aws

ユーザーガイド

Amazon EBS



Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon EBS: ユーザーガイド

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon の商標およびトレードドレスは Amazon 以外の製品およびサービスに使用することはできません。また、お客様に誤解を与える可能性がある形式で、または Amazon の信用を損なう形式で使用することもできません。Amazon が所有していないその他のすべての商標は Amazon との提携、関連、支援関係の有無にかかわらず、それら該当する所有者の資産です。

Table of Contents

Amazon EBS とは？ 1
Amazon EBS の機能 1
関連サービス 2
Amazon EBS へのアクセス 3
料金 4
Amazon EBS のセットアップ 5
にサインアップする AWS アカウント 5
(オプション) Amazon EBS 暗号化用のカスタマーマネージドキーを作成および使用する 5
(オプション) Amazon EBS スナップショットのブロックパブリックアクセスを有効にする 6
EBS ボリューム 8
機能とメリット 9
データの可用性 9
データの永続性 10
データの暗号化 11
データセキュリティ 11
スナップショット 11
柔軟性 12
EBS ボリュームの種類 13
ソリッドステートドライブ (SSD) ボリューム 13
ハードディスクドライブ (HDD) ボリューム 16
旧世代のボリューム 17
汎用 SSD ボリューム 17
Provisioned IOPS SSD ボリューム 23
スループット最適化 HDD ボリュームと Cold HDD ボリューム 27
EBS ボリュームの制約 38
ストレージキャパシティ 38
サービスの制限 39
パーティションスキーム 40
データブロックサイズ 41
EBS ボリュームと NVMe 44
ボリュームをデバイス名にマッピング 45
I/O オペレーションタイムアウト 49
Abort コマンド 50
ボリュームのライフサイクル 50

ボリュームの作成 ... 52
ボリュームを共有する ... 56
ボリュームをコピーする ... 62
インスタンスへのボリュームのアタッチ ... 69
複数のインスタンスへのボリュームのアタッチ ... 72
ボリュームを使用できるようにする ... 81
ボリュームの詳細の表示 ... 95
ボリュームの変更 ... 102
インスタンスからのボリュームのデタッチ ... 128
ボリュームの削除 ... 133
ボリュームの置き換え ... 134
ステータスチェック ... 137
ボリュームイベント ... 141
障害のあるボリュームの操作 ... 143
I/O の自動有効化 ... 146
障害テスト ... 148
利用可能な実験 ... 149
考慮事項 ... 149
I/O フォールトインジェクションを一時停止する ... 149
レイテンシーインジェクション ... 151
EBS スナップショット ... 155
スナップショットの仕組み ... 156
スナップショットのライフサイクル ... 160
スナップショットの作成 ... 161
スナップショットに関する情報の表示 ... 168
スナップショットをコピーする ... 171
スナップショットの共有 ... 186
スナップショットのアーカイブ ... 194
スナップショットの削除 ... 229
高速スナップショット復元 ... 234
考慮事項 ... 235
料金と請求 ... 235
ボリューム作成クレジット ... 236
高速スナップショット復元を設定 ... 237
高速スナップショット復元の状態を確認 ... 240
高速スナップショット復元を使用して復元したボリュームの表示 ... 242

スナップショットロック ... 243
概念 ... 244
考慮事項 ... 247
アクセス制御 ... 248
スナップショットのロック ... 251
スナップショットをロック解除する ... 254
スナップショットのロック設定を更新する ... 255
スナップショットロックのモニタリング ... 256
スナップショットのパブリックアクセスのブロック ... 259
IAM 許可 ... 261
パブリックアクセスブロックの設定 ... 263
ブロックパブリックアクセス設定を表示 ... 267
ブロックパブリックアクセスを無効化 ... 269
ブロックパブリックアクセスをモニタリング ... 272
Outposts 上でのローカルスナップショット ... 274
よくある質問 ... 275
前提条件 ... 277
考慮事項 ... 72
IAM によるアクセスの制御 ... 278
ローカルスナップショット を使用する ... 281
ローカルゾーン内のローカルスナップショット ... 287
よくある質問 ... 275
考慮事項 ... 72
IAM によるアクセスの制御 ... 290
EBS 暗号化 ... 294
EBS 暗号化の仕組み ... 294
暗号化されたスナップショットに対する EBS 暗号化の動作 ... 295
暗号化されていないスナップショットに対する EBS 暗号化の動作 ... 295
使用できない KMS キーがデータキーに及ぼす影響 ... 296
Local Zones、Outposts、および Wavelength Zones ... 297
ローカルゾーン ... 297
Outposts と Wavelength Zones ... 298
要件 ... 298
サポートされるボリュームタイプ ... 298
サポートされるインスタンスタイプ ... 299
ユーザーのアクセス許可 ... 299

インスタンスの権限 300
デフォルトで暗号化の有効化 301
デフォルトの暗号化キー 305
EBS リソースの暗号化 306
作成時の空のボリュームの暗号化 307
暗号化されていないリソースの暗号化 307
KMS キーをローテーション 308
例 309
暗号化されていないボリュームを復元する (デフォルトでの暗号化が有効になっていない場合) 309
暗号化されていないボリュームを復元する (デフォルトでの暗号化が有効になっている場合) 310
暗号化されていないスナップショットをコピーする (デフォルトでの暗号化が有効になっていない場合) 311
暗号化されていないスナップショットをコピーする (デフォルトでの暗号化が有効になっている場合) 311
暗号化ボリュームを再暗号化する 312
暗号化スナップショットを再暗号化する 312
暗号化されたボリュームと暗号化されていないボリュームとの間でデータを移行する 313
暗号化の結果 314
EBS パフォーマンス 317
Amazon EBS パフォーマンスのヒント 317
EBS 最適化インスタンスを使用する 317
インスタンス帯域幅を設定する 318
パフォーマンスの計算方法を理解する 318
ワークロードを理解する 318
スナップショットからボリュームを初期化する際のパフォーマンス低下に注意する 318
HDD パフォーマンスが低下する要因 319
st1 および sc1 (Linux インスタンスインスタンスのみ) で高いスループットの読み取りが多いワークロードに先読みを増やす 319
最新の Linux カーネルを使用する (Linux インスタンスのみ) 320
RAID 0 を使用してインスタンスのリソース使用率を最大化する 321
Amazon EBS ボリュームのパフォーマンスをモニタリングする 321
EBS 最適化 321
ボリュームの初期化 322
ボリューム初期化に Amazon EBS プロビジョンドレートを使用する 323

