sumber daya yang ditargetkan oleh kebijakan tertentu setiap kali dijalankan. Ini memungkinkan Anda untuk memicu alarm ketika jumlah sumber daya yang ditargetkan di bawah atau di atas ambang batas yang diharapkan.

Misalnya, jika Anda mengharapkan kebijakan harian Anda untuk membuat cadangan tidak lebih dari `50` volume, Anda dapat membuat alarm yang mengirimkan notifikasi email ketika `sum` untuk `ResourcesTargeted` lebih besar dari `50` selama periode 1 jam. Dengan cara ini, Anda dapat memastikan bahwa tidak ada snapshot yang dibuat secara tidak terduga dari volume yang salah ditandai.

Anda dapat menggunakan perintah berikut untuk membuat alarm ini:

```
$ C:\> aws cloudwatch put-metric-alarm \
    --alarm-name resource-targeted-monitor \
    --alarm-description "Alarm when policy targets more than 50 resources" \
    --metric-name ResourcesTargeted \
    --namespace AWS/EBS \
    --statistic Sum \
    --period 3600 \
    --threshold 50 \
    --comparison-operator GreaterThanThreshold \
    --dimensions "Name=DLMPolicyId,Value=policy_id" \
    --evaluation-periods 1 \
    --alarm-actions sns_topic_arn
```

Contoh 2: SnapshotDeleteFailed metrik

Anda dapat menggunakan metrik `SnapshotDeleteFailed` untuk memantau kegagalan menghapus snapshot sesuai aturan retensi snapshot kebijakan.

Misalnya, jika Anda telah membuat kebijakan yang akan menghapus snapshot secara otomatis setiap dua belas jam, Anda dapat membuat alarm yang memberi tahu tim rekayasa Anda ketika `sum` dari `SnapshotDeletionFailed` lebih besar dari `0` selama periode 1 jam. Hal ini dapat membantu

menyelidiki retensi snapshot yang tidak tepat dan memastikan bahwa biaya penyimpanan Anda tidak bertambah karena snapshot yang tidak diperlukan.

Anda dapat menggunakan perintah berikut untuk membuat alarm ini:

```
$ C:\> aws cloudwatch put-metric-alarm \
    --alarm-name snapshot-deletion-failed-monitor \
    --alarm-description "Alarm when snapshot deletions fail" \
    --metric-name SnapshotsDeleteFailed \
    --namespace AWS/EBS \
    --statistic Sum \
    --period 3600 \
    --threshold 0 \
    --comparison-operator GreaterThanThreshold \
    --dimensions "Name=DLMPolicyId,Value=policy_id" \
    --evaluation-periods 1 \
    --alarm-actions sns_topic_arn
```

Contoh 3: SnapshotsCopiedRegionFailed metrik

Gunakan metrik `SnapshotsCopiedRegionFailed` untuk mengidentifikasi kapan kebijakan Anda gagal menyalin snapshot ke Wilayah lain.

Misalnya, jika kebijakan Anda menyalin snapshot di seluruh Wilayah setiap hari, Anda dapat membuat alarm yang mengirimkan SMS ke tim rekayasa Anda ketika `sum` dari `SnapshotCrossRegionCopyFailed` lebih besar dari `0` selama periode 1 jam. Hal ini dapat berguna untuk memverifikasi apakah snapshot berikutnya dalam garis keturunan berhasil disalin oleh kebijakan.

Anda dapat menggunakan perintah berikut untuk membuat alarm ini:

```
$ C:\> aws cloudwatch put-metric-alarm \
    --alarm-name snapshot-copy-region-failed-monitor \
    --alarm-description "Alarm when snapshot copy fails" \
    --metric-name SnapshotsCopiedRegionFailed \
    --namespace AWS/EBS \
    --statistic Sum \
    --period 3600 \
    --threshold 0 \
    --comparison-operator GreaterThanThreshold \
    --dimensions "Name=DLMPolicyId,Value=policy_id" \
    --evaluation-periods 1 \
```

```
    --alarm-actions sns_topic_arn
```

Mengelola kebijakan yang melaporkan tindakan gagal

Untuk informasi selengkapnya tentang apa yang harus dilakukan ketika salah satu kebijakan Anda melaporkan nilai bukan nol yang tidak terduga untuk metrik tindakan yang gagal, lihat Apa yang harus saya lakukan jika Amazon Data Lifecycle Manager melaporkan tindakan gagal dalam metrik? CloudWatch AWS Artikel Pusat Pengetahuan.

Pemecahan Masalah

Dokumentasi berikut dapat membantu Anda memecahkan masalah yang mungkin terjadi.

Topik

- Kesalahan: Role with name already exists

Kesalahan: **`Role with name already exists`**

Deskripsi

Anda akan `Role with name AWSDataLifecycleManagerDefaultRole already exists` menemukan `Role with name AWSDataLifecycleManagerDefaultRoleForAMIManagement already exists` kesalahan saat mencoba membuat kebijakan menggunakan konsol.

Penyebab

Format ARN peran berbeda bergantung pada apakah itu dibuat menggunakan konsol atau AWS CLI. Meskipun ARN berbeda, peran menggunakan nama peran yang sama, yang menghasilkan konflik penamaan peran antara konsol dan AWS CLI.

Solusi

Untuk mengatasi masalah ini, lakukan solusi berikut:

1. (Untuk kebijakan snapshot yang diaktifkan hanya untuk skrip pra dan pasca) Lampirkan kebijakan AWSDataLifecycleManagerSSMFullAccess AWS terkelola secara manual ke peran AWSDataLifecycleManagerDefaultRoleIAM. Untuk informasi selengkapnya, lihat Menambahkan izin identitas IAM.

2. Saat membuat kebijakan Amazon Data Lifecycle Manager, untuk peran IAM, pilih Pilih peran lain, lalu pilih (untuk kebijakan snapshot), atau AWSDataLifecycleManagerDefaultRole(AWSDataLifecycleManagerDefaultRoleForAMIManagementuntu kebijakan AMI).
3. Terus buat kebijakan seperti biasa.

Menggunakan API langsung EBS untuk mengakses konten snapshot EBS

Anda dapat menggunakan API langsung Amazon Elastic Block Store (Amazon EBS) untuk membuat snapshot EBS, menulis data secara langsung ke snapshot, membaca data di snapshot, dan mengidentifikasi perbedaan atau perubahan di antara dua snapshot. Jika Anda adalah vendor perangkat lunak independen (ISV) yang menawarkan layanan pencadangan untuk Amazon EBS, API langsung EBS membuat pelacakan perubahan inkremental pada volume EBS Anda melalui snapshot menjadi lebih efisien dan hemat biaya. Hal ini dapat dilakukan tanpa harus membuat volume baru dari snapshot, lalu gunakan instans Amazon Elastic Compute Cloud (Amazon EC2) untuk membandingkan perbedaannya.

Anda dapat membuat snapshot inkremental secara langsung dari data on-premise ke volume EBS dan cloud untuk digunakan dalam pemulihan bencana cepat. Dengan kemampuan untuk menulis dan membaca snapshot, Anda dapat menulis data on-premise ke snapshot EBS selama terjadi bencana. Kemudian setelah pemulihan, Anda dapat memulihkannya kembali ke AWS atau lokal dari snapshot. Anda tidak perlu lagi membangun dan memelihara mekanisme yang rumit untuk menyalin data ke dan dari Amazon EBS.

Panduan pengguna ini memberikan gambaran umum tentang elemen-elemen yang membentuk API langsung EBS, dan contoh cara menggunakannya secara efektif. Untuk informasi selengkapnya tentang tindakan, jenis data, parameter, dan kesalahan API, lihat referensi API langsung EBS. Untuk informasi selengkapnya tentang AWS Wilayah, titik akhir, dan kuota layanan yang didukung untuk API langsung EBS, lihat titik akhir dan kuota Amazon EBS di. Referensi Umum AWS

Daftar Isi

Memahami API langsung EBS

Berikut ini adalah elemen utama yang harus Anda pahami sebelum memulai dengan API langsung EBS.

Snapshot

Snapshot adalah sarana utama untuk mencadangkan data dari volume EBS Anda. Dengan API langsung EBS, Anda juga dapat mencadangkan data dari disk on-premise ke snapshot. Untuk menghemat biaya penyimpanan, snapshot berikutnya bersifat bertahap, hanya berisi data volume yang berubah sejak snapshot sebelumnya. Untuk informasi selengkapnya, lihat Snapshot Amazon EBS.

> **Note**
>
> API langsung EBS tidak mendukung snapshot publik dan snapshot lokal di Outposts.

Blok

Blok adalah fragmen data di dalam snapshot. Setiap snapshot dapat berisi ribuan blok. Semua blok dalam snapshot memiliki ukuran tetap.

Indeks blok

Indeks blok adalah indeks logis dalam satuan blok 512 KiB. Untuk mengidentifikasi indeks blok, bagilah offset logis data dalam volume logis dengan ukuran blok (offset logis data/524288). Offset logis dari data harus disesuaikan dengan 512 KiB.

Token blok

Token blok adalah hash pengidentifikasi dari sebuah blok di dalam sebuah snapshot, dan digunakan untuk menemukan data blok. Token blok yang dikembalikan oleh API langsung EBS bersifat sementara. Mereka berubah pada stempel waktu kedaluwarsa yang ditentukan untuk mereka, atau jika Anda menjalankan yang lain ListSnapshotBlocks atau ListChangedBlocks meminta snapshot yang sama.

Checksum

Checksum adalah datum berukuran kecil yang berasal dari blok data untuk tujuan mendeteksi kesalahan yang diperkenalkan selama transmisi atau penyimpanan. API langsung EBS menggunakan checksum untuk memvalidasi integritas data. Saat Anda membaca data dari snapshot EBS, layanan ini memberikan checksum SHA256 dengan enkode Base64 untuk setiap blok data yang ditransmisikan, yang dapat Anda gunakan untuk validasi. Saat Anda menulis data ke snapshot EBS, Anda harus menyediakan checksum SHA256 yang dienkode Base64 untuk setiap blok data yang ditransmisikan. Layanan memvalidasi data yang diterima menggunakan checksum yang disediakan. Untuk informasi selengkapnya, lihat Gunakan checksum dalam panduan ini.

Enkripsi

Enkripsi melindungi data Anda dengan mengubahnya menjadi kode yang tidak terbaca yang hanya dapat diuraikan oleh orang yang memiliki akses ke kunci KMS yang digunakan untuk mengenkripsinya. Anda dapat menggunakan API langsung EBS untuk membaca dan menulis snapshot terenkripsi, tetapi ada beberapa batasan. Untuk informasi selengkapnya, lihat Gunakan enkripsi dalam panduan ini.

Tindakan API

EBS langsung terdiri dari enam tindakan. Ada tiga tindakan baca dan tiga tindakan tulis. Tindakan baca adalah:

- ListSnapshotBlok — mengembalikan indeks blok dan blok token blok dalam snapshot yang ditentukan
- ListChangedBlok — mengembalikan indeks blok dan token blok blok yang berbeda antara dua snapshot tertentu dari volume yang sama dan garis keturunan snapshot.
- GetSnapshotBlok — mengembalikan data dalam blok untuk ID snapshot yang ditentukan, indeks blok, dan token blok.

Tindakan tulis adalah:

- StartSnapshot— memulai snapshot, baik sebagai snapshot tambahan dari yang sudah ada atau sebagai snapshot baru. Snapshot yang dimulai tetap dalam status tertunda hingga selesai menggunakan CompleteSnapshot tindakan.
- PutSnapshotBlok — menambahkan data ke snapshot yang dimulai dalam bentuk blok individual. Anda harus menentukan checksum SHA256 berkode Base64 untuk blok data yang . Layanan

memvalidasi checksum setelah transmisi selesai. Permintaan gagal jika checksum yang dihitung oleh layanan tidak sesuai dengan yang Anda tentukan.

- CompleteSnapshot— menyelesaikan snapshot yang dimulai yang dalam keadaan tertunda. Snapshot lalu diubah ke status selesai.

Izin IAM untuk API langsung EBS

Seorang pengguna harus memiliki kebijakan berikut untuk menggunakan API langsung EBS. Untuk informasi selengkapnya, lihat Mengubah izin untuk pengguna.

Untuk informasi selengkapnya tentang sumber daya, tindakan, dan kunci konteks syarat API langsung EBS untuk digunakan dalam kebijakan izin IAM, lihat Tindakan, sumber daya, dan kunci syarat untuk Amazon Elastic Block Store di Referensi Otorisasi Layanan.

> **Important**
>
> Berhati-hatilah saat menetapkan kebijakan berikut kepada pengguna. Dengan menetapkan kebijakan ini, Anda dapat memberikan akses ke pengguna yang ditolak akses ke sumber daya yang sama melalui API Amazon EC2, seperti CopySnapshot tindakan atau. CreateVolume

Izin untuk membaca snapshot

Kebijakan berikut memungkinkan API langsung EBS baca digunakan pada semua snapshot di Wilayah tertentu AWS . Dalam kebijakan, ganti *<Wilayah>* dengan Wilayah dari snapshot.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ebs:ListSnapshotBlocks",
                "ebs:ListChangedBlocks",
                "ebs:GetSnapshotBlock"
            ],
            "Resource": "arn:aws:ec2:<Region>::snapshot/*"
        }
```

```
    ]
}
```

Kebijakan berikut mengizinkan pembacaan API langsung EBS untuk digunakan pada snapshot dengan tanda nilai kunci tertentu. Dalam kebijakan, ganti *<Kunci>* dengan nilai kunci dari tanda, dan *<Nilai>* dengan nilai dari tanda.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ebs:ListSnapshotBlocks",
                "ebs:ListChangedBlocks",
                "ebs:GetSnapshotBlock"
            ],
            "Resource": "arn:aws:ec2:*::snapshot/*",
            "Condition": {
                "StringEqualsIgnoreCase": {
                    "aws:ResourceTag/<Key>": "<Value>"
                }
            }
        }
    ]
}
```

Kebijakan berikut mengizinkan semua pembacaan API langsung EBS untuk digunakan pada semua snapshot dalam akun tersebut hanya pada rentang waktu tertentu. Kebijakan ini mengizinkan penggunaan API langsung EBS berdasarkan pada kunci syarat global `aws:CurrentTime`. Dalam kebijakan tersebut, pastikan Anda mengganti rentang tanggal dan waktu yang ditampilkan sesuai rentang tanggal dan waktu untuk kebijakan Anda.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ebs:ListSnapshotBlocks",
                "ebs:ListChangedBlocks",
                "ebs:GetSnapshotBlock"
```

```
            ],
            "Resource": "arn:aws:ec2:*::snapshot/*",
            "Condition": {
                "DateGreaterThan": {
                    "aws:CurrentTime": "2018-05-29T00:00:00Z"
                },
                "DateLessThan": {
                    "aws:CurrentTime": "2020-05-29T23:59:59Z"
                }
            }
        }
    ]
}
```

Untuk informasi selengkapnya, lihat Mengubah izin untuk pengguna pada Panduan Pengguna IAM.

Izin untuk menulis snapshot

Kebijakan berikut memungkinkan API langsung EBS tulis digunakan pada semua snapshot di Wilayah tertentu AWS . Dalam kebijakan, ganti *<Wilayah>* dengan Wilayah dari snapshot.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ebs:StartSnapshot",
                "ebs:PutSnapshotBlock",
                "ebs:CompleteSnapshot"
            ],
            "Resource": "arn:aws:ec2:<Region>::snapshot/*"
        }
    ]
}
```

Kebijakan berikut mengizinkan penulisan API langsung EBS untuk digunakan pada snapshot dengan tanda nilai kunci tertentu. Dalam kebijakan, ganti *<Kunci>* Dengan nilai kunci dari tanda, dan *<Nilai>* dengan nilai dari tanda.

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

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ebs:StartSnapshot",
                "ebs:PutSnapshotBlock",
                "ebs:CompleteSnapshot"
            ],
            "Resource": "arn:aws:ec2:*::snapshot/*",
            "Condition": {
                "StringEqualsIgnoreCase": {
                    "aws:ResourceTag/<Key>": "<Value>"
                }
            }
        }
    ]
}
```

Kebijakan berikut mengizinkan seluruh API langsung EBS untuk digunakan. Hal ini juga memungkinkan tindakan `StartSnapshot` hanya jika ID snapshot induk ditentukan. Oleh karena itu, kebijakan ini memblokir kemampuan untuk memulai snapshot baru menggunakan snapshot induk.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ebs:*",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "ebs:ParentSnapshot": "arn:aws:ec2:*::snapshot/*"
                }
            }
        }
    ]
}
```

Kebijakan berikut mengizinkan seluruh API langsung EBS untuk digunakan. Kebijakan tersebut juga hanya memungkinkan kunci tanda `user` dibuat untuk snapshot baru. Kebijakan ini juga memastikan bahwa pengguna memiliki akses untuk membuat tanda. Tindakan `StartSnapshot` adalah satu-satunya tindakan yang dapat menentukan tanda.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ebs:*",
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": "user"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "*"
        }
    ]
}
```

Kebijakan berikut mengizinkan semua penulisan API langsung EBS untuk digunakan pada semua snapshot dalam akun tersebut hanya pada rentang waktu tertentu. Kebijakan ini mengizinkan penggunaan API langsung EBS berdasarkan pada kunci syarat global `aws:CurrentTime`. Dalam kebijakan tersebut, pastikan Anda mengganti rentang tanggal dan waktu yang ditampilkan sesuai rentang tanggal dan waktu untuk kebijakan Anda.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ebs:StartSnapshot",
                "ebs:PutSnapshotBlock",
                "ebs:CompleteSnapshot"
            ],
            "Resource": "arn:aws:ec2:*::snapshot/*",
            "Condition": {
                "DateGreaterThan": {
                    "aws:CurrentTime": "2018-05-29T00:00:00Z"
                },
```

```
                "DateLessThan": {
                    "aws:CurrentTime": "2020-05-29T23:59:59Z"
                }
            }
        }
    ]
}
```

Untuk informasi selengkapnya, lihat Mengubah izin untuk pengguna pada Panduan Pengguna IAM.

Izin untuk digunakan AWS KMS keys

Kebijakan berikut memberikan izin untuk mendekripsi snapshot terenkripsi menggunakan kunci KMS tertentu. Kebijakan ini juga memberikan izin untuk mengenkripsi snapshot baru menggunakan kunci KMS default untuk enkripsi EBS. Dalam kebijakan, ganti <Region>dengan Region kunci KMS, *< AccountId >* dengan ID AWS akun kunci KMS, dan *< KeyId >* dengan ID kunci KMS.

> **Note**
>
> Secara default, semua prinsipal di akun memiliki akses ke kunci KMS AWS terkelola default untuk enkripsi Amazon EBS, dan mereka dapat menggunakannya untuk operasi enkripsi dan dekripsi EBS. Jika Anda menggunakan kunci yang dikelola pelanggan, Anda harus membuat kebijakan kunci baru atau memodifikasi kebijakan kunci yang ada untuk kunci yang dikelola pelanggan untuk memberi pengguna utama akses utama ke kunci yang dikelola pelanggan. Untuk informasi selengkapnya, lihat Kebijakan kunci di AWS KMS di Panduan Developer AWS Key Management Service .

> **Tip**
>
> Untuk mengikuti prinsip hak akses paling rendah, jangan biarkan akses penuh ke `kms:CreateGrant`. Sebagai gantinya, gunakan tombol `kms:GrantIsForAWSResource` kondisi untuk memungkinkan pengguna membuat hibah pada kunci KMS hanya ketika hibah dibuat atas nama pengguna oleh AWS layanan, seperti yang ditunjukkan pada contoh berikut.

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:GenerateDataKey",
                "kms:GenerateDataKeyWithoutPlaintext",
                "kms:ReEncrypt*",
                "kms:CreateGrant",
                "ec2:CreateTags",
                "kms:DescribeKey"
            ],
            "Resource": "arn:aws:kms:<Region>:<AccountId>:key/<KeyId>",
            "Condition": {
                "Bool": {
                    "kms:GrantIsForAWSResource": true
                }
            }
        }
    ]
}
```

Untuk informasi selengkapnya, lihat Mengubah izin untuk pengguna pada Panduan Pengguna IAM.

Menggunakan API langsung EBS

Topik berikut menunjukkan cara membaca dan menulis snapshot menggunakan API langsung EBS. Anda dapat membaca dan menulis snapshot hanya menggunakan AWS API AWS CLI, dan AWS SDK. Untuk informasi selengkapnya, lihat:

- Menginstal AWS CLI dan Mengkonfigurasi AWS CLI
- Referensi API langsung EBS
- AWS SDK

> **Important**
>
> API langsung EBS memerlukan AWS tanda tangan Versi Tanda Tangan 4. Untuk informasi selengkapnya, lihat Gunakan tanda tangan Signature Versi 4.

Topik

Membaca snapshot dengan API langsung EBS

Langkah-langkah berikut menjelaskan cara menggunakan API langsung EBS untuk membaca snapshot:

1. Gunakan ListSnapshotBlocks tindakan untuk melihat semua indeks blok dan memblokir token blok dalam snapshot. Atau gunakan ListChangedBlocks tindakan untuk hanya melihat indeks blok dan token blok blok yang berbeda antara dua snapshot dengan volume yang sama dan garis keturunan snapshot. Tindakan ini membantu Anda mengidentifikasi token blok dan indeks blok dari blok yang mungkin ingin Anda dapatkan datanya.
2. Gunakan GetSnapshotBlock tindakan, dan tentukan indeks blok dan token blok blok yang ingin Anda dapatkan datanya.

Topik berikut menunjukkan cara membaca dan menulis snapshot menggunakan API langsung EBS.

Topik

- Blok daftar yang berbeda antara dua snapshot
- Dapatkan data blok dari snapshot

Mencantumkan blok dalam snapshot

AWS CLI

Contoh perintah list-snapshot-blocks berikut mengembalikan indeks blok dan token blok dari blok yang berada dalam snapshot `snap-0987654321`. Parameter `--starting-block-index` membatasi hasil untuk memblokir indeks yang lebih besar dari `1000`, dan parameter `--max-results` membatasi hasil untuk `100` blok pertama.

```
aws ebs list-snapshot-blocks --snapshot-id snap-0987654321 --starting-block-
index 1000 --max-results 100
```

Contoh respons berikut untuk perintah sebelumnya mencantumkan indeks blok dan blok token dalam snapshot. Gunakan `get-snapshot-block` memerintahkan dan menentukan indeks blok dan token blok dari blok yang ingin Anda dapatkan datanya. Token blok berlaku hingga waktu kedaluwarsa yang tercantum.

```
{
      "Blocks": [
          {
              "BlockIndex": 1001,
              "BlockToken": "AAABAV3/
PNhXOynVdMYHUpPsetaSvjLB1dtIGfbJv5OJ0sX855EzGTWos4a4"
          },
          {
              "BlockIndex": 1002,
              "BlockToken": "AAABATGQIgwr0WwIuqIMjCA/Sy7e/
YoQFZsHejzGNvjKauzNgzeI13YHBfQB"
          },
          {
              "BlockIndex": 1007,
              "BlockToken": "AAABAZ9CTuQtUvp/
dXqRWw4d07eOgTZ3jvn6hiW30W9duM8MiMw6yQayzF2c"
          },
          {
              "BlockIndex": 1012,
              "BlockToken": "AAABAQdzxhw0rVV6PNmsfo/
YRIxo9JPR85XxPf1BLjg0Hec6pygYr6laE1p0"
```

```
          },
          {
              "BlockIndex": 1030,
              "BlockToken": "AAABAaYvPax6mv+iGWLdTUjQtFWouQ7Dqz6nSD9L
+CbXnvpkswA6iDID523d"
          },
          {
              "BlockIndex": 1031,
              "BlockToken": "AAABATgWZC0XcFwUKvTJbUXMiSPg59KVxJGL
+BWBClkw6spzCxJVqDVaTskJ"
          },
          ...
      ],
      "ExpiryTime": 1576287332.806,
      "VolumeSize": 32212254720,
      "BlockSize": 524288
  }
```

AWS API

Permintaan contoh ListSnapshotBlok berikut mengembalikan indeks blok dan token blok blok yang ada di snapshotsnap-0acEXAMPLEcf41648. Parameter `startingBlockIndex` membatasi hasil untuk memblokir indeks yang lebih besar dari `1000`, dan parameter `maxResults` membatasi hasil untuk `100` blok pertama.

```
GET /snapshots/snap-0acEXAMPLEcf41648/blocks?maxResults=100&startingBlockIndex=1000
 HTTP/1.1
  Host: ebs.us-east-2.amazonaws.com
  Accept-Encoding: identity
  User-Agent: <User agent parameter>
  X-Amz-Date: 20200617T231953Z
  Authorization: <Authentication parameter>
```

Contoh respons berikut untuk permintaan sebelumnya mencantumkan indeks blok dan blok token dalam snapshot. Gunakan GetSnapshotBlock tindakan dan tentukan indeks blok dan token blok blok yang ingin Anda dapatkan datanya. Token blok berlaku hingga waktu kedaluwarsa yang tercantum.

```
HTTP/1.1 200 OK
  x-amzn-RequestId: d6e5017c-70a8-4539-8830-57f5557f3f27
  Content-Type: application/json
  Content-Length: 2472
```

```
  Date: Wed, 17 Jun 2020 23:19:56 GMT
  Connection: keep-alive

  {
      "BlockSize": 524288,
      "Blocks": [
          {
              "BlockIndex": 0,
              "BlockToken": "AAUBAcuWqOCnDNuKle11s7IIX6jp6FYcC/q8oT93913HhvLvA
+3JRrSybp/0"
          },
          {
              "BlockIndex": 1536,
              "BlockToken":
 "AAUBAWudwfmofcrQhGVlLwuRKm2b8ZXPiyrgoykTRC6IU1NbxKWDY1pPjvnV"
          },
          {
              "BlockIndex": 3072,
              "BlockToken":
 "AAUBAV7p6pC5fKAC7TokoNCtAnZhqq27u6YEXZ3MwRevBkDjmMx6iuA6tsBt"
          },
          {
              "BlockIndex": 3073,
              "BlockToken":
 "AAUBAbqt9zpqBUEvtO2HINAfFaWToOwlPjbIsQOlx6JUN/0+iMQl0NtNbnX4"
          },
          ...
      ],
      "ExpiryTime": 1.59298379649E9,
      "VolumeSize": 3
  }
```

Blok daftar yang berbeda antara dua snapshot

Ingatlah hal berikut saat membuat permintaan paginasi untuk mencantumkan daftar blok yang diubah di antara dua snapshot:

- Respons dapat mencakup satu atau beberapa array `ChangedBlocks` yang kosong. Misalnya:
  - Snapshot 1 — snapshot penuh dengan 1000 blok dengan `0` - 999 indeks blok.
  - Snapshot 2 — snapshot inkremental dengan hanya satu blok yang diubah dengan 999 indeks blok.

Daftar blok yang diubah untuk snapshot ini dengan `StartingBlockIndex = 0` dan `MaxResults = 100` mengembalikan array `ChangedBlocks` yang kosong. Anda harus meminta hasil yang tersisa menggunakan `nextToken` sampai blok yang diubah dikembalikan dalam set hasil kesepuluh, yang mencakup blok dengan indeks blok `900` - 999.

- Respons dapat melewati blok yang tidak tertulis dalam snapshot. Misalnya:
  - Snapshot 1 — snapshot penuh dengan 1000 blok dengan `2000` - 2999 indeks blok.
  - Snapshot 2 — snapshot inkremental dengan hanya satu blok yang diubah dengan `2000` indeks blok.

  Dengan mendaftar blok yang diubah untuk snapshot ini dengan `StartingBlockIndex = 0` dan `MaxResults = 100`, responsnya akan melewati `0` - `1999` indeks blok dan menyertakan `2000` indeks blok. Respons tidak akan menyertakan array `ChangedBlocks` yang kosong.

AWS CLI

Contoh perintah list-changed-blocks berikut mengembalikan indeks blok dan token blok dari blok yang berbeda antara snapshot `snap-1234567890` dan `snap-0987654321`. Parameter `--starting-block-index` membatasi hasil untuk indeks bloks yang lebih besar dari `0`, dan parameter `--max-results` membatasi hasil untuk `500` blok pertama.

```
aws ebs list-changed-blocks --first-snapshot-id snap-1234567890 --second-snapshot-
id snap-0987654321 --starting-block-index 0 --max-results 500
```

Contoh respons berikut untuk perintah sebelumnya menunjukkan bahwa indeks blok 0, 6000, 6001, 6002, dan 6003 berbeda di antara dua snapshot. Selain itu, indeks blok 6001, 6002, dan 6003 hanya ada dalam ID snapshot pertama yang ditentukan, dan tidak dalam ID snapshot kedua karena tidak ada token blok kedua yang tercantum dalam respons.

Gunakan perintah `get-snapshot-block` dan tentukan indeks blok dan token blok dari blok yang ingin Anda dapatkan datanya. Token blok berlaku hingga waktu kedaluwarsa yang tercantum.

```
{
      "ChangedBlocks": [
          {
              "BlockIndex": 0,
              "FirstBlockToken": "AAABAVahm9SO60Dyi0ORySzn2ZjGjW/
KN3uygGlS0QOYWesbzBbDnX2dGpmC",
```

```
                "SecondBlockToken":
 "AAABAf8o0o6UFi1rDbSZGIRaCEdDyBu9TlvtCQxxoKV8qrUPQP7vcM6iWGSr"
            },
            {
                "BlockIndex": 6000,
                "FirstBlockToken": "AAABAbYSiZvJ0/
R9tz8suI8dSzecLjN4kkazK8inFXVintPkdaVFLfCMQsKe",
                "SecondBlockToken":
 "AAABAZnqTdzFmKRpsaMAsDxviVqEI/3jJzI2crq2eFDCgHmyNf777elD9oVR"
            },
            {
                "BlockIndex": 6001,
                "FirstBlockToken": "AAABASBpSJ2UAD3PLxJnCt6zun4/
T4sU25Bnb8jB5Q6FRXHFqAIAqE04hJoR"
            },
            {
                "BlockIndex": 6002,
                "FirstBlockToken": "AAABASqX4/
NWjvNceoyMUljcRd0DnwbSwNnes1UkoP62CrQXvn47BY5435aw"
            },
            {
                "BlockIndex": 6003,
                "FirstBlockToken":
 "AAABASmJ0O5JxAOce25rF4P1sdRtyIDsX12tFEDunnePYUKOf4PBROuICb2A"
            },
            ...
        ],
        "ExpiryTime": 1576308931.973,
        "VolumeSize": 32212254720,
        "BlockSize": 524288,
        "NextToken": "AAADARqElNng/sV98CYk/bJDCXeLJmLJHnNSkHvLzVaO0zsPH/QM3Bi3zF//
O6Mdi/BbJarBnp8h"
    }
```

AWS API

Permintaan contoh ListChangedBlok berikut mengembalikan indeks blok dan token blok blok yang berbeda antara snapshot `snap-0acEXAMPLEcf41648` dan. `snap-0c9EXAMPLE1b30e2f` Parameter `startingBlockIndex` membatasi hasil untuk memblokir indeks yang lebih besar dari `0`, dan parameter `maxResults` membatasi hasil untuk `500` blok pertama.

```
GET /snapshots/snap-0c9EXAMPLE1b30e2f/changedblocks?
firstSnapshotId=snap-0acEXAMPLEcf41648&maxResults=500&startingBlockIndex=0 HTTP/1.1
```

```
  Host: ebs.us-east-2.amazonaws.com
  Accept-Encoding: identity
  User-Agent: <User agent parameter>
  X-Amz-Date: 20200617T232546Z
  Authorization: <Authentication parameter>
```

Contoh respons berikut untuk perintah sebelumnya menunjukkan bahwa indeks blok `0`, `3072`, `6002`, dan `6003` berbeda di antara dua snapshot. Selain itu, indeks blok `6002` dan `6003` hanya ada dalam ID snapshot pertama yang ditentukan, dan tidak dalam ID snapshot kedua karena tidak ada token blok kedua yang tercantum dalam respons.

Gunakan tindakan `GetSnapshotBlock`, dan tentukan indeks blok serta token blok dari blok yang ingin Anda dapatkan datanya. Token blok berlaku hingga waktu kedaluwarsa yang tercantum.

```
HTTP/1.1 200 OK
  x-amzn-RequestId: fb0f6743-6d81-4be8-afbe-db11a5bb8a1f
  Content-Type: application/json
  Content-Length: 1456
  Date: Wed, 17 Jun 2020 23:25:47 GMT
  Connection: keep-alive

  {
      "BlockSize": 524288,
      "ChangedBlocks": [
          {
              "BlockIndex": 0,
              "FirstBlockToken": "AAUBAVaWqOCnDNuKle11s7IIX6jp6FYcC/
tJuVT1GgP23AuLntwiMdJ+OJkL",
              "SecondBlockToken": "AAUBASxzy0Y0b33JVRLoYm3NOresCxn5RO+HVFzXW3Y/
RwfFaPX2Edx8QHCh"
          },
          {
              "BlockIndex": 3072,
              "FirstBlockToken":
 "AAUBAcHp6pC5fKAC7TokoNCtAnZhqq27u6fxRfZOLEmeXLmHBf2R/Yb24MaS",
              "SecondBlockToken":
 "AAUBARGCaufCqBRZC8tEkPYGGkSv3vqvOjJ2xKDi3ljDFiytUxBLXYgTmkid"
          },
          {
              "BlockIndex": 6002,
              "FirstBlockToken": "AAABASqX4/
NWjvNceoyMUljcRd0DnwbSwNnes1UkoP62CrQXvn47BY5435aw"
          },
```

```
        {
            "BlockIndex": 6003,
            "FirstBlockToken":
"AAABASmJ0O5JxAOce25rF4P1sdRtyIDsX12tFEDunnePYUKOf4PBROuICb2A"
        },
        ...
    ],
    "ExpiryTime": 1.592976647009E9,
    "VolumeSize": 3
 }
```

Dapatkan data blok dari snapshot

AWS CLI

Contoh perintah get-snapshot-block berikut mengembalikan data dalam indeks blok `6001` dengan token blok `AAABASBpSJ2UAD3PLxJnCt6zun4/T4sU25Bnb8jB5Q6FRXHFqAIAqE04hJoR`, dalam snapshot `snap-1234567890`. Data biner adalah output file `data` dalam direktori `C:\Temp` pada komputer Windows. Jika Anda menjalankan perintah di komputer Linux atau Unix, ganti jalur output dengan `/tmp/data` untuk mengeluarkan data ke file `data` dalam direktori `/tmp`.

```
aws ebs get-snapshot-block --snapshot-id snap-1234567890 --block-index 6001 --block-
token AAABASBpSJ2UAD3PLxJnCt6zun4/T4sU25Bnb8jB5Q6FRXHFqAIAqE04hJoR C:/Temp/data
```

Contoh respons berikut untuk perintah sebelumnya menunjukkan ukuran data yang dikembalikan, checksum untuk memvalidasi data, dan algoritma checksum. Data biner secara otomatis disimpan ke direktori dan file yang Anda tentukan dalam perintah permintaan.

```
{
      "DataLength": "524288",
      "Checksum": "cf0Y6/Fn0oFa4VyjQPOa/iD0zhTflPTKzxGv2OKowXc=",
      "ChecksumAlgorithm": "SHA256"
  }
```

AWS API

Permintaan contoh GetSnapshotBlock berikut mengembalikan data dalam indeks blok `3072` dengan token blokAAUBARGCaufCqBRZC8tEkPYGGkSv3vqvOjJ2xKDi3ljDFiytUxBLXYgTmkid, dalam snapshot`snap-0c9EXAMPLE1b30e2f`.

```
GET /snapshots/snap-0c9EXAMPLE1b30e2f/blocks/3072?
blockToken=AAUBARGCaufCqBRZC8tEkPYGGkSv3vqvOjJ2xKDi3ljDFiytUxBLXYgTmkid HTTP/1.1
  Host: ebs.us-east-2.amazonaws.com
  Accept-Encoding: identity
  User-Agent: <User agent parameter>
  X-Amz-Date: 20200617T232838Z
  Authorization: <Authentication parameter>
```

Contoh respons berikut untuk perintah sebelumnya menunjukkan ukuran data yang dikembalikan, checksum untuk memvalidasi data, dan algoritma checksum. Data biner ditransmisikan dalam tubuh respons dan direpresentasikan seperti *BlockData*pada contoh berikut.

```
HTTP/1.1 200 OK
  x-amzn-RequestId: 2d0db2fb-bd88-474d-a137-81c4e57d7b9f
  x-amz-Data-Length: 524288
  x-amz-Checksum: Vc0yY2j3qg8bUL9I6GQuI2orTudrQRBDMIhcy7bdEsw=
  x-amz-Checksum-Algorithm: SHA256
  Content-Type: application/octet-stream
  Content-Length: 524288
  Date: Wed, 17 Jun 2020 23:28:38 GMT
  Connection: keep-alive

  BlockData
```

Menulis snapshot dengan API langsung EBS

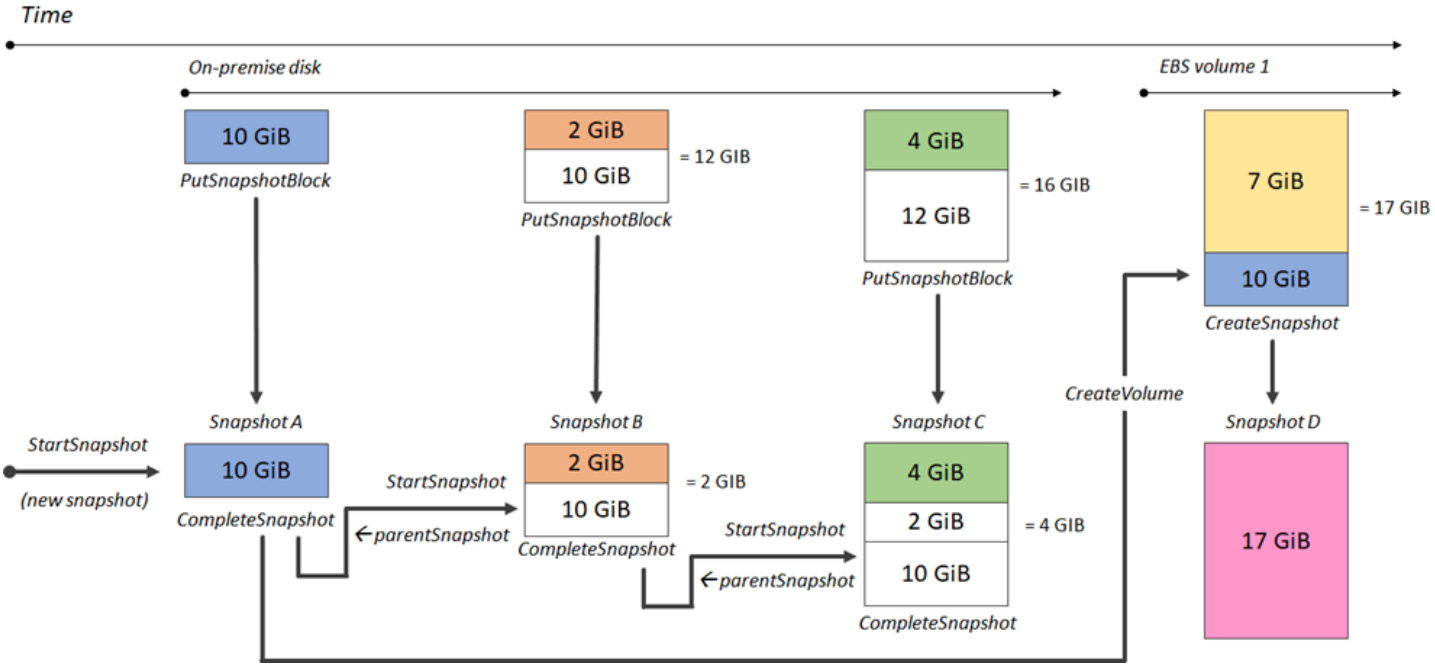
Langkah-langkah berikut menjelaskan cara menggunakan API langsung EBS untuk menulis snapshot inkremental:

1. Gunakan StartSnapshot tindakan dan tentukan ID snapshot induk untuk memulai snapshot sebagai snapshot tambahan dari yang sudah ada, atau hilangkan ID snapshot induk untuk memulai snapshot baru. Tindakan ini mengembalikan ID snapshot baru, yang berada dalam status tertunda.
2. Gunakan PutSnapshotBlock tindakan dan tentukan ID snapshot yang tertunda untuk menambahkan data ke dalamnya dalam bentuk blok individual. Anda harus menentukan checksum SHA256 yang dienkode Base64 untuk blok data yang ditransmisikan. Layanan ini menghitung checksum data yang diterima dan memvalidasinya dengan checksum yang Anda tentukan. Tindakan gagal jika checksum tidak cocok.

3. Setelah selesai menambahkan data ke snapshot yang tertunda, gunakan CompleteSnapshot tindakan untuk memulai alur kerja asinkron yang menyegel snapshot dan memindahkannya ke status selesai.

Ulangi langkah-langkah ini untuk membuat snapshot inkremental baru menggunakan snapshot yang dibuat sebelumnya sebagai induk.

Misalnya, dalam diagram berikut, snapshot A adalah snapshot baru pertama yang dimulai. Snapshot A digunakan sebagai snapshot induk untuk memulai snapshot B. Snapshot B digunakan sebagai snapshot induk untuk memulai dan membuat snapshot C. Snapshot A, B, dan C adalah snapshot inkremental. Snapshot A digunakan untuk membuat volume EBS 1. Snapshot D dibuat dari volume EBS 1. Snapshot D adalah snapshot inkremental A; bukan snapshot inkremental dari B atau C.



Contoh berikut menunjukkan cara menulis snapshot menggunakan API langsung EBS.

Topik

- Mulai snapshot
- Menempatkan data ke dalam snapshot
- Menyelesaikan snapshot

Mulai snapshot

AWS CLI

Contoh perintah start-snapshot berikut memulai snapshot 8 GiB, menggunakan snapshot `snap-123EXAMPLE1234567` sebagai snapshot induk. Snapshot baru akan berupa snapshot inkremental dari snapshot induk. Snapshot berpindah ke status kesalahan jika tidak ada permintaan put atau complete yang dibuat untuk snapshot dalam periode waktu tunggu `60` menit yang ditentukan. Token klien `550e8400-e29b-41d4-a716-446655440000` memastikan idempotensi permintaan tersebut. Jika token klien dihilangkan, AWS SDK secara otomatis menghasilkan satu untuk Anda. Untuk informasi selengkapnya tentang idempotensi, lihat Idempotensi untuk API StartSnapshot .

```
aws ebs start-snapshot --volume-size 8 --parent-snapshot snap-123EXAMPLE1234567 --
timeout 60 --client-token 550e8400-e29b-41d4-a716-446655440000
```

Contoh respons berikut untuk perintah sebelumnya menunjukkan ID snapshot, ID akun AWS , status, ukuran volume dalam GiB, dan ukuran blok di snapshot. Snapshot dimulai dalam status `pending`. Tentukan ID snapshot di bagian perintah `put-snapshot-block` berikutnya untuk menuliskan data ke snapshot, lalu menggunakan perintah `complete-snapshot` untuk menyelesaikan snapshot dan mengubahnya status menjadi `completed`.

```
{
    "SnapshotId": "snap-0aaEXAMPLEe306d62",
    "OwnerId": "111122223333",
    "Status": "pending",
    "VolumeSize": 8,
    "BlockSize": 524288
}
```

AWS API

Permintaan StartSnapshotcontoh berikut memulai snapshot 8 GiB, menggunakan snapshot `snap-123EXAMPLE1234567` sebagai snapshot induk. Snapshot baru akan berupa snapshot inkremental dari snapshot induk. Snapshot berpindah ke status kesalahan jika tidak ada permintaan put atau complete yang dibuat untuk snapshot dalam periode waktu tunggu `60` menit yang ditentukan. Token klien `550e8400-e29b-41d4-a716-446655440000` memastikan idempotensi permintaan tersebut. Jika token klien dihilangkan, AWS SDK secara otomatis

menghasilkan satu untuk Anda. Untuk informasi selengkapnya tentang idempotensi, lihat Idempotensi untuk API StartSnapshot .

```
POST /snapshots HTTP/1.1
Host: ebs.us-east-2.amazonaws.com
Accept-Encoding: identity
User-Agent: <User agent parameter>
X-Amz-Date: 20200618T040724Z
Authorization: <Authentication parameter>

{
    "VolumeSize": 8,
    "ParentSnapshot": snap-123EXAMPLE1234567,
    "ClientToken": "550e8400-e29b-41d4-a716-446655440000",
    "Timeout": 60
}
```

Contoh respons berikut untuk permintaan sebelumnya menunjukkan ID snapshot, ID akun AWS , status, ukuran volume dalam GiB, dan ukuran blok di snapshot. Snapshot dimulai dalam status tertunda. Tentukan ID snapshot di permintaan `PutSnapshotBlocks` berikutnya untuk menuliskan data ke snapshot.

```
HTTP/1.1 201 Created
x-amzn-RequestId: 929e6eb9-7183-405a-9502-5b7da37c1b18
Content-Type: application/json
Content-Length: 181
Date: Thu, 18 Jun 2020 04:07:29 GMT
Connection: keep-alive

{
    "BlockSize": 524288,
    "Description": null,
    "OwnerId": "138695307491",
    "Progress": null,
    "SnapshotId": "snap-052EXAMPLEc85d8dd",
    "StartTime": null,
    "Status": "pending",
    "Tags": null,
    "VolumeSize": 8
}
```

Menempatkan data ke dalam snapshot

AWS CLI

Contoh perintah berikut put-snapshot menulis 524288 Bita data untuk 1000 indeks blok pada snapshot snap-0aaEXAMPLEe306d62. Checksum QOD3gmEQOXATfJx2Aa34W4FU2nZGyXfqtsUuktOw8DM= berkode Base64 dibuat menggunakan algoritme SHA256. Data yang ditransmisikan terdapat di file /tmp/data.

```
aws ebs put-snapshot-block --snapshot-id snap-0aaEXAMPLEe306d62
 --block-index 1000 --data-length 524288 --block-data /tmp/data --
checksum QOD3gmEQOXATfJx2Aa34W4FU2nZGyXfqtsUuktOw8DM= --checksum-algorithm SHA256
```

Contoh respons untuk perintah sebelumnya berikut ini mengonfirmasi panjang data, checksum, dan algoritma checksum untuk data yang diterima oleh layanan.

```
{
    "DataLength": "524288",
    "Checksum": "QOD3gmEQOXATfJx2Aa34W4FU2nZGyXfqtsUuktOw8DM=",
    "ChecksumAlgorithm": "SHA256"
}
```

AWS API

PutSnapshotContoh permintaan berikut menulis 524288 Bytes data untuk memblokir indeks 1000 pada snapshotsnap-052EXAMPLEc85d8dd. Checksum QOD3gmEQOXATfJx2Aa34W4FU2nZGyXfqtsUuktOw8DM= dengan enkode Base64 dibuat menggunakan algoritma SHA256. Data ditransmisikan dalam badan permintaan dan direpresentasikan seperti *BlockData*pada contoh berikut.

```
PUT /snapshots/snap-052EXAMPLEc85d8dd/blocks/1000 HTTP/1.1
Host: ebs.us-east-2.amazonaws.com
Accept-Encoding: identity
x-amz-Data-Length: 524288
x-amz-Checksum: QOD3gmEQOXATfJx2Aa34W4FU2nZGyXfqtsUuktOw8DM=
x-amz-Checksum-Algorithm: SHA256
User-Agent: <User agent parameter>
X-Amz-Date: 20200618T042215Z
X-Amz-Content-SHA256: UNSIGNED-PAYLOAD
Authorization: <Authentication parameter>
```

```
            BlockData
```

Contoh respons untuk permintaan sebelumnya berikut ini mengonfirmasi panjang data, checksum, dan algoritma checksum untuk data yang diterima oleh layanan.

```
HTTP/1.1 201 Created
x-amzn-RequestId: 643ac797-7e0c-4ad0-8417-97b77b43c57b
x-amz-Checksum: QOD3gmEQOXATfJx2Aa34W4FU2nZGyXfqtsUuktOw8DM=
x-amz-Checksum-Algorithm: SHA256
Content-Type: application/json
Content-Length: 2
Date: Thu, 18 Jun 2020 04:22:12 GMT
Connection: keep-alive

{}
```

Menyelesaikan snapshot

AWS CLI

Contoh perintah complete-snapshot berikut menyelesaikan snapshot `snap-0aaEXAMPLEe306d62`. Perintah menentukan bahwa 5 blok ditulis untuk snapshot. Checksum `6D3nmwi5f2F0wlh7xX8QprrJBFzDX8aacdOcA3KCM3c=` mewakili checksum untuk set data lengkap yang ditulis ke snapshot. Untuk informasi selengkapnya tentang checksum, lihat Gunakan checksum sebelumnya di dalam panduan ini.

```
aws ebs complete-snapshot --snapshot-id snap-0aaEXAMPLEe306d62 --changed-blocks-
count 5 --checksum 6D3nmwi5f2F0wlh7xX8QprrJBFzDX8aacdOcA3KCM3c= --checksum-
algorithm SHA256 --checksum-aggregation-method LINEAR
```

Berikut ini adalah contoh tanggapan untuk perintah sebelumnya.

```
{
    "Status": "pending"
}
```

AWS API

CompleteSnapshotContoh permintaan berikut melengkapi snapshot`snap-052EXAMPLEc85d8dd`. Perintah menentukan bahwa 5 blok ditulis untuk snapshot. Checksum

`6D3nmwi5f2F0wlh7xX8QprrJBFzDX8aacdOcA3KCM3c=` merepresentasikan checksum untuk set data lengkap yang ditulis ke snapshot.

```
POST /snapshots/completion/snap-052EXAMPLEc85d8dd HTTP/1.1
Host: ebs.us-east-2.amazonaws.com
Accept-Encoding: identity
x-amz-ChangedBlocksCount: 5
x-amz-Checksum: 6D3nmwi5f2F0wlh7xX8QprrJBFzDX8aacdOcA3KCM3c=
x-amz-Checksum-Algorithm: SHA256
x-amz-Checksum-Aggregation-Method: LINEAR
User-Agent: <User agent parameter>
X-Amz-Date: 20200618T043158Z
Authorization: <Authentication parameter>
```

Berikut ini adalah contoh tanggapan untuk permintaan sebelumnya.

```
HTTP/1.1 202 Accepted
x-amzn-RequestId: 06cba5b5-b731-49de-af40-80333ac3a117
Content-Type: application/json
Content-Length: 20
Date: Thu, 18 Jun 2020 04:31:50 GMT
Connection: keep-alive

{"Status":"pending"}
```

Gunakan enkripsi

Saat Anda memulai snapshot baru menggunakan StartSnapshot, status enkripsi bergantung pada nilai yang Anda tentukan untuk Encrypted, KmsKeyArn, dan ParentSnapshotId, dan apakah AWS akun Anda diaktifkan untuk enkripsi secara default.

> **Note**
>
> - Anda mungkin memerlukan izin IAM tambahan untuk menggunakan API langsung EBS dengan enkripsi. Untuk informasi selengkapnya, lihat Izin untuk digunakan AWS KMS keys.
> - Jika enkripsi Amazon EBS secara default diaktifkan di AWS akun, Anda tidak dapat membuat snapshot yang tidak terenkripsi.
> - Jika enkripsi Amazon EBS secara default diaktifkan di AWS akun Anda, Anda tidak dapat memulai snapshot baru menggunakan snapshot induk yang tidak terenkripsi. Anda harus

terlebih dahulu mengenkripsi snapshot induk dengan menyalinnya. Untuk informasi selengkapnya, lihat Menyalin snapshot Amazon EBS.

Topik

- Hasil enkripsi: Snapshot induk yang tidak terenkripsi
- Hasil enkripsi: Snapshot induk yang tidak terenkripsi
- Hasil enkripsi: Tidak ada snapshot induk

Hasil enkripsi: Snapshot induk yang tidak terenkripsi

Tabel berikut menjelaskan hasil enkripsi untuk setiap kombinasi pengaturan yang memungkinkan.

ParentSnapshotId	Dienkripsi	KmsKeyArn	Enkripsi secara default	Hasil
Tidak terenkripsi	Diabaikan	Diabaikan	Aktif	Permintaan gagal dengan `ValidationException` .
			Nonaktif	Snapshot tidak terenkripsi.
		Ditentukan	Aktif	
			Nonaktif	
Tidak terenkripsi	Benar	Diabaikan	Aktif	Permintaan gagal dengan `ValidationException` .
			Nonaktif	
		Ditentukan	Aktif	
			Nonaktif	
Tidak terenkripsi	Salah	Diabaikan	Aktif	Permintaan gagal dengan `ValidationException` .
			Nonaktif	
		Ditentukan	Aktif	

ParentSnapshotId	Dienkripsi	KmsKeyArn	Enkripsi secara default	Hasil
			Nonaktif	

Hasil enkripsi: Snapshot induk yang tidak terenkripsi

Tabel berikut menjelaskan hasil enkripsi untuk setiap kombinasi pengaturan yang memungkinkan.

ParentSnapshotId	Dienkripsi	KmsKeyArn	Enkripsi secara default	Hasil
Dienkripsi	Diabaikan	Diabaikan	Aktif	Snapshot dienkripsi menggunakan kunci KMS yang sama dengan snapshot induk.
			Nonaktif	
		Ditentukan	Aktif	Permintaan gagal dengan `ValidationException` .
			Nonaktif	
Dienkripsi	Benar	Diabaikan	Aktif	Permintaan gagal dengan `ValidationException` .
			Nonaktif	
		Ditentukan	Aktif	
			Nonaktif	
Dienkripsi	Salah	Diabaikan	Aktif	Permintaan gagal dengan `ValidationException` .
			Nonaktif	
		Ditentukan	Aktif	
			Nonaktif	

Hasil enkripsi: Tidak ada snapshot induk

Tabel berikut menjelaskan hasil enkripsi untuk setiap kombinasi pengaturan yang memungkinkan.

ParentSnapshotId	Dienkripsi	KmsKeyArn	Enkripsi secara default	Hasil
Diabaikan	Benar	Diabaikan	Aktif	Snapshot dienkripsi menggunakan kunci KMS default untuk akun Anda. *
			Nonaktif	
		Ditentukan	Diaktifkan	Snapshot dienkripsi menggunakan kunci KMS yang ditentukan untuk Arn. KmsKey
			Nonaktif	
Diabaikan	Salah	Diabaikan	Aktif	Permintaan gagal dengan `ValidationException` .
			Nonaktif	Snapshot tidak terenkripsi.
		Ditentukan	Aktif	Permintaan gagal dengan `ValidationException` .
			Nonaktif	
Diabaikan	Diabaikan	Diabaikan	Aktif	Snapshot dienkripsi menggunakan kunci KMS default untuk akun Anda. *
			Nonaktif	Snapshot tidak terenkripsi.
		Ditentukan	Diaktifkan	Snapshot dienkripsi menggunakan kunci KMS yang ditentukan untuk Arn. KmsKey
			Nonaktif	

* Kunci KMS default ini bisa berupa kunci yang dikelola pelanggan atau kunci KMS AWS terkelola default untuk enkripsi Amazon EBS.

Gunakan tanda tangan Signature Versi 4

Signature Version 4 adalah proses untuk menambahkan informasi otentikasi ke AWS permintaan yang dikirim oleh HTTP. Untuk keamanan, sebagian besar permintaan AWS harus ditandatangani dengan kunci akses, yang terdiri dari ID kunci akses dan kunci akses rahasia. Kedua kunci ini umumnya disebut sebagai kredensial keamanan Anda. Untuk informasi tentang cara mendapatkan kredensial untuk akun Anda, lihat kredensial keamanan AWS.

Jika Anda ingin membuat permintaan HTTP secara manual, Anda harus mempelajari cara menandatanganinya. Saat Anda menggunakan AWS Command Line Interface (AWS CLI) atau salah satu AWS SDK untuk membuat permintaan AWS, alat ini secara otomatis menandatangani permintaan untuk Anda dengan kunci akses yang Anda tentukan saat Anda mengonfigurasi alat. Saat menggunakan alat ini, Anda tidak perlu mempelajari cara menandatangani permintaan diri.

Untuk informasi selengkapnya, lihat Menandatangani permintaan AWS API di Panduan Pengguna IAM.

Gunakan checksum

GetSnapshotBlock Tindakan mengembalikan data yang ada di blok snapshot, dan PutSnapshotBlock tindakan menambahkan data ke blok dalam snapshot. Data blok yang tidak ditransmisikan sebagai bagian dari proses penandatanganan Signature Versi 4. Oleh karena itu, checksum digunakan untuk memvalidasi integritas data sebagai berikut:

- Saat Anda menggunakan GetSnapshotBlock tindakan, respons menyediakan checksum SHA256 yang dikodekan Base64 untuk data blok menggunakan header X-AMZ-checksum, dan algoritma checksum menggunakan header X-AMZ-Checksum-algorithm. Gunakan checksum yang dikembalikan untuk memvalidasi integritas data. Jika checksum yang Anda hasilkan tidak sesuai dengan yang diberikan oleh Amazon EBS, Anda harus mempertimbangkan data yang tidak valid dan mencoba kembali permintaan Anda.
- Saat Anda menggunakan PutSnapshotBlock tindakan, permintaan Anda harus menyediakan checksum SHA256 yang dikodekan Base64 untuk data blok menggunakan header X-AMZ-checksum, dan algoritma checksum menggunakan header X-AMZ-Checksum-Algorithm. Checksum yang Anda berikan divalidasi dengan checksum yang dibuat oleh Amazon EBS untuk memvalidasi integritas data. Jika checksum tidak sesuai, permintaan gagal.
- Saat Anda menggunakan CompleteSnapshot tindakan, permintaan Anda secara opsional dapat memberikan checksum SHA256 yang dikodekan Base64 agregat untuk kumpulan data lengkap yang ditambahkan ke snapshot. Berikan checksum menggunakan header x-amz-Checksum,

algoritma checksum menggunakan header x-amz-Checksum-Algorithm, dan metode agregasi checksum menggunakan header x-amz-Checksum-Aggregation-Method. Untuk membuat checksum gabungan menggunakan metode agregasi linear, mengatur checksum untuk setiap blok dalam urutan naik indeks blok mereka, menyatukan mereka untuk membentuk satu string, kemudian membuat checksum pada seluruh rangkaian menggunakan SHA256.//

Checksum dalam tindakan ini merupakan bagian dari proses penandatanganan Signature Versi 4.

Idempotensi untuk API StartSnapshot

Idempotensi memastikan bahwa permintaan API hanya selesai satu kali. Dengan permintaan idempotensi, jika permintaan asli selesai, percobaan berikutnya mengembalikan hasil dari permintaan awal yang berhasil dan tidak memiliki efek tambahan.

StartSnapshotAPI mendukung idempotensi menggunakan token klien. Token klien adalah string unik yang Anda tentukan saat membuat permintaan API. Jika Anda mencoba ulang permintaan API dengan token klien yang sama dan parameter permintaan yang sama setelah berhasil diselesaikan, hasil permintaan awal akan dikembalikan. Jika Anda mencoba ulang permintaan dengan token klien yang sama, tetapi mengubah satu atau beberapa parameter permintaan, kesalahan `ConflictException` dikembalikan.

Jika Anda tidak menentukan token klien Anda sendiri, AWS SDK secara otomatis menghasilkan token klien untuk permintaan tersebut guna memastikan bahwa token tersebut idempoten.

Token klien dapat berupa string yang mencakup hingga 64 karakter ASCII. Anda tidak boleh menggunakan kembali token klien yang sama untuk permintaan yang berbeda.

Untuk membuat StartSnapshot permintaan idempoten dengan token klien Anda sendiri menggunakan API

Tentukan parameter permintaan `ClientToken`.

```
POST /snapshots HTTP/1.1
Host: ebs.us-east-2.amazonaws.com
Accept-Encoding: identity
User-Agent: <User agent parameter>
X-Amz-Date: 20200618T040724Z
Authorization: <Authentication parameter>

{
```

```
    "VolumeSize": 8,
    "ParentSnapshot": snap-123EXAMPLE1234567,
    "ClientToken": "550e8400-e29b-41d4-a716-446655440000",
    "Timeout": 60
}
```

Untuk membuat StartSnapshot permintaan idempoten dengan token klien Anda sendiri menggunakan AWS CLI

Tentukan parameter permintaan `client-token`.

```
$ C:\> aws ebs start-snapshot --region us-east-2 --volume-size 8 --parent-
snapshot snap-123EXAMPLE1234567 --timeout 60 --client-token 550e8400-e29b-41d4-
a716-446655440000
```

Kesalahan mencoba lagi

AWS SDK menerapkan logika coba ulang otomatis untuk permintaan yang mengembalikan respons kesalahan. Anda dapat mengonfigurasi pengaturan coba lagi untuk AWS SDK. Untuk informasi selengkapnya, lihat dokumentasi SDK Anda.

Anda dapat mengonfigurasi AWS CLI agar secara otomatis mencoba kembali beberapa permintaan yang gagal. Untuk informasi selengkapnya tentang mengonfigurasi percobaan ulang AWS CLI, lihat AWS CLI mencoba ulang di Panduan Pengguna.AWS Command Line Interface

API Kueri AWS tidak mendukung logika coba ulang untuk permintaan yang gagal. Jika Anda menggunakan permintaan HTTP atau HTTPS, Anda harus menerapkan logika coba lagi dalam aplikasi klien Anda.

Tabel berikut menunjukkan kemungkinan respons kesalahan API. Beberapa kesalahan API dapat dicoba ulang. Aplikasi klien Anda harus selalu mencoba ulang permintaan gagal yang menerima kesalahan yang dapat dicoba ulang.

Kesalahan	Kode respons	Deskripsi	Dilempar oleh	Dicoba ulang?
`InternalServerException`	500	Permintaan gagal karena masalah jaringan atau AWS sisi server.	Semua API	Ya

Kesalahan	Kode respons	Deskripsi	Dilempar oleh	Dicoba ulang?
`ThrottlingException`	400	Jumlah permintaan API telah melampaui batas throttling permintaan API maksimum yang diizinkan untuk akun.	Semua API	Ya
`RequestThrottleException`	400	Jumlah permintaan API telah melampaui batas throttling permintaan API maksimum yang diizinkan untuk snapshot.	GetSnapshotBlock | PutSnapshotBlock	Ya
`ValidationException` dengan pesan `"Failed to read block data"`	400	Blok data yang disediakan tidak dapat dibaca.	PutSnapshotBlok	Ya
`ValidationException` dengan pesan lainnya	400	Sintaksis permintaan salah format, atau input tidak memenuhi batasan yang ditentukan oleh Layanan AWS.	Semua API	Tidak

Kesalahan	Kode respons	Deskripsi	Dilempar oleh	Dicoba ulang?
`ResourceNotFoundException`	404	ID snapshot yang ditentukan tidak ada.	Semua API	Tidak
`ConflictException`	409	Token klien yang ditentukan sebelumnya digunakan dalam permintaan serupa yang memiliki parameter permintaan berbeda. Untuk informasi selengkapnya, lihat Idempotensi untuk API StartSnapshot .	StartSnapshot	Tidak
`AccessDeniedException`	403	Anda tidak memiliki izin untuk melakukan operasi yang diminta.	Semua API	Tidak
`ServiceQuotaExceededException`	402	Permintaan gagal karena memenuhi permintaan akan melebihi satu atau lebih kuota layanan dependen untuk akun Anda.	Semua API	Tidak

Kesalahan	Kode respons	Deskripsi	Dilempar oleh	Dicoba ulang?
`InvalidSignatureException`	403	Tanda tangan otorisasi permintaan telah kedaluwarsa. Anda dapat mencoba lagi permintaan hanya setelah menyegarkan tanda tangan otorisasi.	Semua API	Tidak

Optimalkan performa

Anda dapat menjalankan permintaan API secara bersamaan. Dengan asumsi PutSnapshotBlock latensi adalah 100 ms, maka utas dapat memproses 10 permintaan dalam satu detik. Selain itu, dengan asumsi aplikasi klien Anda menciptakan beberapa alur dan koneksi (misalnya, 100 koneksi), dapat membuat 1000 permintaan (10 * 100) per detik secara keseluruhan. Ini akan sesuai dengan throughput sekitar 500 MB per detik.

Daftar berikut ini berisi beberapa hal yang harus dicari dalam aplikasi Anda:

- Apakah setiap utas menggunakan koneksi terpisah? Jika koneksi dibatasi pada aplikasi, maka banyak alur akan menunggu koneksi tersedia dan Anda akan melihat throughput yang lebih rendah.
- Apakah ada waktu tunggu dalam aplikasi di antara dua permintaan yang dimasukkan? Hal ini akan mengurangi throughput alur yang efektif.
- Batas bandwidth pada instance — Jika bandwidth pada instance dibagi oleh aplikasi lain, itu bisa membatasi throughput yang tersedia untuk PutSnapshotBlock permintaan.

Pastikan untuk mencatat beban kerja lain yang mungkin berjalan di akun untuk menghindari hambatan. Anda juga harus membangun mekanisme coba ulang ke dalam alur kerja API langsung EBS untuk menangani throttling, waktu habis, dan ketidaktersediaan layanan.

Tinjau kuota layanan API langsung EBS untuk menentukan permintaan API maksimum yang dapat Anda jalankan per detik. Untuk informasi selengkapnya, lihat Titik Akhir dan Kuota Amazon Elastic Block Store dalam Referensi Umum AWS .

Titik akhir layanan API langsung EBS

Endpoint adalah URL yang berfungsi sebagai titik masuk untuk layanan AWS web. API langsung EBS mendukung jenis titik akhir berikut:

- Titik akhir IPv4
- Titik akhir tumpukan ganda yang mendukung IPv4 dan IPv6
- Titik akhir FIPS

Saat Anda membuat permintaan, Anda dapat menentukan titik akhir dan Wilayah yang akan digunakan. Jika Anda tidak menentukan titik akhir, titik akhir IPv4 digunakan secara default. Untuk menggunakan tipe titik akhir yang berbeda, Anda harus menentukannya dalam permintaan Anda. Untuk contoh cara melakukannya, lihat Menentukan titik akhir.

Untuk informasi selengkapnya tentang Wilayah, lihat Wilayah dan Zona Ketersediaan di Panduan Pengguna Amazon EC2. Untuk daftar titik akhir untuk API langsung EBS, lihat Titik akhir untuk API langsung EBS di Referensi Umum Amazon Web Services.

Topik

Titik akhir IPv4

Titik akhir IPv4 hanya mendukung lalu lintas IPv4. Titik akhir IPv4 tersedia untuk semua Wilayah.

API langsung EBS hanya mendukung titik akhir IPv4 Regional yang dapat Anda gunakan untuk membuat permintaan. Anda harus menentukan Region sebagai bagian dari nama endpoint. Nama endpoint menggunakan konvensi penamaan berikut:

- `ebs.region.amazonaws.com`

Misalnya, untuk mengarahkan permintaan Anda ke titik akhir `us-east-2` IPv4, Anda harus menentukan `ebs.us-east-2.amazonaws.com` sebagai titik akhir. Untuk daftar titik akhir untuk API langsung EBS, lihat Titik akhir untuk API langsung EBS di Referensi Umum Amazon Web Services.

Harga

Anda tidak dikenai biaya untuk data yang ditransfer langsung antara API langsung EBS dan instans Amazon EC2 menggunakan titik akhir IPv4 di Wilayah yang sama. Namun, jika ada layanan perantara, seperti AWS PrivateLink titik akhir, NAT Gateway, atau Gateway Transit VPC Amazon, Anda akan dikenakan biaya terkait.

Titik akhir tumpukan ganda (IPv4 dan IPv6)

Titik akhir tumpukan ganda mendukung lalu lintas IPv4 dan IPv6. Titik akhir tumpukan ganda tersedia untuk semua Wilayah.

Untuk menggunakan IPv6, Anda harus menggunakan titik akhir tumpukan ganda. Saat Anda membuat permintaan ke titik akhir tumpukan ganda, URL titik akhir memutuskan ke alamat IPv6 atau IPv4, tergantung pada protokol yang digunakan oleh jaringan dan klien Anda.

API langsung EBS hanya mendukung titik akhir tumpukan ganda regional, yang berarti Anda harus menentukan Wilayah sebagai bagian dari nama titik akhir. Nama titik akhir tumpukan ganda menggunakan konvensi penamaan berikut:

- `ebs.`*`region`*`.api.aws`

Misalnya, nama titik akhir tumpukan ganda untuk Wilayah `eu-west-1` adalah `ebs.eu-west-1.api.aws`. Untuk daftar titik akhir untuk API langsung EBS, lihat Titik akhir untuk API langsung EBS di Referensi Umum Amazon Web Services.

Harga

Anda tidak dikenai biaya untuk data yang ditransfer langsung antara API langsung EBS dan instans Amazon EC2 menggunakan titik akhir tumpukan ganda di Wilayah yang sama. Namun, jika ada layanan perantara, seperti AWS PrivateLink titik akhir, NAT Gateway, atau Gateway Transit VPC Amazon, Anda akan dikenakan biaya terkait.

Titik akhir FIPS

API langsung EBS menyediakan IPv4 dan tumpukan ganda (IPv4 dan IPv6) yang divalidasi FIPS untuk Wilayah berikut:

- us-east-1 — AS Timur (Virginia Utara)
- us-east-2 — AS Timur (Ohio)
- us-west-1 — AS Barat (California Utara)
- us-west-2 — AS Barat (Oregon)
- ca-central-1 – Kanada (Pusat)

Titik akhir IPv4 FIPS menggunakan konvensi penamaan berikut: ebs-fips.*region*.amazonaws.com. Misalnya, titik akhir IPv4 FIPS untuk us-east-1 adalah ebs-fips.us-east-1.amazonaws.com.

Titik akhir tumpukan ganda FOPS menggunakan konvensi penamaan berikut: ebs-fips.*region*.api.aws. Misalnya, titik akhir tumpukan ganda FIPS untuk us-east-1 adalah ebs-fips.us-east-1.api.aws.

Untuk informasi selengkapnya tentang titik akhir FIPS, lihat Titik akhir FIPS di Referensi Umum Amazon Web Services.

Menentukan titik akhir

Bagian ini memberikan beberapa contoh cara menentukan titik akhir saat membuat permintaan.

AWS CLI

Contoh berikut menunjukkan cara menentukan titik akhir untuk Wilayah us-east-2 menggunakan AWS CLI.

- Tumpukan ganda

```
aws ebs list-snapshot-blocks --snapshot-id snap-0987654321 --starting-block-index
 1000 --endpoint-url https://ebs.us-east-2.api.aws
```

- IPv4

```
aws ebs list-snapshot-blocks --snapshot-id snap-0987654321 --starting-block-index
 1000 --endpoint-url https://ebs.us-east-2.amazonaws.com
```

AWS SDK for Java 2.x

Contoh berikut menunjukkan cara menentukan titik akhir untuk Wilayah `us-east-2` menggunakan AWS SDK for Java 2.x.

- Tumpukan ganda

```
AwsClientBuilder.EndpointConfiguration config = new
 AwsClientBuilder.EndpointConfiguration("https://ebs.us-east-2.api.aws", "us-
east-2");
AmazonEBS ebs = AmazonEBSClientBuilder.standard()
    .withEndpointConfiguration(config)
    .build();
```

- IPv4

```
AwsClientBuilder.EndpointConfiguration config = new
 AwsClientBuilder.EndpointConfiguration("https://ebs.us-east-2.amazonaws.com",
 "us-east-2");
AmazonEBS ebs = AmazonEBSClientBuilder.standard()
    .withEndpointConfiguration(config)
    .build();
```

AWS SDK for Go

Contoh berikut menunjukkan cara menentukan titik akhir untuk Wilayah `us-east-2` menggunakan AWS SDK for Go.

- Tumpukan ganda

```
sess := session.Must(session.NewSession())
svc := ebs.New(sess, &aws.Config{
    Region: aws.String(endpoints.UsEast1RegionID),
    Endpoint: aws.String("https://ebs.us-east-2.api.aws")
})
```

- IPv4

```
sess := session.Must(session.NewSession())
svc := ebs.New(sess, &aws.Config{
    Region: aws.String(endpoints.UsEast1RegionID),
```

```
    Endpoint: aws.String("https://ebs.us-east-2.amazonaws.com")
})
```

Harga API langsung EBS

Topik

- Harga API
- Biaya jaringan

Harga API

Harga yang Anda bayar untuk menggunakan API langsung EBS tergantung pada permintaan yang Anda buat. Untuk informasi selengkapnya, lihat harga Amazon EBS.

- ListChangedBlok dan ListSnapshotBlocks API dikenakan biaya per permintaan. Misalnya, jika Anda membuat 100.000 permintaan ListSnapshotBlocks API di Wilayah yang mengenakan biaya $0,0006 per 1.000 permintaan, Anda akan dikenakan biaya $0,06 ($0,0006 per 1.000 permintaan x 100).
- GetSnapshotBlok dibebankan per blok yang dikembalikan. Misalnya, jika Anda membuat 100.000 permintaan GetSnapshotBlock API di Wilayah yang mengenakan biaya $0,003 per 1.000 blok yang dikembalikan, Anda akan dikenakan biaya $0,30 ($0,003 per 1.000 blok dikembalikan x 100).
- PutSnapshotBlok dibebankan per blok tertulis. Misalnya, jika Anda membuat 100.000 permintaan PutSnapshotBlock API di Wilayah yang mengenakan biaya $0,006 per 1.000 blok yang ditulis, Anda akan dikenakan biaya $0,60 ($0,006 per 1.000 blok yang ditulis x 100).

Biaya jaringan

Biaya transfer data

Data yang ditransfer langsung antara API langsung EBS dan instans Amazon EC2 di Wilayah AWS yang sama gratis saat menggunakan titik akhir non-FIPS. Untuk informasi selengkapnya, lihat AWS titik akhir layanan. Jika AWS layanan lain berada di jalur transfer data Anda, Anda akan dikenakan biaya pemrosesan data terkait. Layanan ini termasuk, namun tidak terbatas pada, PrivateLink titik akhir, NAT Gateway, dan Transit Gateway.

Titik akhir antarmuka VPC

Jika Anda menggunakan API langsung EBS dari instans AWS Lambda atau fungsi Amazon EC2 di subnet pribadi, Anda dapat menggunakan titik akhir antarmuka VPC, alih-alih menggunakan gateway NAT, untuk mengurangi biaya transfer data jaringan. Untuk informasi selengkapnya, lihat Menggunakan titik akhir VPC dengan API langsung EBS.

Menggunakan titik akhir VPC dengan API langsung EBS

Anda dapat membangun koneksi privat antara VPC dan API langsung EBS dengan membuat titik akhir VPC antarmuka, yang didukung oleh AWS PrivateLink. Anda dapat mengakses API langsung EBS seolah-olah berada di VPC Anda, tanpa menggunakan gateway internet, perangkat NAT, koneksi VPN, atau koneksi AWS Direct Connect . Instans dalam VPC Anda tidak memerlukan alamat IP publik untuk berkomunikasi dengan API langsung EBS.

Kami membuat antarmuka jaringan endpoint di setiap subnet yang Anda aktifkan untuk titik akhir antarmuka.

Untuk informasi selengkapnya, lihat Akses Layanan AWS melalui AWS PrivateLink di AWS PrivateLink Panduan.

Pertimbangan untuk titik akhir VPC API langsung EBS

Sebelum Anda menyiapkan titik akhir VPC antarmuka untuk API langsung EBS, tinjau Pertimbangan di Panduan AWS PrivateLink .

Secara default, akses penuh ke API langsung EBS diizinkan melalui titik akhir. Anda dapat mengontrol akses ke titik akhir antarmuka menggunakan kebijakan titik akhir VPC. Anda dapat melampirkan kebijakan titik akhir ke titik akhir VPC yang mengontrol akses ke API langsung EBS. Kebijakan titik akhir menentukan informasi berikut:

- Kepala sekolah yang dapat melakukan tindakan.
- Tindakan yang bisa dilakukan.
- Sumber daya di mana tindakan dapat dilakukan.

Untuk informasi selengkapnya, lihat Mengendalikan Akses ke Layanan dengan Titik Akhir VPC dalam Panduan Pengguna VPC Amazon.

Berikut ini adalah contoh kebijakan endpoint untuk API langsung EBS. Saat dilampirkan ke titik akhir, kebijakan ini memberikan akses ke semua tindakan API langsung EBS di semua sumber daya, kecuali snapshot yang ditandai dengan kunci dan nilai. `Environment Test`

```
{
    "Statement": [
        {
            "Effect": "Deny",
            "Action": "ebs:*",
            "Principal": "*",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Environment": "Test"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ebs:*",
            "Principal": "*",
            "Resource": "*"
        }
    ]
}
```

Buat suatu titik akhir VPC antarmuka untuk API langsung EBS

Anda dapat membuat titik akhir VPC untuk API langsung EBS menggunakan konsol Amazon VPC atau AWS Command Line Interface (AWS CLI). Untuk informasi selengkapnya, lihat Membuat titik akhir VPC di Panduan Pengguna AWS PrivateLink .

Buat titik akhir VPC untuk API langsung EBS menggunakan nama layanan berikut:

- `com.amazonaws.region.ebs`

Jika Anda mengaktifkan DNS privat untuk titik akhir, Anda dapat mengajukan permintaan API ke API langsung EBS menggunakan nama DNS default untuk Wilayah, misalnya, `ebs.us-east-1.amazonaws.com`.

Log API Panggilan untuk API langsung EBS dengan AWS CloudTrail

Layanan API langsung EBS terintegrasi dengan AWS CloudTrail. CloudTrail adalah layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan. CloudTrail menangkap semua panggilan API yang dilakukan di API langsung EBS sebagai peristiwa. Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail acara secara terus menerus ke bucket Amazon Simple Storage Service (Amazon S3). Jika Anda tidak mengonfigurasi jejak, Anda masih dapat melihat peristiwa manajemen terbaru di CloudTrail konsol dalam Riwayat acara. Peristiwa data tidak ditangkap dalam riwayat Peristiwa. Anda dapat menggunakan informasi yang dikumpulkan oleh CloudTrail untuk menentukan permintaan yang dibuat untuk API langsung EBS, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan.

Untuk informasi selengkapnya CloudTrail, lihat Panduan AWS CloudTrail Pengguna.

Informasi API langsung EBS di CloudTrail

CloudTrail diaktifkan di AWS akun Anda saat Anda membuat akun. Ketika aktivitas peristiwa yang didukung terjadi di API langsung EBS, aktivitas tersebut direkam dalam suatu CloudTrail peristiwa bersama dengan peristiwa AWS layanan lainnya dalam riwayat Acara. Anda dapat melihat, mencari, dan mengunduh acara terbaru di AWS akun Anda. Untuk informasi selengkapnya, lihat Melihat Acara dengan Riwayat CloudTrail Acara.

Untuk catatan peristiwa yang sedang berlangsung di AWS akun Anda, termasuk peristiwa untuk API langsung EBS, buat jejak. Jejak memungkinkan CloudTrail untuk mengirimkan file log ke bucket S3. Secara default, saat Anda membuat jejak di konsol, jejak tersebut berlaku untuk semua AWS Wilayah. Jejak mencatat peristiwa dari semua Wilayah di AWS partisi dan mengirimkan file log ke bucket 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

Tindakan API yang didukung

Untuk API langsung EBS, Anda dapat menggunakan CloudTrail untuk mencatat dua jenis peristiwa:

- Acara manajemen - Acara manajemen memberikan visibilitas ke dalam operasi manajemen yang dilakukan pada snapshot di akun Anda AWS . Tindakan API berikut dicatat secara default sebagai peristiwa manajemen dalam jejak:
  - StartSnapshot
  - CompleteSnapshot

  Untuk informasi selengkapnya tentang peristiwa pengelolaan logging, lihat peristiwa manajemen logging untuk jejak di Panduan CloudTrail Pengguna.
- Peristiwa data — Peristiwa ini memberikan visibilitas tentang operasi snapshot yang dilakukan pada atau dalam snapshot. Tindakan API berikut secara opsional dapat dicatat sebagai peristiwa data di jejak:
  - ListSnapshotBlok
  - ListChangedBlok
  - GetSnapshotBlock
  - PutSnapshotBlok

  Peristiwa data tidak dicatat secara default saat Anda membuat jejak. Anda hanya dapat menggunakan penyeleksi peristiwa lanjutan untuk merekam peristiwa data pada panggilan API langsung EBS. Untuk informasi selengkapnya, lihat Mencatat peristiwa data untuk jejak di Panduan CloudTrail Pengguna.

> **Note**
>
> Jika Anda melakukan tindakan pada snapshot yang dibagikan dengan Anda, peristiwa data tidak dikirim ke AWS akun yang memiliki snapshot tersebut.

Informasi identitas

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

- Baik permintaan tersebut dibuat dengan kredensial pengguna root atau pengguna.

- Apakah permintaan tersebut dibuat dengan kredensial keamanan sementara untuk satu peran atau pengguna terfederasi.
- Apakah permintaan itu dibuat oleh AWS layanan lain.

Untuk informasi selengkapnya, lihat CloudTrail pengguna IdentityElement.

Pahami Entri File Log API langsung EBS

Trail adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai file log ke bucket 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.

Berikut ini adalah contoh entri CloudTrail log.

StartSnapshot


```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "123456789012",
        "arn": "arn:aws:iam::123456789012:root",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "user"
    },
    "eventTime": "2020-07-03T23:27:26Z",
    "eventSource": "ebs.amazonaws.com",
    "eventName": "StartSnapshot",
    "awsRegion": "eu-west-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "PostmanRuntime/7.25.0",
    "requestParameters": {
        "volumeSize": 8,
        "clientToken": "token",
        "encrypted": true
    },
    "responseElements": {
        "snapshotId": "snap-123456789012",
```

```
        "ownerId": "123456789012",
        "status": "pending",
        "startTime": "Jul 3, 2020 11:27:26 PM",
        "volumeSize": 8,
        "blockSize": 524288,
        "kmsKeyArn": "HIDDEN_DUE_TO_SECURITY_REASONS"
    },
    "requestID": "be112233-1ba5-4ae0-8e2b-1c302EXAMPLE",
    "eventID": "6e12345-2a4e-417c-aa78-7594fEXAMPLE",
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789012"
}
```

CompleteSnapshot

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "123456789012",
        "arn": "arn:aws:iam::123456789012:root",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "user"
    },
    "eventTime": "2020-07-03T23:28:24Z",
    "eventSource": "ebs.amazonaws.com",
    "eventName": "CompleteSnapshot",
    "awsRegion": "eu-west-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "PostmanRuntime/7.25.0",
    "requestParameters": {
        "snapshotId": "snap-123456789012",
        "changedBlocksCount": 5
    },
    "responseElements": {
        "status": "completed"
    },
    "requestID": "be112233-1ba5-4ae0-8e2b-1c302EXAMPLE",
    "eventID": "6e12345-2a4e-417c-aa78-7594fEXAMPLE",
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789012"
}
```

ListSnapshotBlocks


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDAT4HPB2AO3JEXAMPLE",
        "arn": "arn:aws:iam::123456789012:user/user",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "user"
    },
    "eventTime": "2021-06-03T00:32:46Z",
    "eventSource": "ebs.amazonaws.com",
    "eventName": "ListSnapshotBlocks",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "111.111.111.111",
    "userAgent": "PostmanRuntime/7.28.0",
    "requestParameters": {
        "snapshotId": "snap-abcdef01234567890",
        "maxResults": 100,
        "startingBlockIndex": 0
    },
    "responseElements": null,
    "requestID": "example6-0e12-4aa9-b923-1555eexample",
    "eventID": "example4-218b-4f69-a9e0-2357dexample",
    "readOnly": true,
    "resources": [
        {
            "accountId": "123456789012",
            "type": "AWS::EC2::Snapshot",
            "ARN": "arn:aws:ec2:us-west-2::snapshot/snap-abcdef01234567890"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": false,
    "recipientAccountId": "123456789012",
    "eventCategory": "Data",
    "tlsDetails": {
        "tlsVersion": "TLSv1.2",
        "cipherSuite": "ECDHE-RSA-AES128-SHA",
        "clientProvidedHostHeader": "ebs.us-west-2.amazonaws.com"
    }
```

```
}
```

ListChangedBlocks

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDAT4HPB2AO3JEXAMPLE",
        "arn": "arn:aws:iam::123456789012:user/user",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "user"
    },
    "eventTime": "2021-06-02T21:11:46Z",
    "eventSource": "ebs.amazonaws.com",
    "eventName": "ListChangedBlocks",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "111.111.111.111",
    "userAgent": "PostmanRuntime/7.28.0",
    "requestParameters": {
        "firstSnapshotId": "snap-abcdef01234567890",
        "secondSnapshotId": "snap-9876543210abcdef0",
        "maxResults": 100,
        "startingBlockIndex": 0
    },
    "responseElements": null,
    "requestID": "example0-f4cb-4d64-8d84-72e1bexample",
    "eventID": "example3-fac4-4a78-8ebb-3e9d3example",
    "readOnly": true,
    "resources": [
        {
            "accountId": "123456789012",
            "type": "AWS::EC2::Snapshot",
            "ARN": "arn:aws:ec2:us-west-2::snapshot/snap-abcdef01234567890"
        },
       {
            "accountId": "123456789012",
            "type": "AWS::EC2::Snapshot",
            "ARN": "arn:aws:ec2:us-west-2::snapshot/snap-9876543210abcdef0"
        }
    ],
    "eventType": "AwsApiCall",
```

```
    "managementEvent": false,
    "recipientAccountId": "123456789012",
    "eventCategory": "Data",
    "tlsDetails": {
        "tlsVersion": "TLSv1.2",
        "cipherSuite": "ECDHE-RSA-AES128-SHA",
        "clientProvidedHostHeader": "ebs.us-west-2.amazonaws.com"
    }
}
```

GetSnapshotBlock

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDAT4HPB2AO3JEXAMPLE",
        "arn": "arn:aws:iam::123456789012:user/user",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "user"
    },
    "eventTime": "2021-06-02T20:43:05Z",
    "eventSource": "ebs.amazonaws.com",
    "eventName": "GetSnapshotBlock",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "111.111.111.111",
    "userAgent": "PostmanRuntime/7.28.0",
    "requestParameters": {
        "snapshotId": "snap-abcdef01234567890",
        "blockIndex": 1,
        "blockToken": "EXAMPLEiL5E3pMPFpaDWjExM2/mnSKh1mQfcbjwe2mM7EwhrgCdPAEXAMPLE"
    },
    "responseElements": null,
    "requestID": "examplea-6eca-4964-abfd-fd9f0example",
    "eventID": "example6-4048-4365-a275-42e94example",
    "readOnly": true,
    "resources": [
        {
          "accountId": "123456789012",
          "type": "AWS::EC2::Snapshot",
          "ARN": "arn:aws:ec2:us-west-2::snapshot/snap-abcdef01234567890"
        }
```

```
    ],
    "eventType": "AwsApiCall",
    "managementEvent": false,
    "recipientAccountId": "123456789012",
    "eventCategory": "Data",
    "tlsDetails": {
        "tlsVersion": "TLSv1.2",
        "cipherSuite": "ECDHE-RSA-AES128-SHA",
        "clientProvidedHostHeader": "ebs.us-west-2.amazonaws.com"
    }
}
```

PutSnapshotBlock

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDAT4HPB2AO3JEXAMPLE",
        "arn": "arn:aws:iam::123456789012:user/user",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "user"
    },
    "eventTime": "2021-06-02T21:09:17Z",
    "eventSource": "ebs.amazonaws.com",
    "eventName": "PutSnapshotBlock",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "111.111.111.111",
    "userAgent": "PostmanRuntime/7.28.0",
    "requestParameters": {
        "snapshotId": "snap-abcdef01234567890",
        "blockIndex": 1,
        "dataLength": 524288,
        "checksum": "exampleodSGvFSb1e3kxWUgbOQ4TbzPurnsfVexample",
        "checksumAlgorithm": "SHA256"
    },
    "responseElements": {
        "checksum": "exampleodSGvFSb1e3kxWUgbOQ4TbzPurnsfVexample",
        "checksumAlgorithm": "SHA256"
    },
    "requestID": "example3-d5e0-4167-8ee8-50845example",
    "eventID": "example8-4d9a-4aad-b71d-bb31fexample",
```

```
    "readOnly": false,
    "resources": [
        {
            "accountId": "123456789012",
            "type": "AWS::EC2::Snapshot",
            "ARN": "arn:aws:ec2:us-west-2::snapshot/snap-abcdef01234567890"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": false,
    "recipientAccountId": "123456789012",
    "eventCategory": "Data",
    "tlsDetails": {
        "tlsVersion": "TLSv1.2",
        "cipherSuite": "ECDHE-RSA-AES128-SHA",
        "clientProvidedHostHeader": "ebs.us-west-2.amazonaws.com"
    }
}
```

Pertanyaan umum

Apakah snapshot dapat diakses menggunakan API langsung EBS jika berstatus tertunda?

Tidak. Snapshot hanya dapat diakses jika memiliki status selesai.

Apakah indeks blok dikembalikan oleh EBS langsung sesuai urutan angka?

Ya. Indeks blok yang dikembalikan bersifat unik, dan dalam urutan numerik.

Dapatkah saya mengirimkan permintaan dengan nilai MaxResults parameter di bawah 100?

Tidak. Nilai MaxResult parameter minimum yang dapat Anda gunakan adalah 100. Jika Anda mengirimkan permintaan dengan nilai MaxResult parameter di bawah 100, dan ada lebih dari 100 blok dalam snapshot, maka API akan mengembalikan setidaknya 100 hasil.

Dapatkah saya menjalankan permintaan API secara bersamaan?

Anda dapat menjalankan permintaan API secara bersamaan. Pastikan untuk mencatat beban kerja lain yang mungkin berjalan di akun untuk menghindari hambatan. Anda juga harus membangun mekanisme coba ulang ke dalam alur kerja API langsung EBS untuk menangani throttling, waktu habis, dan ketidaktersediaan layanan. Untuk informasi selengkapnya, lihat Optimalkan performa.

Tinjau kuota layanan API langsung EBS untuk menentukan permintaan API yang dapat Anda jalankan per detik. Untuk informasi selengkapnya, lihat Titik Akhir dan Kuota Amazon Elastic Block Store dalam Referensi Umum AWS .

Saat menjalankan ListChangedBlocks aksi, apakah mungkin untuk mendapatkan respons kosong meskipun ada blok di snapshot?

Ya. Jika blok yang diubah berjumlah sedikit di snapshot, respons mungkin kosong tetapi API akan mengembalikan nilai token halaman berikutnya. Gunakan nilai token halaman berikutnya untuk melanjutkan ke halaman hasil berikutnya. Anda dapat mengonfirmasi bahwa Anda telah mencapai halaman terakhir hasil ketika API mengembalikan nilai token halaman berikutnya dari nol.

Jika NextToken parameter ditentukan bersama dengan StartingBlockIndex parameter, mana dari keduanya yang digunakan?

Yang NextToken digunakan, dan StartingBlockIndex diabaikan.

Berapa lama token blok dan token berikutnya berlaku?

Token blok berlaku selama tujuh hari, dan token berikutnya berlaku selama 60 menit.

Apakah snapshot terenkripsi didukung?

Ya. Snapshot yang dienkripsi dapat diakses menggunakan API langsung EBS.

Untuk mengakses snapshot terenkripsi, pengguna harus memiliki akses ke kunci KMS yang digunakan untuk mengenkripsi snapshot, dan tindakan dekripsi. AWS KMS Lihat Izin IAM untuk API langsung EBS bagian sebelumnya dalam panduan ini untuk AWS KMS kebijakan yang akan ditetapkan kepada pengguna.

Apakah snapshot publik didukung?

Snapshot publik tidak didukung.

Apakah snapshot lokal Amazon EBS pada Outposts didukung?

Snapshot lokal Amazon EBS pada Outposts tidak didukung.

Apakah blok snapshot daftar mengembalikan semua indeks blok dan token blok dalam snapshot, atau hanya yang memiliki data yang ditulis ke dalamnya?

Blok tersebut hanya mengembalikan indeks dan token blok yang memiliki data yang ditulis kepadanya.

Dapatkah saya memperoleh riwayat kunjungan API yang dibuat berdasarkan API langsung EBS di akun saya untuk tujuan analisis keamanan dan pemecahan masalah operasional?

Ya. Untuk menerima riwayat panggilan API API langsung EBS yang dilakukan di akun Anda, aktifkan AWS CloudTrail di AWS Management Console. Untuk informasi selengkapnya, lihat Log API Panggilan untuk API langsung EBS dengan AWS CloudTrail.

Keamanan di Amazon Elastic Block Store

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 Elastic Block Store, lihat AWS Layanan dalam Lingkup berdasarkan 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 Amazon EBS. Topik berikut menunjukkan cara mengonfigurasi Amazon EBS untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga mempelajari cara menggunakan AWS layanan lain yang membantu Anda memantau dan mengamankan sumber daya Amazon EBS Anda.

Topik

Perlindungan data di Amazon Elastic Block Store

Model tanggung jawab AWS bersama model berlaku untuk perlindungan data di Amazon Elastic Block Store. Seperti yang dijelaskan dalam model AWS ini, bertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS Cloud. Anda bertanggung jawab untuk

memelihara kendali atas isi yang dihost 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 kredensyal 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 sumber daya. AWS Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Siapkan API dan pencatatan aktivitas pengguna dengan AWS CloudTrail.
- Gunakan solusi AWS enkripsi, bersama dengan semua kontrol keamanan default di dalamnya Layanan AWS.
- Gunakan layanan keamanan terkelola lanjut seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.
- Jika Anda memerlukan modul kriptografi tervalidasi FIPS 140-2 saat mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Lihat informasi yang lebih lengkap tentang titik akhir FIPS yang tersedia di Standar Pemrosesan Informasi Federal (FIPS) 140-2.

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 Amazon EBS atau lainnya Layanan AWS menggunakan konsol, API AWS CLI, atau AWS SDK. 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.

Topik

- Keamanan data Amazon EBS
- Enkripsi saat istirahat dan dalam transit

- Manajemen kunci KMS

Keamanan data Amazon EBS

Volume Amazon EBS disajikan kepada Anda sebagai perangkat blok mentah yang tidak terformat. Perangkat-perangkat ini adalah perangkat logis yang dibuat pada infrastruktur EBS dan layanan Amazon EBS akan memastikan bahwa perangkat-perangkat tersebut secara logis kosong (yakni bahwa, blok mentah tersebut sudah dikosongkan atau mengandung data pseudorandom secara kriptografis) sebelum digunakan atau digunakan kembali oleh pelanggan.

Jika Anda memiliki prosedur yang mengharuskan semua data dihapus menggunakan metode tertentu, baik setelah atau sebelum digunakan (atau keduanya), seperti yang dirinci dalam DoD 5220.22-M (Manual Operasi Program Keamanan Industri Nasional) atau NIST 800-88(Pedoman untuk Sanitisasi Media), Anda memiliki kemampuan untuk melakukannya di Amazon EBS. Aktivitas tingkat blok tersebut akan tercermin ke media penyimpanan yang mendasarinya dalam layanan Amazon EBS tersebut.

Enkripsi saat istirahat dan dalam transit

Enkripsi Amazon EBS adalah solusi enkripsi yang memungkinkan Anda mengenkripsi volume Amazon EBS dan snapshot Amazon EBS menggunakan kunci kriptografi. AWS Key Management Service Operasi enkripsi EBS terjadi pada server yang meng-host instans Amazon EC2, memastikan keamanan data-at-restkeduanya data-in-transitdan antara instans dan volume terlampir dan setiap snapshot berikutnya. Untuk informasi selengkapnya, lihat Enkripsi EBS Amazon.

Manajemen kunci KMS

Saat membuat volume atau snapshot Amazon EBS terenkripsi, Anda menentukan kunci. AWS Key Management Service Secara default, Amazon EBS menggunakan kunci KMS AWS terkelola untuk Amazon EBS di akun dan Region () Anda. `aws/ebs` Namun, Anda dapat menentukan kunci KMS yang dikelola pelanggan yang Anda buat dan kelola. Menggunakan kunci KMS yang dikelola pelanggan memberi Anda lebih banyak fleksibilitas, termasuk kemampuan untuk membuat, memutar, dan menonaktifkan kunci KMS.

Untuk menggunakan kunci KMS yang dikelola pelanggan, Anda harus memberikan izin kepada pengguna untuk menggunakan kunci KMS. Untuk informasi selengkapnya, lihat Izin untuk pengguna.

> **Important**
>
> Amazon EBS hanya mendukung kunci KMS simetris. Anda tidak dapat menggunakan kunci KMS asimetris untuk mengenkripsi volume dan snapshot Amazon EBS. Untuk bantuan menentukan apakah kunci KMS simetris atau asimetris, lihat Mengidentifikasi kunci KMS asimetris.

Untuk setiap volume, Amazon EBS meminta AWS KMS untuk menghasilkan kunci data unik yang dienkripsi di bawah kunci KMS yang Anda tentukan. Amazon EBS menyimpan kunci data terenkripsi dengan volume. Kemudian, saat Anda melampirkan volume ke instans Amazon EC2, Amazon EBS memanggil AWS KMS untuk mendekripsi kunci data. Amazon EBS menggunakan kunci data plaintext dalam memori hypervisor untuk mengenkripsi semua I/O ke volume. Lihat informasi yang lebih lengkap di Cara kerja enkripsi EBS.

Manajemen identitas dan akses untuk Amazon Elastic Block Store

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 diotorisasi (memiliki izin) untuk menggunakan sumber daya Amazon EBS. IAM adalah Layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

Topik

- Audiens
- Mengautentikasi dengan identitas
- Mengelola akses menggunakan kebijakan
- Bagaimana Amazon Elastic Block Store bekerja dengan IAM
- Contoh kebijakan berbasis identitas untuk Amazon Elastic Block Store
- Memecahkan masalah identitas dan akses Amazon EBS

Audiens

Cara Anda menggunakan AWS Identity and Access Management (IAM) berbeda, tergantung pada pekerjaan yang Anda lakukan di Amazon EBS.

Pengguna layanan — Jika Anda menggunakan layanan Amazon EBS untuk melakukan pekerjaan Anda, administrator Anda memberi Anda kredensi dan izin yang Anda butuhkan. Saat Anda menggunakan lebih banyak fitur Amazon EBS 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 Amazon EBS, lihatMemecahkan masalah identitas dan akses Amazon EBS.

Administrator layanan - Jika Anda bertanggung jawab atas sumber daya Amazon EBS di perusahaan Anda, Anda mungkin memiliki akses penuh ke Amazon EBS. Tugas Anda adalah menentukan fitur dan sumber daya Amazon EBS mana yang harus diakses pengguna layanan Anda. Kemudian, Anda harus mengirimkan permintaan kepada administrator IAM Anda untuk mengubah izin pengguna layanan Anda. Tinjau informasi di halaman ini untuk memahami konsep Basic IAM. Untuk mempelajari lebih lanjut tentang bagaimana perusahaan Anda dapat menggunakan IAM dengan Amazon EBS, lihat. Bagaimana Amazon Elastic Block Store bekerja dengan IAM

Administrator IAM - Jika Anda seorang administrator IAM, Anda mungkin ingin mempelajari detail tentang cara menulis kebijakan untuk mengelola akses ke Amazon EBS. Untuk melihat contoh kebijakan berbasis identitas Amazon EBS yang dapat Anda gunakan di IAM, lihat. Contoh kebijakan berbasis identitas untuk Amazon Elastic Block Store

Mengautentikasi dengan identitas

Otentikasi adalah cara Anda masuk AWS menggunakan kredensyal 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 kredensi 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 gabungan, 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 kredensil Anda. Jika Anda tidak menggunakan AWS alat, Anda harus menandatangani permintaan sendiri. Untuk informasi selengkapnya tentang penggunaan metode yang disarankan untuk menandatangani permintaan sendiri, lihat Menandatangani permintaan AWS API di 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 lebih lanjut, lihat Autentikasi multi-faktor dalam Panduan Pengguna AWS IAM Identity Center dan Menggunakan autentikasi multi-faktor (MFA) di AWS 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 Anda. Lindungi kredensial pengguna root Anda dan gunakan kredensial tersebut untuk melakukan tugas yang hanya dapat dilakukan pengguna root. Untuk daftar tugas lengkap yang mengharuskan Anda masuk sebagai pengguna root, lihat Tugas yang memerlukan kredensial pengguna root dalam Panduan Pengguna IAM.

Identitas terfederasi

Sebagai praktik terbaik, mewajibkan pengguna manusia, termasuk pengguna yang memerlukan akses administrator, untuk menggunakan federasi dengan penyedia identitas untuk mengakses Layanan AWS dengan menggunakan kredensi 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 kredensil yang disediakan melalui sumber identitas. Ketika identitas federasi mengakses Akun AWS, mereka mengambil peran, dan peran memberikan kredensi sementara.

Untuk pengelolaan akses terpusat, sebaiknya 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 Apa yang dimaksud 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, sebaiknya andalkan kredensial temporer, dan bukan membuat pengguna IAM yang memiliki kredensial jangka panjang seperti kata sandi dan kunci akses. Namun, jika Anda memiliki kasus penggunaan khusus yang memerlukan kredensial jangka panjang dengan pengguna IAM, sebaiknya rotasikan 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 kumpulan pengguna IAM. Anda tidak dapat masuk sebagai grup. Anda dapat menggunakan grup untuk menentukan izin untuk beberapa pengguna sekaligus. Grup membuat izin lebih mudah dikelola untuk sekelompok besar pengguna. Misalnya, Anda dapat memiliki grup yang bernama 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 tersebut dimaksudkan untuk dapat diambil oleh siapa pun yang membutuhkannya. Pengguna memiliki kredensial jangka panjang permanen, tetapi peran memberikan kredensial sementara. Untuk mempelajari selengkapnya, silakan lihat Kapan harus membuat pengguna IAM (bukan peran) 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. Anda dapat mengambil peran IAM untuk sementara AWS Management Console dengan beralih peran. Anda dapat mengambil peran dengan memanggil operasi AWS CLI atau AWS API atau dengan menggunakan URL kustom. Untuk informasi selengkapnya tentang metode untuk menggunakan peran, lihat Menggunakan peran IAM dalam Panduan Pengguna IAM.

Peran IAM dengan kredensial sementara berguna dalam situasi berikut:

- Akses pengguna gabungan – Untuk menetapkan izin ke sebuah identitas gabungan, Anda dapat membuat peran dan menentukan izin untuk peran tersebut. Saat identitas terfederasi diautentikasi, identitas tersebut dikaitkan dengan peran dan diberikan izin yang ditentukan oleh peran. Untuk informasi tentang peran untuk federasi, lihat Membuat peran untuk Penyedia Identitas pihak ketiga dalam Panduan Pengguna IAM. Jika Anda menggunakan Pusat Identitas IAM, Anda mengonfigurasi sekumpulan izin. Untuk mengontrol apa yang dapat diakses identitas Anda setelah

identitas tersebut diautentikasi, Pusat Identitas IAM mengaitkan izin yang ditetapkan ke peran dalam IAM. Untuk informasi tentang rangkaian izin, lihat Rangkaian 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 (pengguna utama tepercaya) dengan akun berbeda untuk mengakses sumber daya yang ada 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 kebijakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat Bagaimana peran IAM berbeda dari kebijakan berbasis sumber daya dalam Panduan Pengguna IAM.
- Akses lintas layanan — Beberapa Layanan AWS menggunakan fitur lain Layanan AWS. Contoh, ketika Anda melakukan panggilan dalam layanan, umumnya layanan tersebut menjalankan aplikasi di Amazon EC2 atau menyimpan objek di Amazon S3. Suatu layanan mungkin melakukan hal tersebut menggunakan izin pengguna utama panggilan, 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 tindakan yang kemudian memulai 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 Meneruskan sesi akses.
  - Peran IAM – Peran layanan adalah peran IAM yang diambil layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, memodifikasi, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat Membuat 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 mengambil peran untuk melakukan sebuah 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 kredensi sementara untuk aplikasi yang berjalan pada instans EC2 dan membuat atau permintaan API. AWS CLI AWS Cara ini lebih dianjurkan daripada menyimpan kunci akses dalam instans EC2. Untuk menetapkan AWS peran ke instans EC2 dan membuatnya tersedia untuk semua aplikasinya, Anda membuat profil instance yang dilampirkan ke instance. Profil instans berisi peran dan memungkinkan program yang berjalan di instans EC2 mendapatkan kredensial sementara. Untuk informasi selengkapnya, lihat Menggunakan peran IAM untuk memberikan izin ke aplikasi yang berjalan di instans Amazon EC2 dalam Panduan Pengguna IAM.

Untuk mempelajari apakah kita harus menggunakan peran IAM atau pengguna IAM, lihat Kapan harus membuat peran IAM (bukan pengguna) dalam 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 Ikhtisar kebijakan JSON dalam Panduan Pengguna IAM.

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

Secara default, pengguna dan peran tidak memiliki izin. Untuk memberikan izin kepada pengguna untuk melakukan tindakan pada sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM. Administrator kemudian dapat menambahkan kebijakan IAM ke peran, dan pengguna dapat menjalankan peran.

Kebijakan IAM mendefinisikan izin untuk suatu tindakan terlepas dari metode yang Anda gunakan untuk operasi. Sebagai contoh, anggap saja 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 pengguna dan peran, di sumber daya mana, dan dengan ketentuan apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Membuat kebijakan IAM dalam Panduan Pengguna IAM.

Kebijakan berbasis identitas dapat dikategorikan lebih lanjut sebagai kebijakan inline atau kebijakan terkelola. 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 terkelola atau kebijakan inline, lihat Memilih antara kebijakan terkelola 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 yang dilampiri kebijakan tersebut, kebijakan ini menentukan jenis tindakan yang dapat dilakukan oleh pengguna utama tertentu di sumber daya tersebut dan apa ketentuannya. Anda harus menentukan pengguna utama 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 (ACL)

Daftar kontrol akses (ACL) mengendalikan pengguna utama mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACL sama dengan kebijakan berbasis sumber daya, meskipun tidak menggunakan format dokumen kebijakan JSON.

Amazon S3, AWS WAF, dan Amazon VPC adalah contoh layanan yang mendukung ACL. Untuk mempelajari ACL selengkapnya, silakan lihat Gambaran umum daftar kontrol akses (ACL) di Panduan Developer Layanan Penyimpanan Ringkas Amazon.

Tipe kebijakan lain

AWS mendukung jenis kebijakan tambahan yang kurang umum. Tipe-tipe kebijakan ini dapat mengatur izin maksimum yang diberikan kepada Anda berdasarkan tipe kebijakan yang lebih umum.

- Batasan izin – Batasan izin adalah fitur lanjutan di mana Anda menetapkan izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas kepada entitas IAM (pengguna 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 secara eksplisit terhadap salah satu kebijakan ini akan mengesampingkan izin tersebut. Untuk informasi selengkapnya tentang batasan izin, lihat Batasan izin untuk entitas IAM dalam Panduan Pengguna IAM.
- Kebijakan kontrol layanan (SCP) — SCP 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 organisasi, Anda dapat menerapkan kebijakan kontrol layanan (SCP) ke sebagian atau semua akun Anda. SCP membatasi izin untuk entitas di akun anggota, termasuk masing-masing. Pengguna root akun AWS Untuk informasi selengkapnya tentang Organisasi dan SCP, lihat Cara kerja SCP dalam Panduan Pengguna AWS Organizations .
- Kebijakan sesi – Kebijakan sesi adalah kebijakan lanjutan yang Anda teruskan sebagai parameter saat Anda membuat sesi sementara secara terprogram untuk peran atau pengguna gabungan. 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 di salah satu kebijakan ini akan membatalkan izin tersebut. Untuk informasi selengkapnya, lihat Kebijakan sesi dalam Panduan Pengguna IAM.

Berbagai jenis kebijakan

Jika beberapa jenis kebijakan diberlakukan untuk satu 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 Amazon Elastic Block Store bekerja dengan IAM

Sebelum Anda menggunakan IAM untuk mengelola akses ke Amazon EBS, pelajari fitur IAM apa yang tersedia untuk digunakan dengan Amazon EBS.

Fitur IAM yang dapat Anda gunakan dengan Amazon Elastic Block Store

Fitur IAM	Dukungan Amazon EBS
Kebijakan berbasis identitas	Ya
Kebijakan berbasis sumber daya	Tidak
Tindakan kebijakan	Ya
Sumber daya kebijakan	Ya
Kunci persyaratan kebijakan	Ya
ACL	Tidak
ABAC (tanda dalam kebijakan)	Parsial
Kredensial sementara	Ya
Izin pengguna utama	Ya
Peran layanan	Ya
Peran terkait layanan	Tidak

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

Kebijakan berbasis identitas untuk Amazon EBS

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 pengguna dan peran, di sumber daya mana, dan dengan ketentuan apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Membuat kebijakan IAM dalam Panduan Pengguna IAM.

Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan tindakan dan sumber daya yang diizinkan atau ditolak, serta ketentuan terkait jenis tindakan yang diizinkan atau ditolak. Anda tidak dapat menentukan pengguna utama dalam kebijakan berbasis identitas karena kebijakan ini berlaku untuk pengguna atau peran yang dilampiri kebijakan. Untuk mempelajari semua elemen yang dapat digunakan dalam kebijakan JSON, lihat Referensi elemen kebijakan JSON IAM dalam Panduan Pengguna IAM.

Contoh kebijakan berbasis identitas untuk Amazon EBS

Untuk melihat contoh kebijakan berbasis identitas Amazon EBS, lihat. Contoh kebijakan berbasis identitas untuk Amazon Elastic Block Store

Kebijakan berbasis sumber daya dalam Amazon EBS

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. Untuk sumber daya yang dilampiri kebijakan tersebut, kebijakan ini menentukan jenis tindakan yang dapat dilakukan oleh pengguna utama tertentu di sumber daya tersebut dan apa ketentuannya. Anda harus menentukan pengguna utama dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau. Layanan AWS

Untuk mengaktifkan akses lintas akun, Anda dapat menentukan seluruh akun atau entitas IAM di akun lain sebagai pengguna utama dalam kebijakan berbasis sumber daya. Menambahkan pengguna utama lintas akun ke kebijakan berbasis sumber daya bagian dari membangun hubungan kepercayaan. Ketika prinsipal dan sumber daya berbeda Akun AWS, administrator IAM di akun tepercaya juga harus memberikan izin entitas utama (pengguna atau peran) untuk mengakses sumber daya. Izin diberikan dengan melampirkan kebijakan berbasis identitas ke entitas tersebut.

Namun, jika kebijakan berbasis sumber daya memberikan akses kepada pengguna utama dalam akun yang sama, kebijakan berbasis identitas lainnya tidak diperlukan. Untuk informasi selengkapnya, lihat Perbedaan peran IAM dengan kebijakan berbasis sumber daya di Panduan Pengguna IAM.

Tindakan kebijakan untuk Amazon EBS

Mendukung tindakan kebijakan	Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, pengguna utama mana yang dapat melakukan tindakan pada sumber daya apa, dan dalam 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.

Menyertakan tindakan dalam suatu kebijakan untuk memberikan izin melakukan operasi terkait.

Untuk melihat daftar tindakan Amazon EBS, lihat Tindakan, sumber daya, dan kunci kondisi di Referensi Otorisasi Layanan.

Tindakan kebijakan di Amazon EBS menggunakan awalan berikut sebelum tindakan:

```
ec2
```

Untuk menetapkan secara spesifik beberapa tindakan dalam satu pernyataan, pisahkan tindakan-tindakan tersebut dengan koma.

```
"Action": [
  "ec2:action1",
  "ec2;:action2"
]
```

Untuk melihat contoh kebijakan berbasis identitas Amazon EBS, lihat. Contoh kebijakan berbasis identitas untuk Amazon Elastic Block Store

Sumber daya kebijakan untuk Amazon EBS

Mendukung sumber daya kebijakan	Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, pengguna utama mana yang dapat melakukan tindakan pada sumber daya apa, dan dalam kondisi apa.

Elemen kebijakan JSON `Resource` menentukan objek atau beberapa 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 mengindikasikan bahwa pernyataan tersebut berlaku untuk semua sumber daya.

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

Untuk melihat daftar jenis sumber daya Amazon EBS dan ARNnya, lihat Sumber Daya yang Ditentukan oleh Amazon Elastic Block Store di Referensi Otorisasi Layanan. Untuk mempelajari tindakan mana yang dapat Anda tentukan ARN dari setiap sumber daya, lihat Tindakan yang Ditentukan oleh Amazon Elastic Block Store.

Beberapa tindakan Amazon EBS API mendukung beberapa sumber daya. Untuk menentukan beberapa sumber daya dalam satu pernyataan, pisahkan ARN dengan koma. Misalnya, `DescribeVolumes` mengakses vol-01234567890abcdef dan vol-09876543210fedcba, jadi prinsipal harus memiliki izin untuk mengakses kedua sumber daya.

```
"Resource": [
  "arn:aws:ec2:us-east-1:123456789012:volume/vol-01234567890abcdef",
  "arn:aws:ec2:us-east-1:123456789012:volume/vol-09876543210fedcba"
]
```

Kunci kondisi kebijakan untuk Amazon EBS

Mendukung kunci kondisi kebijakan spesifik layanan	Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, pengguna utama mana yang dapat melakukan tindakan pada sumber daya apa, dan dalam kondisi apa.

Elemen `Condition` (atau blok `Condition`) memungkinkan Anda menentukan kondisi di mana suatu pernyataan akan diterapkan. Elemen `Condition` bersifat opsional. Anda dapat membuat ekspresi kondisional 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 satu pernyataan, atau beberapa kunci dalam satu elemen `Condition`, AWS akan mengevaluasinya dengan 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, silakan lihat Elemen kebijakan IAM: variabel dan tanda di 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.

Misalnya, kondisi berikut memungkinkan prinsipal untuk melakukan tindakan pada volume hanya jika jenis volumenya`gp2`.

```
"Condition":{
  "StringLikeIfExists":{
    "ec2:VolumeType":"gp2"
  }
}
```

Untuk melihat daftar kunci kondisi Amazon EBS, lihat Kunci tindakan, sumber daya, dan kondisi di Referensi Otorisasi Layanan. Untuk mempelajari tindakan dan sumber daya yang dapat Anda gunakan kunci kondisi, lihat Tindakan yang Ditentukan oleh Amazon Elastic Block Store.

ACL di Amazon EBS

Mendukung ACL	Tidak

Daftar kontrol akses (ACL) mengontrol pengguna utama (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACL sama dengan kebijakan berbasis sumber daya, meskipun tidak menggunakan format dokumen kebijakan JSON.

ABAC dengan Amazon EBS

Mendukung ABAC (tanda dalam kebijakan)	Parsial

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. Pemberian tanda ke entitas dan sumber daya adalah langkah pertama dari ABAC. Kemudian rancanglah kebijakan ABAC untuk mengizinkan operasi-operasi ketika tanda milik pengguna utama cocok dengan tanda yang ada di sumber daya yang ingin diakses.

ABAC sangat berguna di lingkungan yang berkembang dengan cepat dan berguna dalam situasi di mana pengelolaan kebijakan menjadi rumit.

Untuk mengendalikan akses berdasarkan tag, berikan informasi tentang tanda di elemen syarat dari sebuh kebijakan dengan menggunakan kunci-kunci persyaratan `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 hanya untuk beberapa jenis sumber daya, nilainya adalah Parsial.

Untuk informasi selengkapnya tentang ABAC, lihat Apa itu ABAC? di Panduan Pengguna IAM. Untuk melihat tutorial terkait langkah-langkah penyiapan ABAC, lihat Menggunakan kontrol akses berbasis atribut (ABAC) di Panduan Pengguna IAM.

Menggunakan kredensyal sementara dengan Amazon EBS

Mendukung kredensial sementara	Ya

Beberapa Layanan AWS tidak berfungsi saat Anda masuk menggunakan kredensyal 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 membuat kredensial sementara secara otomatis saat masuk ke konsol sebagai pengguna dan kemudian beralih peran. Untuk informasi selengkapnya tentang cara beralih peran, lihat Beralih peran (konsol) di Panduan Pengguna IAM.

Anda dapat membuat kredensyal sementara secara manual menggunakan API AWS CLI atau AWS . Anda kemudian dapat menggunakan kredensyal 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 Amazon EBS

Mendukung sesi akses maju (FAS)	Ya

Saat Anda menggunakan pengguna atau peran IAM untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan tindakan yang kemudian memulai 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 Meneruskan sesi akses.

Peran layanan untuk Amazon EBS

Mendukung peran layanan	Ya

Peran layanan adalah peran IAM yang diambil oleh layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, mengubah, dan menghapus peran layanan dari dalam IAM.

Untuk informasi selengkapnya, lihat Membuat peran untuk mendelegasikan izin ke Layanan AWS dalam Panduan pengguna IAM.

> **Warning**
>
> Mengubah izin untuk peran layanan dapat merusak fungsionalitas Amazon EBS. Edit peran layanan hanya jika Amazon EBS memberikan panduan untuk melakukannya.

Peran terkait layanan untuk Amazon EBS

Mendukung peran terkait layanan	Tidak

Peran terkait layanan adalah jenis peran layanan yang ditautkan ke. Layanan AWS Layanan tersebut dapat mengambil peran untuk melakukan sebuah 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 pengelolaan peran terkait layanan, lihat Layanan AWS yang berfungsi dengan IAM. Temukan sebuah 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 Amazon Elastic Block Store

Secara default, pengguna dan peran tidak memiliki izin untuk membuat atau memodifikasi sumber daya Amazon EBS. 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 pada sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM. Administrator kemudian dapat menambahkan kebijakan IAM ke peran, dan pengguna dapat menjalankan peran.

Untuk mempelajari cara membuat kebijakan berbasis identitas IAM menggunakan contoh dokumen kebijakan JSON ini, lihat Membuat kebijakan IAM dalam Panduan Pengguna IAM.

Untuk detail tentang tindakan dan jenis sumber daya yang ditentukan oleh Amazon EBS, termasuk format ARN untuk setiap jenis sumber daya, lihat Tindakan, Sumber Daya, dan Kunci Kondisi untuk Amazon Elastic Block Store di Referensi Otorisasi Layanan.

Topik

- Praktik terbaik kebijakan
- Menggunakan konsol Amazon EBS
- Izinkan pengguna melihat izin mereka sendiri
- Cara menggunakan volume
- Cara menggunakan snapshot

Praktik terbaik kebijakan

Kebijakan berbasis identitas menentukan apakah seseorang dapat membuat, mengakses, atau menghapus sumber daya Amazon EBS di akun Anda. Tindakan ini dikenai biaya untuk Akun AWS Anda. 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 pekerjaan di 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 melakukan ini dengan menentukan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi tertentu, juga dikenal sebagai izin hak akses paling rendah. Untuk informasi selengkapnya tentang cara menggunakan IAM untuk menerapkan izin, lihat Kebijakan dan izin di IAM di Panduan Pengguna IAM.
- Gunakan kondisi dalam kebijakan IAM untuk membatasi akses lebih lanjut – Anda dapat menambahkan kondisi ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. Misalnya, Anda dapat menulis syarat kebijakan untuk menentukan bahwa semua pengajuan 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: Syarat di Panduan Pengguna IAM.

- Menggunakan IAM Access Analyzer untuk memvalidasi kebijakan IAM Anda guna memastikan izin yang aman dan berfungsi – 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 Analizer Akses IAM di 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 mewajibkan MFA saat operasi API dipanggil, tambahkan kondisi MFA pada kebijakan Anda. Untuk informasi selengkapnya, lihat Mengonfigurasi akses API yang dilindungi MFA di Panduan Pengguna IAM.

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

Menggunakan konsol Amazon EBS

Untuk mengakses konsol Amazon Elastic Block Store, Anda harus memiliki set izin minimum. Izin ini harus memungkinkan Anda untuk membuat daftar dan melihat detail tentang sumber daya Amazon EBS 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. Sebaliknya, izinkan akses hanya ke tindakan yang cocok dengan operasi API yang coba dilakukan.

Untuk memastikan bahwa pengguna dan peran masih dapat menggunakan konsol Amazon EBS, lampirkan juga Amazon EBS `ConsoleAccess` atau kebijakan `ReadOnly` AWS terkelola ke entitas. Untuk informasi selengkapnya, lihat Menambahkan izin ke pengguna di Panduan Pengguna IAM.

Izinkan pengguna melihat izin mereka sendiri

Contoh ini menunjukkan cara membuat kebijakan yang mengizinkan para pengguna IAM melihat kebijakan inline dan terkelola yang dilampirkan ke identitas pengguna mereka. Kebijakan ini mencakup izin untuk menyelesaikan tindakan ini di konsol atau menggunakan API atau secara terprogram. AWS CLI AWS

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Cara menggunakan volume

Contoh

- Contoh: Melampirkan dan melepaskan lampiran volume
- Contoh: Membuat volume
- Contoh: Membuat volume dengan tanda
- Contoh: Bekerja dengan volume menggunakan konsol Amazon EC2

Contoh: Melampirkan dan melepaskan lampiran volume

Ketika tindakan API memerlukan pemerintah untuk menentukan beberapa sumber daya, Anda harus membuat pernyataan kebijakan yang memungkinkan para pengguna mengakses semua sumber daya yang diperlukan. Jika Anda harus menggunakan elemen `Condition` dengan satu atau beberapa sumber daya ini, maka Anda harus membuat beberapa pernyataan seperti yang ditunjukkan dalam contoh berikut ini.

Kebijakan berikut memungkinkan pengguna untuk melampirkan volume dengan tag "`volume_user=`iam-user-name" ke instance dengan tag "`department=dev`“, dan untuk melepaskan volume tersebut dari instance tersebut. Jika Anda melampirkan kebijakan ini ke grup IAM, variabel kebijakan `aws:username` akan memberikan izin kepada setiap pengguna yang ada dalam grup untuk melampirkan atau melepaskan volume dari instans dengan nama tanda `volume_user` yang menjadikan nama pengguna IAM-nya sebagai nilai.

```
{
   "Version": "2012-10-17",
   "Statement": [
       {
      "Effect": "Allow",
      "Action": [
        "ec2:AttachVolume",
        "ec2:DetachVolume"
      ],
      "Resource": "arn:aws:ec2:us-east-1:account-id:instance/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/department": "dev"
        }
      }
   },
   {
      "Effect": "Allow",
      "Action": [
        "ec2:AttachVolume",
        "ec2:DetachVolume"
      ],
      "Resource": "arn:aws:ec2:us-east-1:account-id:volume/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/volume_user": "${aws:username}"
        }
```

```
      }
    }
  ]
}
```

Contoh: Membuat volume

Kebijakan berikut memungkinkan pengguna untuk menggunakan tindakan API CreateVolume. Pengguna diperbolehkan untuk membuat volume hanya jika volume tersebut dienkripsi dan hanya jika ukuran volume kurang dari 20 GiB.

```
{
  "Version": "2012-10-17",
  "Statement": [
        {
      "Effect": "Allow",
      "Action": [
         "ec2:CreateVolume"
      ],
      "Resource": "arn:aws:ec2:us-east-1:account-id:volume/*",
      "Condition":{
         "NumericLessThan": {
             "ec2:VolumeSize" : "20"
          },
          "Bool":{
              "ec2:Encrypted" : "true"
          }
       }
    }
  ]
}
```

Contoh: Membuat volume dengan tanda

Kebijakan berikut mencakup kunci syarat `aws:RequestTag` yang mengharuskan para pengguna untuk menandai setiap volume yang mereka buat dengan tanda `costcenter=115` dan `stack=prod`. Jika pengguna tidak meneruskan tanda tertentu ini, atau jika mereka tidak menentukan tanda sama sekali, maka permintaan itu akan gagal.

Untuk tindakan-tindakan yang digunakan untuk membuat sumber daya yang menerapkan tanda, para pengguna juga harus memiliki izin untuk menggunakan tindakan `CreateTags`. Pernyataan kedua menggunakan kunci syarat `ec2:CreateAction` untuk memungkinkan para pengguna membuat

tanda hanya dalam konteks `CreateVolume`. Para pengguna tidak dapat menandai volume yang ada atau sumber daya lainnya.

```
{
  "Version": "2012-10-17",
  "Statement": [
        {
      "Sid": "AllowCreateTaggedVolumes",
      "Effect": "Allow",
      "Action": "ec2:CreateVolume",
      "Resource": "arn:aws:ec2:us-east-1:account-id:volume/*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/costcenter": "115",
          "aws:RequestTag/stack": "prod"
         }
       }
     },
     {
       "Effect": "Allow",
       "Action": [
         "ec2:CreateTags"
       ],
       "Resource": "arn:aws:ec2:us-east-1:account-id:volume/*",
       "Condition": {
         "StringEquals": {
             "ec2:CreateAction" : "CreateVolume"
         }
       }
     }
  ]
}
```

Kebijakan berikut memungkinkan para pengguna untuk membuat volume tanpa harus menentukan tanda. Tindakan `CreateTags` akan dievaluasi hanya jika tanda ditentukan dalam permintaan `CreateVolume`. Jika para pengguna menentukan tanda, maka tanda tersebut harus `purpose=test`. Tidak ada tanda lain yang diperbolehkan dalam permintaan.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```
      "Effect": "Allow",
      "Action": "ec2:CreateVolume",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
         "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:us-east-1:account-id:volume/*",
      "Condition": {
         "StringEquals": {
             "aws:RequestTag/purpose": "test",
             "ec2:CreateAction" : "CreateVolume"
          },
         "ForAllValues:StringEquals": {
             "aws:TagKeys": "purpose"
          }
       }
    }
  ]
}
```

Contoh: Bekerja dengan volume menggunakan konsol Amazon EC2

Kebijakan berikut memberi pengguna izin untuk melihat dan membuat volume, serta melampirkan serta melepaskan volume ke instans tertentu menggunakan konsol Amazon EC2.

Para pengguna dapat melampirkan volume apa pun ke instans yang memiliki tanda "`purpose=test`" dan juga melepaskan volume yang dilampirkan dari instans tersebut. Untuk melampirkan volume menggunakan konsol Amazon EC2, akan sangat membantu bagi para pengguna jika mereka memiliki izin untuk menggunakan tindakan `ec2:DescribeInstances`, karena hal ini akan memungkinkan mereka memilih instans dari daftar yang sudah diisi sebelumnya dalam kotak dialog Lampirkan Volume. Akan tetapi, hal ini juga akan memungkinkan para pengguna untuk menampilkan semua instans dalam halaman Instans dalam konsol tersebut, sehingga Anda dapat menghilangkan tindakan ini.

Dalam pernyataan pertama, tindakan `ec2:DescribeAvailabilityZones` diperlukan untuk memastikan bahwa seorang pengguna dapat memilih Zona Ketersediaan saat membuat volume.

Para pengguna tidak dapat memberikan tanda pada volume-volume yang mereka buat (baik ketika volume sedang dibuat maupun setelah volume dibuat).

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeVolumes",
        "ec2:DescribeAvailabilityZones",
        "ec2:CreateVolume",
        "ec2:DescribeInstances"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:AttachVolume",
        "ec2:DetachVolume"
      ],
      "Resource": "arn:aws:ec2:region:111122223333:instance/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/purpose": "test"
        }
     }
   },
   {
      "Effect": "Allow",
      "Action": [
        "ec2:AttachVolume",
        "ec2:DetachVolume"
      ],
      "Resource": "arn:aws:ec2:region:111122223333:volume/*"
   }
   ]
}
```

Cara menggunakan snapshot

Berikut ini adalah contoh kebijakan untuk keduanya `CreateSnapshot` (point-in-timesnapshot dari volume EBS) dan `CreateSnapshots` (snapshot multi-volume).

Contoh-contoh

- Contoh: Membuat snapshot
- Contoh: Membuat beberapa snapshot
- Contoh: Membuat snapshot dengan tanda
- Contoh: Membuat snapshot multivolume dengan tanda
- Contoh: Menyalin beberapa snapshot
- Contoh: Memodifikasi pengaturan izin untuk snapshot

Contoh: Membuat snapshot

Kebijakan berikut memungkinkan pelanggan untuk menggunakan tindakan API CreateSnapshot. Pelanggan dapat membuat snapshot hanya jika volume sudah dienkripsi dan hanya jika ukuran volume kurang dari 20 GiB.

```
{
   "Version":"2012-10-17",
   "Statement": [
      {
         "Effect":"Allow",
         "Action":"ec2:CreateSnapshot",
         "Resource":"arn:aws:ec2:us-east-1::snapshot/*"
      },
      {
         "Effect":"Allow",
         "Action":"ec2:CreateSnapshot",
         "Resource":"arn:aws:ec2:us-east-1:account-id:volume/*",
         "Condition":{
            "NumericLessThan":{
               "ec2:VolumeSize":"20"
            },
            "Bool":{
               "ec2:Encrypted":"true"
            }
         }
      }
   ]
}
```

Contoh: Membuat beberapa snapshot

Kebijakan berikut memungkinkan pelanggan untuk menggunakan tindakan API CreateSnapshots. Pelanggan dapat membuat beberapa snapshot hanya jika semua volume pada instans memiliki tipe GP2.

```
{
   "Version":"2012-10-17",
   "Statement": [
      {
         "Effect":"Allow",
         "Action":"ec2:CreateSnapshots",
         "Resource":[
"arn:aws:ec2:us-east-1::snapshot/*",
"arn:aws:ec2:*:*:instance/*"
   ]
      },
      {
         "Effect":"Allow",
         "Action":"ec2:CreateSnapshots",
         "Resource":"arn:aws:ec2:us-east-1:*:volume/*",
         "Condition":{
            "StringLikeIfExists":{
               "ec2:VolumeType":"gp2"
             }
     }

      }
   ]
}
```

Contoh: Membuat snapshot dengan tanda

Kebijakan berikut mencakup kunci syarat `aws:RequestTag` yang mengharuskan pelanggan menerapkan tanda `costcenter=115` dan `stack=prod` ke setiap snapshot baru. Jika pengguna tidak meneruskan tanda tertentu ini, atau jika mereka tidak menentukan tanda sama sekali, maka permintaan itu akan gagal.

Untuk tindakan-tindakan yang digunakan untuk membuat sumber daya yang menerapkan tanda, pelanggan juga harus memiliki izin untuk menggunakan tindakan `CreateTags`. Pernyataan ketiga menggunakan kunci syarat `ec2:CreateAction` untuk memungkinkan para pelanggan membuat

tanda hanya dalam konteks `CreateSnapshot`. Para pelanggan tidak dapat menandai volume yang ada atau sumber daya lainnya.

```
{
   "Version":"2012-10-17",
   "Statement": [
      {
         "Effect":"Allow",
         "Action":"ec2:CreateSnapshot",
         "Resource":"arn:aws:ec2:us-east-1:account-id:volume/*"
      },
      {
         "Sid":"AllowCreateTaggedSnapshots",
         "Effect":"Allow",
         "Action":"ec2:CreateSnapshot",
         "Resource":"arn:aws:ec2:us-east-1::snapshot/*",
         "Condition":{
            "StringEquals":{
               "aws:RequestTag/costcenter":"115",
               "aws:RequestTag/stack":"prod"
            }
         }
      },
      {
         "Effect":"Allow",
         "Action":"ec2:CreateTags",
         "Resource":"arn:aws:ec2:us-east-1::snapshot/*",
         "Condition":{
            "StringEquals":{
               "ec2:CreateAction":"CreateSnapshot"
            }
         }
      }
   ]
}
```

Contoh: Membuat snapshot multivolume dengan tanda

Kebijakan berikut mencakup kunci syarat `aws:RequestTag` yang mengharuskan pelanggan menerapkan tanda `costcenter=115` dan `stack=prod` saat membuat set snapshot multivolume. Jika pengguna tidak meneruskan tanda tertentu ini, atau jika mereka tidak menentukan tanda sama sekali, maka permintaan itu akan gagal.

```
{
   "Version":"2012-10-17",
   "Statement": [
      {
         "Effect":"Allow",
         "Action":"ec2:CreateSnapshots",
         "Resource":[
"arn:aws:ec2:us-east-1::snapshot/*",
"arn:aws:ec2:*:*:instance/*",
"arn:aws:ec2:*:*:volume/*"

   ]
      },
      {
         "Sid":"AllowCreateTaggedSnapshots",
         "Effect":"Allow",
         "Action":"ec2:CreateSnapshots",
         "Resource":"arn:aws:ec2:us-east-1::snapshot/*",
         "Condition":{
            "StringEquals":{
               "aws:RequestTag/costcenter":"115",
               "aws:RequestTag/stack":"prod"
            }
         }
      },
      {
         "Effect":"Allow",
         "Action":"ec2:CreateTags",
         "Resource":"arn:aws:ec2:us-east-1::snapshot/*",
         "Condition":{
            "StringEquals":{
               "ec2:CreateAction":"CreateSnapshots"
            }
         }
      }
   ]
}
```

Kebijakan berikut memungkinkan para pelanggan untuk membuat snapshot tanpa harus menentukan tanda. Tindakan `CreateTags` akan dievaluasi hanya jika tanda ditentukan dalam permintaan `CreateSnapshot` atau `CreateSnapshots`. Tanda dapat dihilangkan dalam permintaan. Jika tanda

ditentukan, maka tanda tersebut harus `purpose=test`. Tidak ada tanda lain yang diperbolehkan dalam permintaan.

```
{
   "Version":"2012-10-17",
   "Statement": [
      {
         "Effect":"Allow",
         "Action":"ec2:CreateSnapshot",
         "Resource":"*"
      },
      {
         "Effect":"Allow",
         "Action":"ec2:CreateTags",
         "Resource":"arn:aws:ec2:us-east-1::snapshot/*",
         "Condition":{
            "StringEquals":{
               "aws:RequestTag/purpose":"test",
               "ec2:CreateAction":"CreateSnapshot"
            },
            "ForAllValues:StringEquals":{
               "aws:TagKeys":"purpose"
            }
         }
      }
   ]
}
```

Kebijakan berikut memungkinkan pelanggan untuk membuat set snapshot multivolume tanpa perlu menentukan tanda. Tindakan `CreateTags` akan dievaluasi hanya jika tanda ditentukan dalam permintaan `CreateSnapshot` atau `CreateSnapshots`. Tanda dapat dihilangkan dalam permintaan. Jika tanda ditentukan, maka tanda tersebut harus `purpose=test`. Tidak ada tanda lain yang diperbolehkan dalam permintaan.

```
{
   "Version":"2012-10-17",
   "Statement": [
      {
         "Effect":"Allow",
         "Action":"ec2:CreateSnapshots",
         "Resource":"*"
      },
```

```
      {
         "Effect":"Allow",
         "Action":"ec2:CreateTags",
         "Resource":"arn:aws:ec2:us-east-1::snapshot/*",
         "Condition":{
            "StringEquals":{
               "aws:RequestTag/purpose":"test",
               "ec2:CreateAction":"CreateSnapshots"
            },
            "ForAllValues:StringEquals":{
               "aws:TagKeys":"purpose"
            }
         }
      }
   ]
}
```

Kebijakan berikut mengizinkan snapshot untuk dibuat hanya jika volume sumber diberi tanda dengan `User:`*`username`* untuk pelanggan, dan snapshot itu sendiri diberi tanda dengan `Environment:Dev` dan `User:`*`username`*. Pelanggan dapat menambahkan tanda tambahan untuk snapshot tersebut.

```
{
   "Version":"2012-10-17",
   "Statement": [
           {
         "Effect":"Allow",
         "Action":"ec2:CreateSnapshot",
         "Resource":"arn:aws:ec2:us-east-1:account-id:volume/*",
         "Condition":{
            "StringEquals":{
               "aws:ResourceTag/User":"${aws:username}"
            }
         }
      },
      {
         "Effect":"Allow",
         "Action":"ec2:CreateSnapshot",
         "Resource":"arn:aws:ec2:us-east-1::snapshot/*",
         "Condition":{
            "StringEquals":{
               "aws:RequestTag/Environment":"Dev",
               "aws:RequestTag/User":"${aws:username}"
            }
```

```
         }
      },
      {
         "Effect":"Allow",
         "Action":"ec2:CreateTags",
         "Resource":"arn:aws:ec2:us-east-1::snapshot/*"
      }
   ]
}
```

Kebijakan untuk `CreateSnapshots` berikut mengizinkan snapshot untuk dibuat hanya jika volume sumber diberi tanda dengan `User:`*`username`* untuk pelanggan, dan snapshot itu sendiri diberi tanda dengan `Environment:Dev` dan `User:`*`username`*.

```
{
   "Version":"2012-10-17",
   "Statement": [
      {
         "Effect":"Allow",
         "Action":"ec2:CreateSnapshots",
         "Resource":"arn:aws:ec2:us-east-1:*:instance/*",
 },
      {
         "Effect":"Allow",
         "Action":"ec2:CreateSnapshots",
         "Resource":"arn:aws:ec2:us-east-1:account-id:volume/*",
         "Condition":{
            "StringEquals":{
               "aws:ResourceTag/User":"${aws:username}"
            }
         }
      },
      {
         "Effect":"Allow",
         "Action":"ec2:CreateSnapshots",
         "Resource":"arn:aws:ec2:us-east-1::snapshot/*",
         "Condition":{
            "StringEquals":{
               "aws:RequestTag/Environment":"Dev",
               "aws:RequestTag/User":"${aws:username}"
            }
         }
      },
```

```
      {
         "Effect":"Allow",
         "Action":"ec2:CreateTags",
         "Resource":"arn:aws:ec2:us-east-1::snapshot/*"
      }
   ]
}
```

Kebijakan berikut memungkinkan penghapusan snapshot hanya jika snapshot tersebut diberi tanda dengan Pengguna:username untuk pelanggan.

```
{
   "Version":"2012-10-17",
   "Statement": [
            {
         "Effect":"Allow",
         "Action":"ec2:DeleteSnapshot",
         "Resource":"arn:aws:ec2:us-east-1::snapshot/*",
         "Condition":{
            "StringEquals":{
               "aws:ResourceTag/User":"${aws:username}"
            }
         }
      }
   ]
}
```

Kebijakan berikut memungkinkan pelanggan untuk membuat snapshot tetapi menolak tindakan jika snapshot yang dibuat memiliki kunci tanda `value=stack`.

```
{
   "Version":"2012-10-17",
   "Statement": [
      {
         "Effect":"Allow",
         "Action":[
            "ec2:CreateSnapshot",
            "ec2:CreateTags"
         ],
         "Resource":"*"
      },
      {
```

```
         "Effect":"Deny",
         "Action":"ec2:CreateSnapshot",
         "Resource":"arn:aws:ec2:us-east-1::snapshot/*",
         "Condition":{
            "ForAnyValue:StringEquals":{
               "aws:TagKeys":"stack"
            }
         }
      }
   ]
}
```

Kebijakan berikut memungkinkan pelanggan untuk membuat beberapa snapshot tetapi menolak tindakan jika snapshot yang dibuat tersebut memiliki kunci tanda `value=stack`.

```
{
   "Version":"2012-10-17",
   "Statement": [
      {
         "Effect":"Allow",
         "Action":[
            "ec2:CreateSnapshots",
            "ec2:CreateTags"
         ],
         "Resource":"*"
      },
      {
         "Effect":"Deny",
         "Action":"ec2:CreateSnapshots",
         "Resource":"arn:aws:ec2:us-east-1::snapshot/*",
         "Condition":{
            "ForAnyValue:StringEquals":{
               "aws:TagKeys":"stack"
            }
         }
      }
   ]
}
```

Kebijakan berikut memungkinkan Anda untuk menggabungkan beberapa tindakan ke dalam satu kebijakan. Anda hanya dapat membuat snapshot (dalam konteks `CreateSnapshots`) saat snapshot tersebut dibuat di Wilayah `us-east-1`. Anda hanya dapat membuat beberapa snapshot (dalam

konteks `CreateSnapshots`) ketika snapshot-snapshot tersebut sedang dibuat di Wilayah `us-east-1` dan ketika tipe instans-nya adalah `t2*`.

```
{
   "Version":"2012-10-17",
   "Statement": [
          {
         "Effect":"Allow",
         "Action":[
            "ec2:CreateSnapshots",
            "ec2:CreateSnapshot",
            "ec2:CreateTags"
         ],
         "Resource": [
            "arn:aws:ec2:*:*:instance/*",
            "arn:aws:ec2:*:*:snapshot/*",
            "arn:aws:ec2:*:*:volume/*"
         ],
         "Condition":{
            "StringEqualsIgnoreCase": {
              "ec2:Region": "us-east-1"
            },
            "StringLikeIfExists": {
              "ec2:InstanceType": ["t2.*"]
            }
         }
      }
   ]
}
```

Contoh: Menyalin beberapa snapshot

Izin tingkat sumber daya yang ditentukan untuk CopySnapshottindakan hanya berlaku untuk snapshot baru. izin tersebut tidak dapat ditentukan untuk snapshot sumber.

Kebijakan contoh berikut mengizinkan prinsipal utama untuk menyalin snapshot hanya jika snapshot baru dibuat dengan tanda kunci `purpose` dan nilai tanda `production` (`purpose=production`).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
```

```
            "Sid": "AllowCopySnapshotWithTags",
            "Effect": "Allow",
            "Action": "ec2:CopySnapshot",
            "Resource": "arn:aws:ec2:*:account-id:snapshot/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/purpose": "production"
                }
            }
        }
    ]
}
```

Contoh: Memodifikasi pengaturan izin untuk snapshot

Kebijakan berikut memungkinkan modifikasi snapshot hanya jika snapshot ditandai dengan`User:username`, di mana *nama pengguna adalah nama pengguna* AWS akun pelanggan. Permintaan akan gagal jika syarat ini tidak dipenuhi.

```
{
   "Version":"2012-10-17",
   "Statement": [
           {
         "Effect":"Allow",
         "Action":"ec2:ModifySnapshotAttribute",
         "Resource":"arn:aws:ec2:us-east-1::snapshot/*",
         "Condition":{
            "StringEquals":{
               "aws:ResourceTag/user-name":"${aws:username}"
            }
         }
      }
   ]
}
```

Memecahkan masalah identitas dan akses Amazon EBS

Gunakan informasi berikut untuk membantu Anda mendiagnosis dan memperbaiki masalah umum yang mungkin Anda temui saat bekerja dengan Amazon EBS dan IAM.

Masalah

- Saya tidak berwenang untuk melakukan tindakan di Amazon EBS
- Saya tidak berwenang untuk melakukan iam: PassRole
- Saya ingin mengizinkan orang di luar saya Akun AWS untuk mengakses sumber daya Amazon EBS saya

Saya tidak berwenang untuk melakukan tindakan di Amazon EBS

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 kredensial masuk.

Contoh kesalahan berikut terjadi ketika pengguna `mateojackson` IAM mencoba menggunakan konsol untuk melihat detail tentang volume tetapi tidak memiliki `ec2:DescribeVolumes` izin.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 ec2:DescribeVolumes on resource: volume-id
```

Dalam hal ini, Mateo meminta AWS administratornya untuk mengizinkannya menggambarkan volume.

Saya tidak berwenang untuk melakukan iam: PassRole

Jika Anda menerima kesalahan bahwa Anda tidak diizinkan untuk melakukan `iam:PassRole` tindakan, kebijakan Anda harus diperbarui agar Anda dapat meneruskan peran ke Amazon EBS.

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 Amazon EBS. 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 Amazon EBS saya

Anda dapat membuat peran yang dapat digunakan pengguna di akun lain atau pengguna 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 (ACL), Anda dapat menggunakan kebijakan tersebut untuk memberi pengguna akses ke sumber daya Anda.

Untuk mempelajari selengkapnya, periksa hal berikut:

- Untuk mengetahui apakah Amazon EBS mendukung fitur-fitur ini, lihatBagaimana Amazon Elastic Block Store 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 Memberikan akses kepada pengguna eksternal yang sah (federasi identitas) dalam Panduan Pengguna IAM.
- Untuk mempelajari perbedaan antara penggunaan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat Perbedaan antara peran IAM dan kebijakan berbasis sumber daya di Panduan Pengguna IAM.

Validasi kepatuhan untuk Amazon Elastic Block Store

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:

- Panduan Memulai Cepat Keamanan dan Kepatuhan — Panduan penerapan ini membahas pertimbangan arsitektur dan memberikan langkah-langkah untuk menerapkan lingkungan dasar AWS yang berfokus pada keamanan dan kepatuhan.
- Arsitektur untuk Keamanan dan Kepatuhan HIPAA di Amazon Web Services — Whitepaper ini menjelaskan bagaimana perusahaan dapat menggunakan AWS untuk membuat aplikasi yang memenuhi syarat HIPAA.

> **Note**
>
> Tidak semua memenuhi Layanan AWS syarat HIPAA. Untuk informasi selengkapnya, lihat Referensi Layanan yang Memenuhi 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 di Amazon Elastic Block Store

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, Amazon EBS menawarkan beberapa fitur untuk membantu mendukung ketahanan data dan kebutuhan pencadangan Anda.

- Menerapkan otomatisasi snapshot EBS menggunakan Amazon Data Lifecycle Manager
- Menyalin snapshot EBS di seluruh Wilayah

Memantau Amazon Elastic Block Store

Pemantauan merupakan bagian penting dalam menjaga keandalan, ketersediaan, dan kinerja Amazon Elastic Block Store dan AWS solusi Anda yang lain. AWS menyediakan alat pemantauan berikut untuk menonton Amazon EBS, melaporkan ketika ada sesuatu yang salah, dan mengambil tindakan otomatis bila perlu:

- AWS CloudTrailmenangkap panggilan API dan peristiwa terkait yang dibuat oleh atau atas nama Anda Akun AWS dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Anda dapat mengidentifikasi pengguna dan akun mana yang dipanggil AWS, alamat IP sumber dari mana panggilan dilakukan, dan kapan panggilan terjadi. Untuk informasi selengkapnya, silakan lihat Panduan Pengguna AWS CloudTrail.
- Amazon CloudWatch memantau AWS sumber daya Anda dan aplikasi yang Anda jalankan AWS secara real time. Anda dapat mengumpulkan dan melacak metrik, membuat dasbor yang disesuaikan, dan mengatur alarm yang memberi tahu Anda atau mengambil tindakan saat metrik tertentu mencapai ambang batas yang ditentukan. Misalnya, Anda dapat CloudWatch melacak penggunaan CPU atau metrik lain dari instans Amazon EC2 Anda dan secara otomatis meluncurkan instans baru bila diperlukan. Untuk informasi selengkapnya, lihat Panduan CloudWatch Pengguna Amazon.
- Amazon EventBridge dapat digunakan untuk mengotomatiskan AWS layanan Anda dan merespons secara otomatis peristiwa sistem, seperti masalah ketersediaan aplikasi atau perubahan sumber daya. Acara dari AWS layanan dikirim ke EventBridge dalam waktu dekat. Anda dapat menuliskan aturan sederhana untuk menunjukkan peristiwa mana yang sesuai kepentingan Anda, dan tindakan otomatis mana yang diambil ketika suatu peristiwa sesuai dengan suatu aturan. Untuk informasi selengkapnya, lihat Panduan EventBridge Pengguna Amazon.

Topik

AWS CloudTrail untuk Amazon EBS

Amazon Elastic Block Store (Amazon EBS) terintegrasi AWS CloudTrail dengan, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan di Amazon EBS. CloudTrail menangkap semua panggilan API untuk Amazon EBS sebagai peristiwa. Panggilan yang diambil termasuk panggilan dari konsol Amazon EBS dan panggilan kode ke operasi Amazon EBS API. Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail acara secara berkelanjutan ke bucket Amazon S3, termasuk acara untuk Amazon EBS. 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 Amazon EBS, alamat IP dari mana permintaan itu dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan.

Untuk mempelajari selengkapnya CloudTrail, lihat Panduan AWS CloudTrail Pengguna.

Informasi Amazon EBS di CloudTrail

CloudTrail diaktifkan pada Akun AWS saat Anda membuat akun. Ketika aktivitas terjadi di Amazon EBS, aktivitas tersebut direkam dalam suatu CloudTrail peristiwa bersama dengan peristiwa AWS layanan lainnya dalam riwayat Acara. Anda dapat melihat, mencari, dan mengunduh peristiwa terbaru di Akun AWS Anda. Untuk informasi selengkapnya, lihat Melihat peristiwa dengan Riwayat CloudTrail acara.

Untuk catatan acara yang sedang berlangsung di Anda Akun AWS, termasuk acara untuk Amazon EBS, 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 Amazon EBS API dicatat oleh CloudTrail. Misalnya, panggilan ke CreateVolume, DeleteVolumedan CreateSnapshottindakan menghasilkan entri dalam file CloudTrail log.

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

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

Untuk informasi selengkapnya, lihat Elemen userIdentity CloudTrail .

Memahami entri file log Amazon EBS

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 CreateVolumetindakan.

```
{
    "eventVersion": "1.09",
    "userIdentity": {
        "type": "Root",
        "principalId": "AROAJABCHBVMHREXAMPLE:root",
        "arn": "123456789012",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE"
    },
    "eventTime": "2024-02-08T08:02:21Z",
    "eventSource": "ec2.amazonaws.com",
    "eventName": "CreateVolume",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "12.12.123.123",
    "userAgent": "aws-cli/1.10.10 Python/2.7.9 Windows/7botocore/1.4.1",
```

```
    "requestParameters": {
        "size": "100",
        "zone": "us-east-1a",
        "volumeType": "gp3",
        "iops": "3000",
        "encrypted": true,
        "masterEncryptionKeyId": "arn:aws:kms:us-east-1:123456789012:key/12345678-
a202-4b72-8030-example23456",
        "throughput": "125",
        "clientToken": "12345678-2427-4336-a555-e8607example"
    },
    "responseElements": {
        "requestId": "12345678-4229-4cfd-9cb1-0b094example",
        "volumeId": "vol-01234567890abcdef",
        "size": "100",
        "zone": "us-east-1a",
        "status": "creating",
        "createTime": 1707379341000,
        "volumeType": "gp3",
        "iops": 3000,
        "encrypted": true,
        "masterEncryptionKeyId": "arn:aws:kms:us-east-1:123456789012:key/12345678-
a202-4b72-8030-example23456",
        "tagSet": {},
        "multiAttachEnabled": false,
        "throughput": 125
    },
    "requestID": "12345678-4229-4cfd-9cb1-0b094example",
    "eventID": "12345678-4b33-4c18-90a1-76d4bexample",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.3",
        "cipherSuite": "TLS_AES_128_GCM_SHA256",
        "clientProvidedHostHeader": "ec2.us-east-1.amazonaws.com"
    },
    "sessionCredentialFromConsole": "true"
}
```

CloudWatch Metrik Amazon untuk Amazon EBS

CloudWatch Metrik Amazon adalah data statistik yang dapat Anda gunakan untuk melihat, menganalisis, dan menyetel alarm tentang perilaku operasional volume Anda.

Data tersedia secara otomatis dalam periode 1 menit tanpa biaya.

Ketika Anda mendapatkan data dari CloudWatch, Anda dapat menyertakan parameter `Period` permintaan untuk menentukan granularitas data yang dikembalikan. Ini berbeda dengan periode yang kami gunakan saat mengumpulkan data (periode 1 menit). Kami menyarankan Anda untuk menentukan periode dalam permintaan Anda yang sama dengan atau lebih besar dari periode pengumpulan untuk memastikan bahwa data yang dikembalikan valid.

Anda bisa mendapatkan data menggunakan CloudWatch API atau konsol Amazon EC2. Konsol mengambil data mentah dari CloudWatch API dan menampilkan serangkaian grafik berdasarkan data. Bergantung pada kebutuhan Anda, Anda mungkin lebih memilih menggunakan data dari API atau grafik di konsol.

Topik

Metrik untuk volume Amazon EBS

Namespace `AWS/EBS` mencakup metrik berikut untuk volume EBS yang dilampirkan ke semua tipe instans. Semua jenis volume Amazon EBS secara otomatis mengirim metrik 1 menit ke CloudWatch, tetapi hanya jika volume dilampirkan ke instans.

Untuk mendapatkan informasi tentang ruang disk yang tersedia dari sistem operasi di instans, lihat Lihat ruang disk kosong.

> **Note**
>
> Beberapa metrik memiliki perbedaan pada instans yang dibuat pada Sistem Nitro. Untuk daftar jenis instance ini, lihat Instans yang dibangun di atas Sistem Nitro.

Metrik	Deskripsi	Unit	Dimensi	Statistik yang bermakna
VolumeReadBytes	Memberikan informasi tentang operasi yang dibaca dalam periode waktu tertentu. • Statistik Sum melaporkan jumlah total byte yang ditransfer selama periode tersebut. • Statistik Average melaporkan ukuran rata-rata setiap operasi baca selama periode tersebut, kecuali volume yang dilampirkan ke instans Nitro, di mana rata-rata tersebut merepresentasikan rata-rata selama periode tertentu. • Statistik SampleCount melaporkan total jumlah operasi baca selama periode tersebut, kecuali volume yang dilampirkan ke instans berbasis Nitro, di mana jumlah sampel mewakili tersebut merepresentasikan jumlah titik data yang digunakan	Byte	VolumeId	• Average • Sum • SampleCount • Minimum | Maximum — hanya untuk volume yang dilampirkan ke instans berbasis Nitro

Metrik	Deskripsi	Unit	Dimensi	Statistik yang bermakna
	dalam perhitungan statistik. **Note** Untuk instans Xen, data dilaporkan hanya ketika ada aktivitas baca pada volume.			

Metrik	Deskripsi	Unit	Dimensi	Statistik yang bermakna
`VolumeWriteBytes`	Memberikan informasi tentang operasi baca dalam jangka waktu tertentu • Statistik `Sum` melaporkan jumlah total bita yang ditransfer selama periode tersebut. • Statistik `Average` melaporkan ukuran rata-rata setiap operasi tulis selama periode tersebut, kecuali pada volume yang dipasang ke instans berbasis Nitro, di mana rata-rata mewakili rata-rata selama periode tertentu. • Statistik `SampleCount` melaporkan jumlah total operasiselama periode tersebut, kecuali pada volume yang dipasang ke instans berbasis, di mana jumlah sampel mewakili jumlah titik data yang digunakan dalam perhitungan statistik.	Byte	`VolumeId`	• `Average` • `Sum` • `SampleCount` • `Minimum` | `Maximum` — hanya untuk volume yang dilampirkan ke instans berbasis Nitro

Metrik	Deskripsi	Unit	Dimensi	Statistik yang bermakna
	Note Untuk instans Xen, data dilaporkan hanya ketika ada aktivitas baca pada volume.			
`VolumeReadOps`	Total jumlah operasi baca dalam periode waktu tertentu. Operasi baca dihitung setelah selesai. Untuk menghitung operasi baca rata-rata per detik (baca IOPS) untuk periode, bagi total operasi baca dalam periode tersebut dengan jumlah detik dalam periode tersebut.	Hitung	`VolumeId`	• `Average` • `Sum` • `Minimum` | `Maximum` — hanya untuk volume yang dilampirkan ke instans berbasis Nitro

Metrik	Deskripsi	Unit	Dimensi	Statistik yang bermakna
`VolumeWriteOps`	Total jumlah operasida lam periode waktu tertentu. Operasi tulis dihitung pada penyelesa ian. Untuk menghitung rata-rata operasi tulis per detik (IOPS tulis) untuk periode tersebut, bagi total operasi tulis dalam periode dengan jumlah detik dalam periode tersebut.	Hitung	`VolumeId`	• `Average` • `Sum` • `Minimum` | `Maximum` — hanya untuk volume yang dilampirk an ke instans berbasis Nitro

Metrik	Deskripsi	Unit	Dimensi	Statistik yang bermakna
`VolumeTotalReadTime`	**Note** Tidak didukung dengan volume dengan Multi-Lampiran aktif. Untuk instans Xen, data dilaporkan hanya ketika ada aktivitas baca pada volume. Total jumlah detik yang dihabiskan oleh semua operasi baca yang selesai dalam jangka waktu tertentu. Jika beberapa permintaan dikirimkan pada waktu yang sama, total ini dapat lebih besar dari lama periode. Misalnya, selama 1 menit (60 detik): jika 150 operasi selesai selama periode tersebut, dan setiap operasi memerlukan 1 detik, nilainya adalah 150 detik.	Detik	`VolumeId`	• `Average` — tidak relevan untuk volume yang dilampirkan ke instans berbasis Nitro • `Sum` • `Minimum` | `Maximum` — hanya untuk volume yang dilampirkan ke instans berbasis Nitro

Metrik	Deskripsi	Unit	Dimensi	Statistik yang bermakna
`VolumeTotalWriteTime`	**Note** Tidak didukung dengan volume dengan Multi-Lampiran aktif. Untuk instans Xen, data dilaporkan hanya ketika ada aktivitas baca pada volume. Total jumlah detik yang dihabiskan oleh semua operasiyang selesai dalam periode waktu tertentu. Jika beberapa permintaan dikirimkan pada waktu yang sama, total ini dapat lebih besar dari lama periode. Misalnya, selama 1 menit (60 detik): jika 150 operasi selesai selama periode tersebut, dan setiap operasi memerlukan 1 detik, nilainya adalah 150 detik.	Detik	`VolumeId`	• `Average` — tidak relevan untuk volume yang dilampirkan ke instans berbasis Nitro • `Sum` • `Minimum` | `Maximum` — hanya untuk volume yang dilampirkan ke instans berbasis Nitro

Metrik	Deskripsi	Unit	Dimensi	Statistik yang bermakna
`VolumeIdleTime`	**Note** Tidak didukung dengan volume dengan Multi-Lampiran aktif. Total jumlah detik dalam periode waktu tertentu ketika tidak ada operasi baca atauyang .	Detik	`VolumeId`	• `Average` — tidak relevan untuk volume yang dilampirkan ke instans berbasis Nitro • `Sum` • `Minimum` | `Maximum` — hanya untuk volume yang dilampirkan ke instans berbasis Nitro

Metrik	Deskripsi	Unit	Dimensi	Statistik yang bermakna
`VolumeQueueLength`	Jumlah permintaan operasi baca danharus diselesaikan dalam jangka waktu tertentu.	Hitung	`VolumeId`	• `Average` • `Sum` — tidak relevan untuk volume yang dilampirkan ke instans Nitro • `Minimum` | `Maximum` — hanya untuk volume yang dilampirkan ke instans Nitro

Metrik	Deskripsi	Unit	Dimensi	Statistik yang bermakna
`VolumeStalledIOCheck`	**Note** Hanya untuk contoh Nitro. Tidak dipublikasikan untuk volume yang dilampirkan ke Amazon ECS dan AWS Fargate tugas. Melaporkan apakah volume telah lulus atau gagal pemeriksaan IO yang terhenti pada menit terakhir.Metrik ini dapat berupa `0` (lulus) atau 1 (gagal) .Untuk informasi selengkapnya, lihat. Pantau karakteristik I/O menggunakan CloudWatch	Hitung	`VolumeId` | `InstanceId`	• Jumlah • Rata-rata • Minimum • Maksimum

Metrik	Deskripsi	Unit	Dimensi	Statistik yang bermakna
`VolumeThroughputPercentage`	**Note** Khusus volume SSD IOPS yang tersedia. Tidak didukung dengan volume dengan Multi-Lampiran aktif. Persentase operasi I/O per detik (IOPS) yang diberikan dari total IOPS terprovisi untuk volume Amazon EBS. Volume SSD IOPS yang Tersedia memberikan performa yang telah disediakan 99,9 persen. Selama , jika tidak ada permintaan I/O tertunda lainnya dalam satu menit, nilai metrik akan menjadi 100 persen. Selain itu, performa I/O volume dapat terdegradasi sementara karena tindakan yang telah Anda ambil (misalnya, membuat snapshot volume selama	Persen	`VolumeId`	• `Average` • `Minimum` | `Maximum`

Metrik	Deskripsi	Unit	Dimensi	Statistik yang bermakna
	penggunaan puncak, menjalankan volume pada instans EBS yang dioptimalkan, atau mengakses data pada volume untuk pertama kali).			
`VolumeConsumedReadWriteOps`	**Note** Khusus volume SSD IOPS yang tersedia. Jumlah total operasi baca dan tulis (dinormalkan menjadi 256K unit kapasitas) yang digunakan dalam periode waktu tertentu. Operasi I/O yang lebih kecil dari 256K masing-masing dihitung sebagai 1 IOPS konsumsi. Operasi I/O yang lebih besar dari 256K dihitung dalam 256K unit kapasitas. Misalnya, I/O 1024K akan dihitung sebagai 4 IOPS konsumsi.	Hitung	`VolumeId`	• `Average` • `Sum` • `Minimum` | `Maximum`

Metrik	Deskripsi	Unit	Dimensi	Statistik yang bermakna
`BurstBalance`	**Note** `gp2`,`st1`, dan `sc1` volume saja. Memberikan informasi tentang persentase kredit I/O (untuk `gp2`) atau kredit throughput (untuk `st1` dan `sc1`) yang tersisa dalam bucket lonjakan. Data dilaporkan CloudWatch hanya ketika volume aktif. Jika volume tidak dipasang, tidak ada data yang dilaporkan. Jika performa dasar volume melebihi performa lonjakan maksimum, kredit tidak pernah dihabiskan. Jika volumenya terpasang ke suatu instans yang dibangun di Sistem Nitro, saldo lonjakan tidak dilaporkan. Untuk instans lainnya, saldo lonjakan yang dilaporkan adalah 100%. Untuk informasi selengkapnya,	Persen	`VolumeId`	• `Average` • `Sum` — tidak relevan untuk volume yang dilampirkan ke instans Nitro. • `Minimum` | `Maximum`

Metrik	Deskripsi	Unit	Dimensi	Statistik yang bermakna
	lihat Performa volume gp2.			

Metrik untuk instans Nitro

Namespace `AWS/EC2` mencakup metrik Amazon EBS tambahan untuk volume yang dipasangkan ke instans berbasis Nitro yang bukan merupakan instans bare metal.

Metrik	Deskripsi	Unit	Statistik yang bermakna
`EBSReadOps`	Operasi baca yang diselesaikan dari semua volume Amazon EBS yang dilampirkan ke instans dalam periode waktu yang ditentukan. Untuk menghitung rata-rata operasi I/O per detik dari pembacaan (IOPS Baca) selama periode tersebut, bagilah total operasi dalam periode dengan jumlah detik pada periode tersebut. Jika menggunakan pemantauan dasar (5 menit), Anda dapat membagi jumlah ini dengan 300 untuk menghitung nilai IOPS Baca. Jika Anda memiliki pemantauan terperinci (1 menit), bagilah dengan 60. Anda juga dapat menggunakan fungsi matematika CloudWatch metrik `DIFF_TIME` untuk menemukan operasi per detik. Misalnya, jika Anda telah membuat grafik `EBSReadOps` CloudWatch sebagai`m1`, rumus matematika metrik `m1/(DIFF_TIME(m1))` mengembalikan metrik dalam operasi/detik. Untuk informasi selengkapnya tentang `DIFF_TIME` dan fungsi matematika metrik lainnya, lihat Menggunak	Hitung	• Jumlah • Rata-rata • Minimum • Maksimum

Metrik	Deskripsi	Unit	Statistik yang bermakna
	an matematika metrik di Panduan CloudWatch Pengguna Amazon.		
`EBSWriteOps`	Operasi tulis yang diselesaikan ke semua volume EBS yang dilampirkan pada instans tersebut dalam periode waktu yang ditentukan. Untuk menghitung rata-rata operasi I/O per detik dari penulisan (IOPS Tulis) selama periode tersebut, bagilah total operasi dalam periode dengan jumlah detik pada periode tersebut. Jika menggunakan pemantauan dasar (5 menit), Anda dapat membagi angka ini dengan 300 untuk menghitung nilai IOPS Tulis. Jika Anda memiliki pemantauan terperinci (1 menit), bagilah dengan 60. Anda juga dapat menggunakan fungsi matematika CloudWatch metrik `DIFF_TIME` untuk menemukan operasi per detik. Misalnya, jika Anda telah membuat grafik `EBSWriteOps` CloudWatch sebagai`m1`, rumus matematika metrik `m1/(DIFF_TIME(m1))` mengembalikan metrik dalam operasi/detik. Untuk informasi selengkapnya tentang `DIFF_TIME` dan fungsi matematika metrik lainnya, lihat Menggunakan matematika metrik di Panduan CloudWatch Pengguna Amazon.	Hitung	• Jumlah • Rata-rata • Minimum • Maksimum

Metrik	Deskripsi	Unit	Statistik yang bermakna
`EBSReadBytes`	Bita yang dibaca dari semua volume EBS yang dilampirkan ke instans dalam jangka waktu tertentu. Jumlah yang dilaporkan adalah jumlah bita baca selama periode tersebut. Jika menggunakan pemantauan dasar (5 menit), Anda dapat membagi angka ini dengan 300 untuk menemukan Bita/detik dari Pembacaan. Jika Anda memiliki pemantauan terperinci (1 menit), bagilah dengan 60. Anda juga dapat menggunakan fungsi matematika CloudWatch metrik `DIFF_TIME` untuk menemukan byte per detik. Misalnya, jika Anda telah membuat grafik `EBSReadBytes` CloudWatch sebagai`m1`, rumus matematika metrik `m1/(DIFF_TIME(m1))` mengembalikan metrik dalam byte/detik. Untuk informasi selengkapnya tentang `DIFF_TIME` dan fungsi matematika metrik lainnya, lihat Menggunakan matematika metrik di Panduan CloudWatch Pengguna Amazon.	Byte	• Jumlah • Rata-rata • Minimum • Maksimum

Metrik	Deskripsi	Unit	Statistik yang bermakna
`EBSWriteBytes`	Bita yang ditulis ke semua volume EBS yang dilampirkan ke instans dalam jangka waktu tertentu. Jumlah yang dilaporkan adalah jumlah bita tulis selama periode tersebut. Jika menggunakan pemantauan dasar (5 menit), Anda dapat membagi jumlah ini dengan 300 untuk menemukan Bita/detik dari Penulisan. Jika Anda memiliki pemantauan terperinci (1 menit), bagilah dengan 60. Anda juga dapat menggunakan fungsi matematika CloudWatch metrik `DIFF_TIME` untuk menemukan byte per detik. Misalnya, jika Anda telah membuat grafik `EBSWriteBytes` CloudWatch sebagai`m1`, rumus matematika metrik `m1/(DIFF_TIME(m1))` mengembalikan metrik dalam byte/detik. Untuk informasi selengkapnya tentang `DIFF_TIME` dan fungsi matematika metrik lainnya, lihat Menggunakan matematika metrik di Panduan CloudWatch Pengguna Amazon.	Byte	• Jumlah • Rata-rata • Minimum • Maksimum

Metrik	Deskripsi	Unit	Statistik yang bermakna
`EBSIOBalance%`	Memberikan informasi tentang persentase kredit I/O yang tersisa di bucket lonjakan. Metrik ini hanya tersedia untuk pemantauan dasar. Metrik ini hanya tersedia untuk beberapa ukuran instans `*.4xlarge` dan yang lebih kecil, yang melonjak hingga performa maksimum hanya selama 30 menit setidaknya setiap 24 jam sekali. Untuk informasi selengkapnya, lihat EBS yang dioptimalkan secara default. Statistik `Sum` tidak berlaku untuk metrik ini.	Persen	• Minimum • Maksimum
`EBSByteBalance%`	Memberikan informasi tentang persentase kredit throughput yang tersisa di bucket lonjakan. Metrik ini hanya tersedia untuk pemantauan dasar. Metrik ini hanya tersedia untuk beberapa ukuran instans `*.4xlarge` dan yang lebih kecil, yang melonjak hingga performa maksimum hanya selama 30 menit setidaknya setiap 24 jam sekali. Untuk informasi selengkapnya, lihat EBS yang dioptimalkan secara default. Statistik `Sum` tidak berlaku untuk metrik ini.	Persen	• Minimum • Maksimum

Metrik untuk pemulihan snapshot cepat

Namespace `AWS/EBS` menyertakan metrik untuk pemulihan snapshot cepat.

Metrik	Deskripsi	Unit	Dimensi	Statistik yang bermakna
FastSnapshotRestoreCreditsBucketSize	Jumlah maksimal volume akan membuatyang dapat diakumulasi. Metrik ini dilaporkan per snapshot per Availability Zone.	Hitung	SnapshotId | AvailabilityZone	• Average • Minimum | Maximum **Note** Statistik yang paling bermakna adalah Average. Hasil untuk statistik Minimum dan Maximum adalah sama dengan Average dan dapat digunakan.
FastSnapshotRestoreCreditsBalance	Jumlah volume membuattersedia. Metrik ini dilaporkan per snapshot per Zona Ketersediaan.	Hitung	SnapshotId | AvailabilityZone	• Average • Minimum | Maximum **Note** Statistik yang paling bermakna adalah Average. Hasil untuk statistik Minimum dan Maximum adalah sama dengan Average dan dapat digunakan.

Grafik konsol Amazon EC2

Setelah Anda membuat volume, Anda dapat melihat grafikvolume di konsol Amazon EC2. Pilih volume pada Volume di konsol dan pilih Pemantauan. Tabel berikut mencantumkan grafik yang ditampilkan. Kolom di sebelah kanan menjelaskan bagaimana metrik data mentah dari CloudWatch API digunakan untuk menghasilkan setiap grafik. Periode untuk semua grafik adalah 5 menit.

Grafik	Deskripsi menggunakan metrik mentah
Throughput baca (KiB/dtk)	`Sum(VolumeReadBytes) / Period / 1024`
Throughput tulis (KiB/dtk)	`Sum(VolumeWriteBytes) / Period / 1024`
Baca operasi (Ops/dtk)	`Sum(VolumeReadOps) / Period`
Operasi tulis (Ops/dtk)	`Sum(VolumeWriteOps) / Period`
Panjang antrean rata-rata (Operasi)	`Avg(VolumeQueueLength)`
Waktu yang dihabiskan untuk idle (%)	`Sum(VolumeIdleTime) / Period × 100`

Ukuran baca rata-rata (KiB/op)	`Avg(VolumeReadBytes) / 1024`
	
	Untuk contoh berbasis Nitro, rumus berikut memperoleh Ukuran Baca Rata-rata menggunakan CloudWatch Matematika Metrik:
	
	`(Sum(VolumeReadBytes) / Sum(VolumeReadOps)) / 1024`
	
	`VolumeReadOps` Metrik `VolumeReadBytes` dan tersedia di konsol EBS CloudWatch .

Ukuran tulis rata-rata (KiB/op)	`Avg(VolumeWriteBytes) / 1024` Untuk contoh berbasis Nitro, rumus berikut memperoleh Ukuran Tulis Rata-rata menggunakan CloudWatch Matematika Metrik:

Grafik	Deskripsi menggunakan metrik mentah
	`(Sum(VolumeWriteBytes) / Sum(VolumeWriteOps)) / 1024` `VolumeWriteOps` Metrik `VolumeWriteBytes` dan tersedia di konsol EBS CloudWatch .

Latensi baca rata-rata (ms/op)	`Avg(VolumeTotalReadTime) × 1000`
	
	Untuk contoh berbasis Nitro, rumus berikut memperoleh Latensi Baca Rata-rata menggunakan Matematika Metrik: CloudWatch
	
	`(Sum(VolumeTotalReadTime) / Sum(VolumeReadOps)) × 1000`
	
	`VolumeReadOps` Metrik `VolumeTotalReadTime` dan tersedia di konsol EBS CloudWatch .

Latensi tulis rata-rata (ms/op)	`Avg(VolumeTotalWriteTime) × 1000`
	
	Untuk contoh berbasis Nitro, rumus berikut memperoleh Latensi Tulis Rata-rata menggunakan Matematika Metrik: CloudWatch
	
	`(Sum(VolumeTotalWriteTime) / Sum(VolumeWriteOps)) * 1000`
	
	`VolumeWriteOps` Metrik `VolumeTotalWriteTime` dan tersedia di konsol EBS CloudWatch .

Untuk grafik latensi rata-rata dan grafik ukuran rata-rata, rata-ratanya dihitung terhadap jumlah total operasi (baca atau tulis, mana saja yang berlaku untuk grafik) yang selesai selama periode.

Amazon EventBridge untuk Amazon EBS

Amazon EBS mengirimkan acara ke Amazon EventBridge untuk tindakan yang dilakukan pada volume dan snapshot. Dengan EventBridge, Anda dapat menetapkan aturan yang memicu

tindakan terprogram dalam menanggapi peristiwa ini. Misalnya, Anda dapat membuat aturan yang mengirimkan notifikasi ke email Anda saat snapshot diaktifkan untuk pemulihan snapshot cepat.

Peristiwa di EventBridge direpresentasikan sebagai objek JSON. Kolom-kolom yang unik untuk peristiwa tersebut terdapat di bagian "detail" dari objek JSON. Bidang "peristiwa" berisi nama peristiwa. Bidang "hasil" berisi status selesai dari tindakan yang memicu peristiwa. Untuk informasi selengkapnya, lihat pola EventBridge acara Amazon di Panduan EventBridge Pengguna Amazon.

Untuk informasi selengkapnya, lihat Apa itu Amazon EventBridge? di Panduan EventBridge Pengguna Amazon.

Peristiwa

- Peristiwa volume EBS
- Peristiwa modifikasi volume EBS
- Peristiwa snapshot EBS
- Peristiwa Arsip Snapshots EBS
- Peristiwa pemulihan snapshot cepat EBS
- Menggunakan AWS Lambda untuk menangani EventBridge acara

Peristiwa volume EBS

Amazon EBS mengirimkan peristiwa ke EventBridge saat peristiwa volume berikut terjadi.

Peristiwa

- Buat volume (buatVolume)
- Hapus volume (hapusVolume)
- Pasang atau pasangkan ulang volume (attachVolume, reattachVolume)
- Lepaskan volume (detachVolume)

Buat volume (buatVolume)

`createVolume`Acara dikirim ke AWS akun Anda ketika tindakan untuk membuat volume selesai. Namun, peristiwa tersebut tidak disimpan, dicatat, atau diarsipkan. Peristiwa ini dapat membawa hasil `available` atau `failed`. Pembuatan akan gagal jika tidak valid disediakan, seperti AWS KMS key yang ditunjukkan pada contoh di bawah ini.

Data peristiwa

Daftar di bawah ini adalah contoh objek JSON yang dipancarkan oleh EBS untuk peristiwa `createVolume` yang berhasil.

```
{
   "version": "0",
   "id": "01234567-0123-0123-0123-012345678901",
   "detail-type": "EBS Volume Notification",
   "source": "aws.ec2",
   "account": "012345678901",
   "time": "yyyy-mm-ddThh:mm:ssZ",
   "region": "us-east-1",
   "resources": [
      "arn:aws:ec2:us-east-1:012345678901:volume/vol-01234567"
   ],
   "detail": {
      "result": "available",
      "cause": "",
      "event": "createVolume",
      "request-id": "01234567-0123-0123-0123-0123456789ab"
   }
}
```

Daftar di bawah ini adalah contoh objek JSON yang dipancarkan oleh EBS setelah peristiwa `createVolume` yang gagal. Penyebab kegagalan tersebut adalah kunci KMS yang dinonaktifkan.

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-0123456789ab",
  "detail-type": "EBS Volume Notification",
  "source": "aws.ec2",
  "account": "012345678901",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "sa-east-1",
  "resources": [
    "arn:aws:ec2:sa-east-1:0123456789ab:volume/vol-01234567",
  ],
  "detail": {
    "event": "createVolume",
    "result": "failed",
    "cause": "arn:aws:kms:sa-
east-1:0123456789ab:key/01234567-0123-0123-0123-0123456789ab is disabled.",
```

```
    "request-id": "01234567-0123-0123-0123-0123456789ab",
  }
}
```

Berikut adalah contoh objek JSON yang dipancarkan oleh EBS setelah peristiwa `createVolume` yang gagal. Penyebab kegagalan adalah impor utama KMS yang tertunda.

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-0123456789ab",
  "detail-type": "EBS Volume Notification",
  "source": "aws.ec2",
  "account": "012345678901",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "sa-east-1",
  "resources": [
    "arn:aws:ec2:sa-east-1:0123456789ab:volume/vol-01234567",
  ],
  "detail": {
    "event": "createVolume",
    "result": "failed",
    "cause": "arn:aws:kms:sa-
east-1:0123456789ab:key/01234567-0123-0123-0123-0123456789ab is pending import.",
    "request-id": "01234567-0123-0123-0123-0123456789ab",
  }
}
```

Hapus volume (hapusVolume)

`deleteVolume`Acara dikirim ke AWS akun Anda ketika tindakan untuk menghapus volume selesai. Namun, peristiwa tersebut tidak disimpan, dicatat, atau diarsipkan. Peristiwa ini memiliki hasil `deleted`. Jika penghapusan tidak selesai, peristiwa tidak akan pernah dikirim.

Data peristiwa

Daftar di bawah ini adalah contoh objek JSON yang dipancarkan oleh EBS untuk peristiwa `deleteVolume` yang berhasil.

```
{
   "version": "0",
   "id": "01234567-0123-0123-0123-012345678901",
```

```
   "detail-type": "EBS Volume Notification",
   "source": "aws.ec2",
   "account": "012345678901",
   "time": "yyyy-mm-ddThh:mm:ssZ",
   "region": "us-east-1",
   "resources": [
      "arn:aws:ec2:us-east-1:012345678901:volume/vol-01234567"
   ],
   "detail": {
      "result": "deleted",
      "cause": "",
      "event": "deleteVolume",
      "request-id": "01234567-0123-0123-0123-0123456789ab"
   }
}
```

Pasang atau pasangkan ulang volume (attachVolume, reattachVolume)

Peristiwa `attachVolume` atau `reattachVolume` dikirim ke akun AWS Anda jika volume gagal untuk memasang atau memasang kembali suatu instans. Namun, peristiwa tersebut tidak disimpan, dicatat, atau diarsipkan. Jika Anda menggunakan kunci KMS untuk mengenkripsi volume EBS dan kunci KMS tersebut menjadi tidak valid, EBS akan membuat catatan peristiwa jika kunci KMS tersebut kemudian digunakan untuk memasang atau memasang kembali instans, seperti yang ditunjukkan pada instans di bawah ini.

Data peristiwa

Daftar di bawah ini adalah contoh objek JSON yang dipancarkan oleh EBS setelah peristiwa `attachVolume` yang gagal. Penyebab kegagalan adalah penghapusan kunci KMS yang tertunda.

> **Note**
>
> AWS dapat mencoba untuk menyambung kembali ke volume setelah pemeliharaan server rutin.

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-0123456789ab",
  "detail-type": "EBS Volume Notification",
  "source": "aws.ec2",
```

```
  "account": "012345678901",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-east-1",
  "resources": [
  "arn:aws:ec2:us-east-1:0123456789ab:volume/vol-01234567",
  "arn:aws:kms:us-east-1:0123456789ab:key/01234567-0123-0123-0123-0123456789ab"
  ],
  "detail": {
    "event": "attachVolume",
    "result": "failed",
    "cause": "arn:aws:kms:us-
east-1:0123456789ab:key/01234567-0123-0123-0123-0123456789ab is pending deletion.",
    "request-id": ""
  }
}
```

Daftar di bawah ini adalah contoh objek JSON yang dipancarkan oleh EBS setelah peristiwa `reattachVolume` yang gagal. Penyebab kegagalan adalah penghapusan kunci KMS yang tertunda.

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-0123456789ab",
  "detail-type": "EBS Volume Notification",
  "source": "aws.ec2",
  "account": "012345678901",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-east-1",
  "resources": [
  "arn:aws:ec2:us-east-1:0123456789ab:volume/vol-01234567",
  "arn:aws:kms:us-east-1:0123456789ab:key/01234567-0123-0123-0123-0123456789ab"
  ],
  "detail": {
    "event": "reattachVolume",
    "result": "failed",
    "cause": "arn:aws:kms:us-
east-1:0123456789ab:key/01234567-0123-0123-0123-0123456789ab is pending deletion.",
    "request-id": ""
  }
}
```

Lepaskan volume (detachVolume)

`detachVolume`Acara dikirim ke AWS akun Anda saat volume terlepas dari instans Amazon EC2.

Data peristiwa

Berikut ini adalah contoh `detachVolume` acara yang sukses.


```
{
  "version":"0",
  "id":"2ec37298-1234-e436-70fc-c96b1example",
  "detail-type":"AWS API Call via CloudTrail",
  "source":"aws.ec2",
  "account":"123456789012",
  "time":"2024-03-18T16:35:52Z",
  "region":"us-east-1",
  "resources":[],
  "detail":
  {
    "eventVersion":"1.09",
    "userIdentity":
    {
      "type":"IAMUser",
      "principalId":"AIDAJT12345SQ2EXAMPLE",
      "arn":"arn:aws:iam::123456789012:user/administrator",
      "accountId":"123456789012",
      "accessKeyId":"AKIAJ67890A6EXAMPLE",
      "userName":"administrator"
    },
    "eventTime":"2024-03-18T16:35:52Z",
    "eventSource":"ec2.amazonaws.com",
    "eventName":"DetachVolume",
    "awsRegion":"us-east-1",
    "sourceIPAddress":"12.12.123.12",
    "userAgent":"aws-cli/2.7.12 Python/3.9.11 Windows/10 exe/AMD64 prompt/off command/
ec2.detach-volume",
    "requestParameters":
    {
      "volumeId":"vol-072577c46bexample",
      "force":false
    },
    "responseElements":
    {
      "requestId":"1234513a-6292-49ea-83f8-85e95example",
      "volumeId":"vol-072577c46bexample",
      "instanceId":"i-0217f7eb3dexample",
      "device":"/dev/sdb",
      "status":"detaching",
```

```
      "attachTime":1710776815000
    },
    "requestID":"1234513a-6292-49ea-83f8-85e95example",
    "eventID":"1234551d-a15a-43eb-9e69-c983aexample",
    "readOnly":false,
    "eventType":"AwsApiCall",
    "managementEvent":true,
    "recipientAccountId":"123456789012",
    "eventCategory":"Management",
    "tlsDetails":
    {
      "tlsVersion":"TLSv1.3",
      "cipherSuite":"TLS_AES_128_GCM_SHA256",
      "clientProvidedHostHeader":"ec2.us-east-1.amazonaws.com"
    }
  }
}
```

Peristiwa modifikasi volume EBS

Amazon EBS mengirimkan `modifyVolume` peristiwa ke EventBridge saat volume diubah. Namun, peristiwa tersebut tidak disimpan, dicatat, atau diarsipkan.

```
{
   "version": "0",
   "id": "01234567-0123-0123-0123-012345678901",
   "detail-type": "EBS Volume Notification",
   "source": "aws.ec2",
   "account": "012345678901",
   "time": "yyyy-mm-ddThh:mm:ssZ",
   "region": "us-east-1",
   "resources": [
      "arn:aws:ec2:us-east-1:012345678901:volume/vol-03a55cf56513fa1b6"
   ],
   "detail": {
      "result": "optimizing",
      "cause": "",
      "event": "modifyVolume",
      "request-id": "01234567-0123-0123-0123-0123456789ab"
   }
}
```

Peristiwa snapshot EBS

Amazon EBS mengirimkan peristiwa ke EventBridge saat peristiwa volume berikut terjadi.

Peristiwa

- Membuat snapshot (membuatSnapshot)
- Membuat snapshot (membuatSnapshots)
- Salin snapshot (copySnapshot)
- Bagikan snapshot (shareSnapshot)

Membuat snapshot (membuatSnapshot)

`createSnapshot`Acara dikirim ke AWS akun Anda saat tindakan untuk membuat snapshot selesai. Namun, peristiwa tersebut tidak disimpan, dicatat, atau diarsipkan. Peristiwa ini dapat membawa hasil `succeeded` atau `failed`.

Data peristiwa

Daftar di bawah ini adalah contoh objek JSON yang dipancarkan oleh EBS untuk peristiwa `createSnapshot` yang berhasil. Di `detail` bagian, `source` bidang berisi ARN volume sumber. Kolom `startTime` dan `endTime` mengindikasikan kapan pembuatan snapshot dimulai dan diselesaikan.

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "EBS Snapshot Notification",
  "source": "aws.ec2",
  "account": "012345678901",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-east-1",
  "resources": [
     "arn:aws:ec2::us-west-2:snapshot/snap-01234567"
  ],
  "detail": {
    "event": "createSnapshot",
    "result": "succeeded",
    "cause": "",
    "request-id": "",
    "snapshot_id": "arn:aws:ec2::us-west-2:snapshot/snap-01234567",
```

```
    "source": "arn:aws:ec2::us-west-2:volume/vol-01234567",
    "startTime": "yyyy-mm-ddThh:mm:ssZ",
    "endTime": "yyyy-mm-ddThh:mm:ssZ"  }
}
```

Membuat snapshot (membuatSnapshots)

`createSnapshots`Acara dikirim ke AWS akun Anda saat tindakan untuk membuat snapshot multi-volume selesai. Peristiwa ini dapat membawa hasil `succeeded` atau `failed`.

Data peristiwa

Daftar di bawah ini adalah contoh objek JSON yang dipancarkan oleh EBS untuk peristiwa `createSnapshots` yang berhasil. Di bagian `detail`, bidang `source` berisi ARN volume sumber dari set snapshot multivolume. Bidang `startTime` dan `endTime` mengindikasikan kapan pembuatan snapshot dimulai dan diselesaikan.

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "EBS Multi-Volume Snapshots Completion Status",
  "source": "aws.ec2",
  "account": "012345678901",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2::us-east-1:snapshot/snap-01234567",
    "arn:aws:ec2::us-east-1:snapshot/snap-012345678"
  ],
  "detail": {
    "event": "createSnapshots",
    "result": "succeeded",
    "cause": "",
    "request-id": "",
    "startTime": "yyyy-mm-ddThh:mm:ssZ",
    "endTime": "yyyy-mm-ddThh:mm:ssZ",
    "snapshots": [
      {
        "snapshot_id": "arn:aws:ec2::us-east-1:snapshot/snap-01234567",
        "source": "arn:aws:ec2::us-east-1:volume/vol-01234567",
        "status": "completed"
      },
      {
```

```
        "snapshot_id": "arn:aws:ec2::us-east-1:snapshot/snap-012345678",
        "source": "arn:aws:ec2::us-east-1:volume/vol-012345678",
        "status": "completed"
      }
    ]
  }
}
```

Daftar di bawah ini adalah contoh objek JSON yang dipancarkan oleh EBS setelah peristiwa `createSnapshots` yang gagal. Penyebab kegagalan adalah satu atau lebih snapshot untuk set snapshot multi-volume gagal diselesaikan. Nilai-nilai `snapshot_id` adalah ARNs snapshot yang gagal. `startTime` dan `endTime` mewakili kapan tindakan create-snapshots dimulai dan berakhir.

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "EBS Multi-Volume Snapshots Completion Status",
  "source": "aws.ec2",
  "account": "012345678901",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2::us-east-1:snapshot/snap-01234567",
    "arn:aws:ec2::us-east-1:snapshot/snap-012345678"
  ],
"detail": {
    "event": "createSnapshots",
    "result": "failed",
    "cause": "Snapshot snap-01234567 is in status error",
   "request-id": "",
    "startTime": "yyyy-mm-ddThh:mm:ssZ",
    "endTime": "yyyy-mm-ddThh:mm:ssZ",
    "snapshots": [
      {
        "snapshot_id": "arn:aws:ec2::us-east-1:snapshot/snap-01234567",
        "source": "arn:aws:ec2::us-east-1:volume/vol-01234567",
        "status": "error"
      },
      {
        "snapshot_id": "arn:aws:ec2::us-east-1:snapshot/snap-012345678",
        "source": "arn:aws:ec2::us-east-1:volume/vol-012345678",
        "status": "error"
      }
```

```
    ]
  }
}
```

Salin snapshot (copySnapshot)

`copySnapshot`Acara dikirim ke AWS akun Anda ketika tindakan untuk menyalin snapshot selesai. Namun, peristiwa tersebut tidak disimpan, dicatat, atau diarsipkan. Peristiwa ini dapat membawa hasil `succeeded` atau `failed`.

Jika Anda menyalin snapshot di seluruh Wilayah, peristiwa tersebut dipancarkan di Wilayah tujuan.

Data peristiwa

Daftar di bawah ini adalah contoh objek JSON yang dipancarkan oleh EBS untuk peristiwa `copySnapshot` yang berhasil. Nilai dari `snapshot_id` adalah ARN dari snapshot yang baru dibuat. Di `detail` bagian ini, nilai `source` adalah ARN dari snapshot sumber. `startTime` dan `endTime` mewakili saat tindakan copy-snapshot dimulai dan berakhir. `incremental` menunjukkan apakah snapshot adalah snapshot inkremental (`true`), atau snapshot penuh (`false`).

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "EBS Snapshot Notification",
  "source": "aws.ec2",
  "account": "123456789012",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2::us-west-2:snapshot/snap-01234567"
  ],
  "detail": {
    "event": "copySnapshot",
    "result": "succeeded",
    "cause": "",
    "request-id": "",
    "snapshot_id": "arn:aws:ec2::us-west-2:snapshot/snap-01234567",
    "source": "arn:aws:ec2::eu-west-1:snapshot/snap-76543210",
    "startTime": "yyyy-mm-ddThh:mm:ssZ",
    "endTime": "yyyy-mm-ddThh:mm:ssZ",
    "incremental": "true"
  }
```

```
}
```

Daftar di bawah ini adalah contoh objek JSON yang dipancarkan oleh EBS setelah peristiwa `copySnapshot` yang gagal. Penyebab kegagalan tersebut adalah ID snapshot sumber yang tidak valid. Nilai dari `snapshot_id` adalah ARN dari snapshot yang gagal. Di bagian `detail`, nilai `source` adalah ARN dari sumber snapshot. `startTime` dan `endTime` mewakili ketika tindakan menyalin-snapshot dimulai dan berakhir.

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "EBS Snapshot Notification",
  "source": "aws.ec2",
  "account": "123456789012",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2::us-west-2:snapshot/snap-01234567"
  ],
  "detail": {
    "event": "copySnapshot",
    "result": "failed",
    "cause": "Source snapshot ID is not valid",
    "request-id": "",
    "snapshot_id": "arn:aws:ec2::us-west-2:snapshot/snap-01234567",
    "source": "arn:aws:ec2::eu-west-1:snapshot/snap-76543210",
    "startTime": "yyyy-mm-ddThh:mm:ssZ",
    "endTime": "yyyy-mm-ddThh:mm:ssZ"
  }
}
```

Bagikan snapshot (shareSnapshot)

`shareSnapshot`Acara dikirim ke AWS akun Anda ketika akun lain membagikan snapshot dengannya. Namun, peristiwa tersebut tidak disimpan, dicatat, atau diarsipkan. Hasilnya selalu `succeeded`.

Data peristiwa

Berikut adalah contoh objek JSON yang dipancarkan oleh EBS setelah peristiwa `shareSnapshot` yang gagal. Di `detail` bagian tersebut, nilainya `source` adalah nomor AWS akun pengguna yang membagikan snapshot dengan Anda. `startTime`dan `endTime` mewakili saat tindakan share-

snapshot dimulai dan berakhir. Peristiwa `shareSnapshot` ini hanya akan dilakukan saat snapshot privat dibagikan dengan pengguna lain. Berbagi snapshot publik tidak memicu peristiwa.

```
{
  "version": "0",
  "id": "01234567-01234-0123-0123-012345678901",
  "detail-type": "EBS Snapshot Notification",
  "source": "aws.ec2",
  "account": "012345678901",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2::us-west-2:snapshot/snap-01234567"
  ],
  "detail": {
    "event": "shareSnapshot",
    "result": "succeeded",
    "cause": "",
    "request-id": "",
    "snapshot_id": "arn:aws:ec2::us-west-2:snapshot/snap-01234567",
    "source": 012345678901,
    "startTime": "yyyy-mm-ddThh:mm:ssZ",
    "endTime": "yyyy-mm-ddThh:mm:ssZ"
  }
}
```

Peristiwa Arsip Snapshots EBS

Amazon EBS memancarkan peristiwa yang terkait dengan tindakan pengarsipan snapshot. Untuk informasi selengkapnya, lihat Memantau pengarsipan snapshot.

Peristiwa pemulihan snapshot cepat EBS

Amazon EBS mengirimkan peristiwa ke EventBridge saat status pemulihan snapshot cepat untuk snapshot berubah. Peristiwa dipancarkan atas dasar upaya terbaik.

Berikut adalah contoh data untuk peristiwa ini.

```
{
   "version": "0",
   "id": "01234567-0123-0123-0123-012345678901",
   "detail-type": "EBS Fast Snapshot Restore State-change Notification",
```

```
    "source": "aws.ec2",
    "account": "123456789012",
    "time": "yyyy-mm-ddThh:mm:ssZ",
    "region": "us-east-1",
    "resources": [
       "arn:aws:ec2:us-east-1::snapshot/snap-03a55cf56513fa1b6"
    ],
    "detail": {
       "snapshot-id": "snap-1234567890abcdef0",
       "state": "optimizing",
       "zone": "us-east-1a",
       "message": "Client.UserInitiated - Lifecycle state transition",
    }
}
```

Kemungkinan nilai untuk `state` adalah `enabling`, `optimizing`, `enabled`, `disabling`, dan `disabled`.

Kemungkinan nilai untuk `message` adalah sebagai berikut:

`Client.InvalidSnapshot.InvalidState - The requested snapshot transitioned to an invalid state (Error)`

Permintaan untuk mengaktifkan pemulihan snapshot cepat gagal dan status bertransisi ke `disabling` atau `disabled`. Pemulihan snapshot cepat tidak diaktifkan untuk snapshot ini.

`Client.UserInitiated`

Status berhasil dialihkan ke `enabling` atau `disabling`.

`Client.UserInitiated - Lifecycle state transition`

Status berhasil dialihkan ke `optimizing`, `enabled`, atau `disabled`.

`Server.InsufficientCapacity - There was insufficient capacity available to satisfy the request`

Permintaan untuk mengaktifkan pemulihan snapshot cepat gagal karena kapasitas tidak mencukupi, dan status bertransisi ke `disabling` atau `disabled`. Tunggu dan kemudian coba lagi.

`Server.InternalError - An internal error caused the operation to fail`

Permintaan untuk mengaktifkan pemulihan snapshot cepat gagal karena kesalahan internal, dan status bertransisi ke `disabling` atau `disabled`. Tunggu dan kemudian coba lagi.

`Client.InvalidSnapshot.InvalidState - The requested snapshot was deleted or access permissions were revoked`

Status pemulihan snapshot cepat untuk snapshot telah ditransisikan ke `disabling` atau `disabled` karena snapshot dihapus atau tidak dibagikan oleh pemilik snapshot. Pemulihan snapshot cepat tidak dapat diaktifkan untuk snapshot yang telah dihapus atau tidak lagi dibagikan kepada Anda.

Menggunakan AWS Lambda untuk menangani EventBridge acara

Anda dapat menggunakan Amazon EBS dan Amazon EventBridge untuk mengotomatiskan alur kerja pencadangan data Anda. Ini mengharuskan Anda untuk membuat kebijakan IAM, AWS Lambda fungsi untuk menangani acara, dan EventBridge aturan yang cocok dengan peristiwa yang masuk dan merutekkannya ke fungsi Lambda.

Prosedur berikut menggunakan peristiwa `createSnapshot` untuk menyalin snapshot yang sudah selesai secara otomatis ke Wilayah lain untuk pemulihan bencana.

Untuk menyalin snapshot yang sudah selesai ke Wilayah lain

1. Buat kebijakan IAM, seperti yang ditunjukkan dalam contoh berikut, untuk memberikan izin untuk menggunakan `CopySnapshot` tindakan dan menulis ke log. EventBridge Tetapkan kebijakan kepada pengguna yang akan menangani EventBridge acara tersebut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "arn:aws:logs:*:*:*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CopySnapshot"
      ],
```

```
      "Resource": "*"
    }
  ]
}
```

2. Tentukan fungsi di Lambda yang akan tersedia dari konsol. EventBridge Contoh fungsi Lambda di bawah ini, ditulis dalam Node.js, dipanggil EventBridge ketika `createSnapshot` peristiwa yang cocok dipancarkan oleh Amazon EBS (menandakan bahwa snapshot telah selesai). Saat diinvokasi, fungsi menyalin snapshot dari `us-east-2` untuk `us-east-1`.

```
// Sample Lambda function to copy an EBS snapshot to a different Region

var AWS = require('aws-sdk');
var ec2 = new AWS.EC2();

// define variables
var destinationRegion = 'us-east-1';
var sourceRegion = 'us-east-2';
console.log ('Loading function');

//main function
exports.handler = (event, context, callback) => {

    // Get the EBS snapshot ID from the event details
    var snapshotArn = event.detail.snapshot_id.split('/');
    const snapshotId = snapshotArn[1];
    const description = `Snapshot copy from ${snapshotId} in ${sourceRegion}.`;
    console.log ("snapshotId:", snapshotId);

    // Load EC2 class and update the configuration to use destination Region to
 initiate the snapshot.
    AWS.config.update({region: destinationRegion});
    var ec2 = new AWS.EC2();

    // Prepare variables for ec2.modifySnapshotAttribute call
    const copySnapshotParams = {
        Description: description,
        DestinationRegion: destinationRegion,
        SourceRegion: sourceRegion,
        SourceSnapshotId: snapshotId
    };

    // Execute the copy snapshot and log any errors
```

```
    ec2.copySnapshot(copySnapshotParams, (err, data) => {
        if (err) {
            const errorMessage = `Error copying snapshot ${snapshotId} to Region
 ${destinationRegion}.`;
            console.log(errorMessage);
            console.log(err);
            callback(errorMessage);
        } else {
            const successMessage = `Successfully started copy of snapshot
 ${snapshotId} to Region ${destinationRegion}.`;
            console.log(successMessage);
            console.log(data);
            callback(null, successMessage);
        }
    });
};
```

Untuk memastikan bahwa fungsi Lambda Anda tersedia dari EventBridge konsol, buat di Wilayah tempat EventBridge acara akan terjadi. Lihat informasi selengkapnya di Panduan Developer AWS Lambda.

3. Buka EventBridge konsol Amazon di https://console.aws.amazon.com/events/.
4. Di panel navigasi, pilih buat Aturan, lalu pilih Buat aturan.
5. Untuk Langkah 1: Tentukan detail aturan, lakukan hal berikut ini:
   a. Masukkan nilai untuk Nama dan Deskripsi.
   b. Untuk Bus peristiwa, tetap default.
   c. Pastikan Aktifkan aturan pada bus peristiwa yang dipilih telah diaktifkan.
   d. Untuk Tipe peristiwa, pilih Aturan dengan pola peristiwa.
   e. Pilih Berikutnya.
6. Untuk Langkah 2: Bangun pola peristiwa, lakukan hal berikut ini:
   a. Untuk sumber acara, pilih AWS acara atau acara EventBridge mitra.
   b. Di bagian Pola peristiwa, untuk sumber peristiwa, pastikan bahwa layanan AWS dipilih, dan untuk layanan AWS , pilih EC2.
   c. Untuk Tipe peristiwa, pilih Notifikasi Snapshot EBS, pilih Peristiwa spesifik, lalu pilih createSnapshot.
   d. Pilih Hasil spesifik dan kemudian pilih berhasil.

e. Pilih Berikutnya.

7. Untuk Langkah 3: Pilih target, lakukan hal berikut ini:

   a. Untuk Tipe target, pilih Layanan AWS .

   b. Untuk Pilih target, pilih fungsi Lambda, dan untuk Fungsi pilih fungsi yang Anda buat sebelumnya.

   c. Pilih Selanjutnya

8. Untuk Langkah 4: Konfigurasikan tanda, tentukan tanda untuk aturan jika diperlukan, lalu pilih Berikutnya.

9. Untuk Langkah 5: Tinjau dan buat, tinjau aturan lalu pilih Buat aturan.

Aturan Anda seharusnya kini muncul di tab Aturan. Pada contoh yang ditunjukkan, peristiwa yang Anda konfigurasikan harusoleh EBS pada saat Anda menyalin snapshot berikutnya.

Amazon GuardDuty untuk Amazon EBS

Amazon GuardDuty adalah layanan deteksi ancaman yang membantu melindungi akun, wadah, beban kerja, dan data di AWS lingkungan Anda. Menggunakan model machine learning (ML), serta kemampuan deteksi anomali dan ancaman, GuardDuty terus memantau berbagai sumber log dan aktivitas runtime untuk mengidentifikasi dan memprioritaskan potensi risiko keamanan dan aktivitas berbahaya di lingkungan Anda.

Fitur Perlindungan Malware dalam GuardDuty memindai volume Amazon EBS yang terkait dengan instans Amazon EC2 dan beban kerja container untuk mendeteksi potensi ancaman. GuardDuty menawarkan dua cara untuk melakukan ini:

- Aktifkan Perlindungan Malware — Saat GuardDuty menghasilkan temuan yang menunjukkan potensi keberadaan malware di instans Amazon EC2 atau beban kerja kontainer, ia akan secara otomatis memulai pemindaian malware pada sumber daya yang berpotensi disusupi.
- Gunakan pemindaian malware sesuai permintaan tanpa mengaktifkan Perlindungan Malware — Berikan Nama Sumber Daya Amazon (ARN) instans Amazon EC2 Anda untuk memulai pemindaian sesuai permintaan.

Untuk informasi selengkapnya, lihat Panduan GuardDuty Pengguna Amazon.

Kuota untuk Amazon EBS

Anda Akun AWS memiliki kuota default, sebelumnya disebut sebagai batas, untuk masing-masing. Layanan AWS Kecuali dinyatakan sebaliknya, setiap kuota unik untuk suatu Wilayah. Anda dapat meminta penambahan untuk beberapa kuota, sementara kuota lainnya tidak dapat ditambah.

Untuk melihat kuota Amazon EBS, buka konsol Service Quotas. Di panel navigasi, pilih AWS layanan dan pilih Amazon Elastic Block Store (Amazon EBS). Untuk meminta penambahan kuota, lihat Meminta Penambahan Kuota dalam Panduan Pengguna Kuota Layanan.

Anda Akun AWS memiliki kuota berikut yang terkait dengan Amazon EBS.

Nama	Default	Dapat disesuaikan	Deskripsi
Snapshot yang diarsipkan per volume	Setiap Wilayah yang didukung: 25	Ya	Jumlah maksimum snapshot yang diarsipkan per volume.
CompleteSnapshot permintaan per akun	Setiap Wilayah yang didukung: 10 per detik	Tidak	Jumlah maksimum CompleteSnapshot permintaan yang diizinkan per akun.
Salinan snapshot bersamaan per Wilayah tujuan	Setiap Wilayah yang didukung: 20	Tidak	Jumlah maksimum salinan snapshot bersamaan ke satu wilayah tujuan.
Snapshot bersamaan per volume Cold HDD (sc1)	Setiap Wilayah yang didukung: 1	Tidak	Jumlah maksimum snapshot bersamaan per volume Cold HDD (sc1) di Wilayah ini.
Snapshot bersamaan per volume SSD Tujuan Umum (gp2)	Setiap Wilayah yang didukung: 5	Tidak	Jumlah maksimum snapshot bersamaan per

Nama	Default	Dapat disesu an	Deskripsi
			volume General Purpose SSD (gp2) di Wilayah ini.
Snapshot bersamaan per volume SSD Tujuan Umum (gp3)	Setiap Wilayah yang didukung: 5	Tidak	Jumlah maksimum snapshot bersamaan per volume General Purpose SSD (gp3) di Wilayah ini.
Snapshot bersamaan per volume Magnetik (standar)	Setiap Wilayah yang didukung: 5	Tidak	Jumlah maksimum snapshot bersamaan per volume Magnetik (standar) di Wilayah ini.
Snapshot bersamaan per volume Provisioned IOPS SSD (io1)	Setiap Wilayah yang didukung: 5	Tidak	Jumlah maksimum snapshot bersamaan per volume Provisioned IOPS SSD (io1) di Wilayah ini.
Snapshot bersamaan per volume Provisioned IOPS SSD (io2)	Setiap Wilayah yang didukung: 5	Tidak	Jumlah maksimum snapshot bersamaan per volume Provisioned IOPS SSD (io2) di Wilayah ini.
Snapshot bersamaan per Volume HDD yang Dioptimalkan Throughput (st1)	Setiap Wilayah yang didukung: 1	Tidak	Jumlah maksimum snapshot bersamaan per volume Throughput Optimized HDD (st1) di Wilayah ini.

Nama	Default	Dapat disesuaikan	Deskripsi
Pemulihan snapshot cepat	us-timur-1:5 us-timur-2:5 kami-barat-1:5 kami-barat-2:5 af-selatan-1:5 ap-timur-1:5 ap-timur laut-1:5 ap-timur laut-2:5 ap-timur laut-3:5 ap-selatan-1:5 ap-tenggara 1:5 ap-tenggara 2:5 ap-tenggara 3:5 ca-pusat-1:5 eu-sentral-1:5 eu-utara-1:5 eu-selatan-1:5 eu-barat-1:5 eu-barat-2:5 eu-barat-3:5	Ya	Jumlah maksimum snapshot yang dapat diaktifkan untuk pemulihan snapshot cepat di Wilayah ini.

Nama	Default	Dapat disesuaikan	Deskripsi
	saya-selatan-1:5 sa-timur-1:5 Masing-masing Wilayah yang didukung lainnya: 5		
GetSnapshotBlock permintaan per akun	Setiap Wilayah yang didukung: 1.000 per detik	Ya	Jumlah maksimum GetSnapshotBlock permintaan yang diizinkan per akun.
GetSnapshotBlock permintaan per snapshot	Setiap Wilayah yang didukung: 1.000 per detik	Tidak	Jumlah maksimum GetSnapshotBlock permintaan yang diizinkan per snapshot.
IOPS untuk volume Provisioned IOPS SSD (io1)	Setiap Wilayah yang didukung: 300.000	Ya	Jumlah agregat maksimum IOPS yang dapat disediakan di seluruh volume IOPS SDD (io1) yang Disediakan di Wilayah ini.
IOPS untuk volume Provisioned IOPS SSD (io2)	Setiap Wilayah yang didukung: 100.000	Ya	Jumlah agregat maksimum IOPS yang dapat disediakan di seluruh volume IOPS SDD (io2) yang Disediakan di Wilayah ini.

Nama	Default	Dapat disesuaikan	Deskripsi
Modifikasi IOPS untuk volume Provisioned IOPS SSD (io1)	Setiap Wilayah yang didukung: 500.000	Ya	Jumlah agregat maksimum IOPS yang dapat diminta dalam modifikasi volume di seluruh volume IOPS SSD (io1) yang Disediakan di Wilayah ini.
Modifikasi IOPS untuk volume Provisioned IOPS SSD (io2)	Setiap Wilayah yang didukung: 100.000	Ya	IOPS arus maksimum (dari) dan permintaan (ke) untuk permintaan modifikasi volume di seluruh volume IOPS SSD (io2) yang Disediakan di Wilayah ini.
Arsip snapshot dalam proses per akun	Setiap Wilayah yang didukung: 25	Ya	Jumlah maksimum arsip snapshot yang sedang berlangsung per akun.
Snapshot dalam proses memulihkan dari arsip per akun	Setiap Wilayah yang didukung: 5	Ya	Jumlah maksimum snapshot yang sedang berlangsung dipulihkan dari arsip per akun.
ListChangedBlocks permintaan per akun	Setiap Wilayah yang didukung: 50 per detik	Tidak	Jumlah maksimum ListChangedBlocks permintaan yang diizinkan per akun.
ListSnapshotBlocks permintaan per akun	Setiap Wilayah yang didukung: 50 per detik	Tidak	Jumlah maksimum ListSnapshotBlocks permintaan yang diizinkan per akun.

Nama	Default	Dapat disesuaikan	Deskripsi
Snapshot yang tertunda per akun	Setiap Wilayah yang didukung: 100	Tidak	Jumlah maksimum snapshot dalam status tertunda per akun.
PutSnapshotBlock permintaan per akun	Setiap Wilayah yang didukung: 1.000 per detik	Ya	Jumlah maksimum PutSnapshotBlock permintaan yang diizinkan per akun.
PutSnapshotBlock permintaan per snapshot	Setiap Wilayah yang didukung: 1.000 per detik	Tidak	Jumlah maksimum PutSnapshotBlock permintaan yang diizinkan per snapshot.
Snapshot per Wilayah	Setiap Wilayah yang didukung: 100.000	Ya	Jumlah maksimum snapshot per Wilayah
StartSnapshot permintaan per akun	Setiap Wilayah yang didukung: 10 per detik	Tidak	Jumlah maksimum StartSnapshot permintaan yang diizinkan per akun.

Nama	Default	Dapat disesuaikan	Deskripsi
Penyimpanan untuk volume Cold-HDD (sc1) dalam TiB	af-selatan-1:300 ap-timur-1:300 eu-selatan-1:300 saya-selatan-1:300 Masing-masing Wilayah yang didukung lainnya: 50	Ya	Jumlah penyimpanan agregat maksimum, di TiB, yang dapat disediakan di seluruh volume Cold HDD (sc1) di Wilayah ini.
Penyimpanan untuk volume SSD Tujuan Umum (gp2) dalam TiB	af-selatan-1:300 ap-timur-1:300 eu-selatan-1:300 saya-selatan-1:300 Masing-masing Wilayah yang didukung lainnya: 50	Ya	Jumlah penyimpanan agregat maksimum, di TiB, yang dapat disediakan di seluruh volume General Purpose SSD (gp2) di Wilayah ini.

Nama	Default	Dapat disesuaikan	Deskripsi
Penyimpanan untuk volume SSD Tujuan Umum (gp3) dalam TiB	af-selatan-1:300 ap-timur-1:300 eu-selatan-1:300 saya-selatan-1:300 Masing-masing Wilayah yang didukung lainnya: 50	Ya	Jumlah penyimpanan agregat maksimum, di TiB, yang dapat disediakan di seluruh volume General Purpose SSD (gp3) di Wilayah ini.
Penyimpanan untuk volume Magnetik (standar) dalam TiB	af-selatan-1:300 ap-timur-1:300 eu-selatan-1:300 saya-selatan-1:300 Masing-masing Wilayah yang didukung lainnya: 50	Ya	Jumlah penyimpanan agregat maksimum, di TiB, yang dapat disediakan di seluruh volume Magnetik (standar) di Wilayah ini.

Nama	Default	Dapat disesuaikan	Deskripsi
Penyimpanan untuk volume Provisioned IOPS SSD (io1) EBS	af-selatan-1:300 ap-timur-1:300 eu-selatan-1:300 saya-selatan-1:300 Masing-masing Wilayah yang didukung lainnya: 50	Ya	Jumlah penyimpanan agregat maksimum, di TiB, yang dapat disediakan di seluruh volume IOPS SSD (io1) yang Disediakan di Wilayah ini.
Penyimpanan untuk volume Provisioned IOPS SSD (io2) EBS	Setiap Wilayah yang didukung: 20	Ya	Jumlah penyimpanan agregat maksimum, di TiB, yang dapat disediakan di seluruh volume IOPS SSD (io2) yang Disediakan di Wilayah ini.
Penyimpanan untuk volume HDD yang Dioptimalkan Throughput (st1) dalam TiB	af-selatan-1:300 ap-timur-1:300 eu-selatan-1:300 saya-selatan-1:300 Masing-masing Wilayah yang didukung lainnya: 50	Ya	Jumlah penyimpanan agregat maksimum, di TiB, yang dapat disediakan di seluruh volume HDD yang Dioptimalkan Throughput (st1) di Wilayah ini.

Nama	Default	Dapat disesuaikan	Deskripsi
Modifikasi penyimpanan untuk volume Cold-HDD (sc1) dalam TiB	Setiap Wilayah yang didukung: 500	Ya	Jumlah penyimpanan agregat maksimum, di TiB, yang dapat diminta dalam modifikasi volume di seluruh volume Cold HDD (sc1) di Wilayah ini.
Modifikasi penyimpanan untuk volume SSD Tujuan Umum (gp2) dalam TiB	Setiap Wilayah yang didukung: 500	Ya	Jumlah penyimpanan agregat maksimum, di TiB, yang dapat diminta dalam modifikasi volume di seluruh volume General Purpose SSD (gp2) di Wilayah ini.
Modifikasi penyimpanan untuk volume SSD Tujuan Umum (gp3) dalam TiB	Setiap Wilayah yang didukung: 500	Ya	Jumlah penyimpanan agregat maksimum, di TiB, yang dapat diminta dalam modifikasi volume di seluruh volume General Purpose SSD (gp3) di Wilayah ini.
Modifikasi penyimpanan untuk volume Magnetik (standar) dalam TiB	Setiap Wilayah yang didukung: 500	Ya	Jumlah penyimpanan agregat maksimum, di TiB, yang dapat diminta dalam modifikasi volume di seluruh volume Magnetik (standar) di Wilayah ini.

Nama	Default	Dapat disesuaikan	Deskripsi
Modifikasi penyimpanan untuk volume Provisioned IOPS SSD (io1) dalam TiB	Setiap Wilayah yang didukung: 500	Ya	Jumlah penyimpanan agregat maksimum, di TiB, yang dapat diminta dalam modifikasi volume di seluruh volume IOPS SSD (io1) yang Disediakan di Wilayah ini.
Modifikasi penyimpanan untuk volume Provisioned IOPS SSD (io2) dalam TiB	Setiap Wilayah yang didukung: 20	Ya	Jumlah penyimpanan agregat maksimum, di TiB, yang dapat diminta dalam modifikasi volume di seluruh volume IOPS SSD (io2) yang Disediakan di Wilayah ini.
Modifikasi penyimpanan untuk volume HDD yang Dioptimalkan Throughput (st1) dalam TiB	Setiap Wilayah yang didukung: 500	Ya	Jumlah penyimpanan agregat maksimum, di TiB, yang dapat diminta dalam modifikasi volume di seluruh volume HDD yang Dioptimalkan Throughput (st1) di Wilayah ini.

Pertimbangan

- Kuota Anda dapat berubah seiring waktu. Amazon EBS terus memantau penyimpanan yang disediakan dan penggunaan IOPS Anda di setiap Wilayah dan dapat secara otomatis meningkatkan kuota Anda, berdasarkan per wilayah, berdasarkan penggunaan Anda. Meskipun Amazon EBS dapat secara otomatis meningkatkan kuota berdasarkan penggunaan Anda, Anda dapat meminta peningkatan kuota jika diperlukan. Misalnya, jika Anda berencana untuk menggunakan lebih banyak gp3 penyimpanan di AS Timur (Virginia Utara) daripada kuota Anda

saat ini, Anda dapat meminta peningkatan kuota untuk jenis volume tersebut di Wilayah tersebut sebelum penggunaan yang direncanakan.

- Kuota untuk salinan snapshot Bersamaan per Wilayah tujuan tidak dapat disesuaikan menggunakan Service Quotas. Namun, Anda dapat meminta kenaikan kuota ini dengan menghubungi AWS Support.
- Kuota modifikasi IOPS dan modifikasi Penyimpanan berlaku untuk nilai arus agregat (untuk ukuran atau IOPS, tergantung pada kuota) volume yang dapat mengalami modifikasi secara bersamaan. Anda dapat membuat permintaan modifikasi bersamaan untuk volume yang telah menggabungkan nilai saat ini (untuk ukuran atau IOPS) hingga kuota. Misalnya, jika modifikasi IOPS Anda untuk kuota volume IOPS SSD (io1) yang disediakan adalah`50,000`, Anda dapat membuat permintaan modifikasi IOPS bersamaan untuk sejumlah `io1` volume selama IOPS gabungan saat ini sama dengan atau kurang dari. `50,000` Jika Anda memiliki tiga `io1` volume yang masing-masing disediakan dengan `20,000` IOPS, Anda dapat meminta modifikasi IOPS untuk dua volume secara bersamaan (). `20,000 * 2 < 50,000` Jika Anda mengirimkan permintaan modifikasi IOPS bersamaan untuk volume ketiga, Anda melebihi kuota dan permintaan tersebut gagal (). `20,000 * 3 > 50,000`

Riwayat dokumen untuk Panduan Pengguna Amazon EBS

Tabel berikut menjelaskan rilis dokumentasi untuk Amazon EBS.

Perubahan	Deskripsi	Tanggal
Aktifkan kebijakan default Amazon Data Lifecycle Manager di seluruh akun	Anda dapat menggunak annya AWS CloudFormation StackSets untuk mengaktif kan kebijakan default Amazon Data Lifecycle Manager di seluruh AWS organisasi atau di seluruh akun tertentu. AWS	April 26, 2024
AWSDataLifecycleMa nagerSSMFullAccess AWS kebijakan terkelola	Memperbarui kebijakan untuk mendukung snapshot yang konsisten dengan aplikasi untuk SAP HANA menggunak an dokumen SSM `AWSSystem sManagerSAP-Create DLMSnapshotForSAPH ANA` .	17 November 2023
VolumeStalledMetrik IOCheck	Anda dapat menggunak an metrik `VolumeSta lledIOCheck` untuk memeriksa apakah volume telah berhasil atau gagal melewati pemeriksaan IO yang macet di menit terakhir.	16 November 2023
Kebijakan default Amazon Data Lifecycle Manager	Anda sekarang dapat membuat kebijakan default Amazon Data Lifecycle Manager untuk snapshot EBS dan AMI yang didukung EBS untuk mencadangkan	16 November 2023

	semua volume dan instans di Wilayah.	
Kunci snapshot Amazon EBS	Anda dapat mengunci snapshot Amazon EBS untuk melindunginya dari penghapusan yang tidak disengaja atau berbahaya, atau menyimpannya dalam format WORM untuk durasi tertentu.	15 November 2023
Blokir akses publik untuk snapshot	Anda sekarang dapat menggunakan blokir akses publik untuk snapshot guna mencegah berbagi snapshot secara publik.	9 November 2023
Praskrip dan pascaskrip Amazon Data Lifecycle Manager	Sekarang Anda dapat menggunakan praskrip dan pascaskrip dalam kebijakan snapshot Amazon Data Lifecycle Manager untuk mengotomatisasi siklus hidup snapshot yang konsisten dengan aplikasi.	7 November 2023
Reservasi NVMe	Volume `io2` yang diaktifkan Multi-Lampiran mendukung reservasi NVMe, yang merupakan set protokol pagar penyimpanan standar industri.	18 September 2023

Pengujian kesalahan pada Amazon EBS	Gunakan AWS FIS untuk menghentikan sementara I/O antara volume EBS dan instance yang dilampirkan untuk menguji bagaimana beban kerja Anda menangani interupsi I/O.	27 Januari 2023
Volume `io2` Block Express	Anda dapat memodifikasi ukuran dan IOPS yang tersedia dari volume `io2` Block Express dan Anda dapat mengaktifkannya untuk pemulihan snapshot cepat.	31 Mei 2022
Keranjang Sampah untuk snapshot Amazon EBS	Keranjang Sampah untuk snapshot Amazon EBS adalah fitur pemulihan snapshot yang memungkinkan Anda memulihkan snapshot yang terhapus secara tidak sengaja.	29 November 2021
Arsip Snapshot Amazon EBS	Arsip Snapshot Amazon EBS adalah tingkat penyimpan an baru yang dapat Anda gunakan untuk penyimpanan berbiaya rendah dan jangka panjang dari snapshot yang jarang diakses.	29 November 2021
CloudWatch metrik untuk Amazon Data Lifecycle Manager	Anda dapat memantau kebijakan Amazon Data Lifecycle Manager menggunak an Amazon. CloudWatch	28 Juli 2021

CloudTrail peristiwa data untuk API langsung EBS	API ListSnapshotBlocks , ListChangedBlocks GetSnapshotBlock, dan PutSnapshotBlock API dapat dicatat peristiwa data di CloudTrail.	27 Juli 2021
Volume `io2` Block Express	`io2`Volume Block Express sekarang tersedia secara umum.	19 Juli 2021
Snapshot lokal Amazon EBS di Outposts	Anda sekarang dapat menggunakan snapshot lokal Amazon EBS pada Outposts untuk menyimpan snapshot volume pada Outpost secara lokal di Amazon S3 pada Outpost itu sendiri.	4 Februari 2021
Dukungan Multi-Lampirkan untuk volume io2	Anda sekarang dapat mengaktifkan volume SSD IOPS yang tersedia (`io2`) untuk Multi-Lampiran Amazon EBS.	18 Desember 2020
Amazon Data Lifecycle Manager	Gunakan Amazon Data Lifecycle Manager untuk mengotomatiskan proses berbagi snapshot dan menyalinnya di seluruh akun. AWS	17 Desember 2020

Volume gp3	Tipe volume SSD Tujuan Umum Amazon EBS baru. Anda dapat menentukan IOPS yang tersedia dan throughput saat Anda membuat atau memodifikasi volume.	1 Desember 2020
Ukuran volume HDD dengan throughput yang dioptimalkan dan HDD Cold	Volume HDD dengan throughput yang dioptimalkan (`st1`) dan HDD Cold (`sc1`) dapat mempunyai ukuran berkisar dari 125 GiB hingga 16 TiB.	30 November 2020
Amazon Data Lifecycle Manager	Anda dapat menggunakan Amazon Data Lifecycle Manager untuk mengotomatisasi pembuatan, retensi, dan penghapusan AMI yang didukung EBS.	9 November 2020
Amazon Data Lifecycle Manager	Kebijakan Amazon Data Lifecycle Manager dapat dikonfigurasi dengan hingga empat jadwal.	17 September 2020
Volume IOPS SSD (io2) yang disediakan untuk Amazon EBS	Volume (`io2`) IOPS SSD yang disediakan dirancang untuk memberikan ketahanan volume 99,999 persen dengan AFR tidak lebih dari 0,001 persen.	24 Agustus 2020
Pemulihan snapshot cepat	Anda dapat mengaktifkan pemulihan snapshot cepat untuk snapshot yang dibagikan dengan Anda.	21 Juli 2020

Amazon EBS Multi-Lampiran	Anda sekarang dapat melampirkan satu volume SSD IOPS yang Tersedia (io1) ke hingga 16 instans berbasis Nitro yang berada di Zona Ketersediaan yang sama.	14 Februari 2020
Pemulihan snapshot cepat Amazon EBS	Anda dapat mengaktifkan pemulihan snapshot cepat pada snapshot EBS untuk memastikan bahwa volume EBS yang dibuat dari snapshot telah sepenuhnya diinisialisasi saat pembuatan dan langsung memberikan semua performa yang disediakan.	20 November 2019
Snapshot multi-volume Amazon EBS	Anda dapat mengambil snapshot yang akurat point-in-time, terkoordinasi dengan data, dan konsisten crash di beberapa volume EBS yang dilampirkan ke instans EC2.	29 Mei 2019
Enkripsi Amazon EBS secara default	Setelah Anda mengaktif kan enkripsi secara default di suatu Wilayah, semua volume EBS baru yang dibuat di Wilayah tersebut akan dienkripsi dengan kunci KMS default untuk enkripsi EBS.	23 Mei 2019

Mengotomatiskan siklus hidup snapshot	Anda dapat menggunakan Amazon Data Lifecycle Manager guna mengotomatisasi pembuatan dan penghapusan snapshot untuk volume EBS Anda.	12 Juli 2018
Lakukan modifikasi pada volume EBS terlampir	Dengan sebagian besar volume EBS terlampir ke sebagian besar instans EC2, Anda dapat mengubah ukuran volume, tipe, dan IOPS tanpa mencopot lampiran volume atau menghentikan instans.	13 Februari 2017
Salin snapshot Amazon EBS terenkripsi antara Akun AWS	Anda sekarang dapat menyalin snapshot EBS terenkripsi di antaranya. Akun AWS	Juni 21, 2016
Throughput Dioptimalkan HDD dan tipe volume Cold HDD	Anda sekarang dapat membuat volume HDD Throughput yang Dioptimalkan (st1) dan HDD Cold (sc1).	19 April 2016
Jenis volume SSD Tujuan Umum	Volume SSD Tujuan Umum menawarkan penyimpanan hemat biaya yang ideal untuk berbagai beban kerja. Volume ini memberikan latensi satu digit milidetik, kemampuan melonjak hingga 3.000 IOPS untuk waktu yang lama, dan performa dasar 3 IOPS/GiB. Ukuran volume SSD Tujuan Umum dapat bervariasi mulai 1 GiB hingga 1 TiB.	Juni 16, 2014

Enkripsi Amazon EBS	Enkripsi Amazon EBS menawarkan enkripsi volume dan snapshot data EBS yang mulus, sehingga tidak perlu membangun dan memelihara infrastruktur manajemen kunci yang aman. Enkripsi EBS memungkinkan keamanan data diam dengan mengenkripsi data Anda menggunakan Kunci yang dikelola AWS. Enkripsi dilakukan di server yang melakukan hosting instans EC2, yang menyediakan enkripsi data saat berpindah antara instans EC2 dan penyimpanan EBS.	Mei 21, 2014
Salinan snapshot tambahan	Anda sekarang dapat menjalankan salinan snapshot inkremental.	11 Juni 2013
Salinan snapshot EBS	Anda dapat menggunakan salinan snapshot untuk membuat cadangan data, untuk membuat volume Amazon EBS baru, atau untuk membuat Amazon Machine Image (AMI).	17 Desember 2012

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