高速スナップショット復元が有効になっているスナップショットを使用する 328
ボリュームを手動で初期化する 328
ボリューム初期化をモニタリングする 333
設定可能なインスタンス帯域幅の重み付け 337
I/O の特性とモニタリング 338
IOPS 339
ボリューム のキュー長とレイテンシー 341
I/O サイズとボリュームのスループット制限 342
CloudWatch を使用して I/O 特性を監視する 342
リアルタイムの I/O パフォーマンス統計をモニタリングする 344
関連リソース 345
RAID 設定 345
RAID 設定オプション 345
RAID 0 アレイの作成 346
RAID アレイでのボリュームのスナップショットの作成 355
EBS ボリュームのベンチマーク 356
インスタンスのセットアップ 356
ベンチマークツールのインストール 358
ボリュームキュー長の選択 359
C ステートの無効化 360
ベンチマーキングを実行する 361
Amazon Data Lifecycle Manager 366
クォータ 366
仕組み 367
ポリシー 367
ポリシースケジュール 369
ターゲットリソースタグ 370
スナップショット 370
EBS-backed AMI 370
Amazon Data Lifecycle Manager のタグ 370
デフォルトポリシーとカスタムポリシー 371
EBS スナップショットポリシーの比較 371
EBS-backed AMI ポリシーの比較 374
デフォルトポリシーを作成する 376
デフォルトポリシーに関する考慮事項 376
Amazon EBS スナップショットのデフォルトポリシーを作成する 377

EBS-backed AMI のデフォルトポリシーを作成する 381
アカウントとリージョン間でデフォルトポリシーを有効にする 385
スナップショット用のカスタムポリシーを作成 390
スナップショットライフサイクルポリシーを作成する 390
スナップショットライフサイクルポリシーに関する考慮事項 407
その他のリソース 413
アプリケーション整合性のあるスナップショットを自動化 413
事前スクリプトと事後スクリプトのその他のユースケース 450
事前スクリプトと事後スクリプトの仕組み 459
事前スクリプトと事後スクリプトで作成されたスナップショットの識別 462
事前スクリプトと事後スクリプトをモニタリング 463
AMI 用のカスタムポリシーを作成 464
AMI ライフサイクルポリシーを作成する 464
AMI ライフサイクルポリシーに関する考慮事項 471
その他のリソース 475
クロスアカウントのスナップショットコピーの自動化 475
クロスアカウントスナップショットコピーポリシーの作成 476
スナップショット説明フィルターの指定 487
クロスアカウントスナップショットコピーポリシーに関する考慮事項 488
その他のリソース 488
ポリシーの変更 489
ポリシーを削除 492
アクセス制御 493
AWS マネージドポリシー 495
IAM サービスロール 504
ポリシーをモニタリング 511
コンソールと AWS CLI 511
EventBridge を使用してポリシーをモニタリング 511
CloudWatch を使用して、ポリシーをモニタリング 514
を使用した AWS CloudTrail API コールのログ記録 528
サービスエンドポイント 532
IPv4 エンドポイント 533
デュアルスタック (IPv4 および IPv6) エンドポイント 533
FIPS エンドポイント 533
エンドポイントの指定 534
インターフェイス VPC エンドポイント 534

Amazon EBS の VPC エンドポイントに関する考慮事項 535
Amazon EBS 用のインターフェイス VPC エンドポイントを作成する 536
トラブルシューティング 536
エラー: Role with name already exists 536
AWS Backup Amazon EBS 用 538
Amazon EBS ダイレクト API 539
料金 540
API の料金 540
ネットワークコスト 540
概念 541
スナップショット 541
ブロック 541
ブロックインデックス 541
ブロックトークン 541
チェックサム 542
暗号化 542
API アクション 542
署名バージョン 4 の署名 543
アクセス制御 543
スナップショットの読み取り 549
スナップショット内のブロックの一覧表示 550
2 つのスナップショット間で異なるブロックの一覧表示 553
スナップショットからのブロックデータの取得 556
スナップショットへの書き込み 558
スナップショットの開始 559
スナップショットへのデータの書き込み 561
スナップショットの完了 563
暗号化の結果 564
暗号化の結果: 暗号化されていない親スナップショット 565
暗号化の結果: 暗号化された親スナップショット 565
暗号化の結果: 親スナップショットなし 566
スナップショットデータを検証 568
べき等性の確保 568
エラーの再試行 570
パフォーマンスの最適化 573
サービスエンドポイント 574

IPv4 エンドポイント ... 574
デュアルスタック (IPv4 および IPv6) エンドポイント ... 575
FIPS エンドポイント ... 576
エンドポイントの指定 ... 576
SDK コード例 ... 578
StartSnapshot ... 578
PutSnapshotBlock ... 579
CompleteSnapshot ... 580
インターフェイス VPC エンドポイント ... 581
EBS direct API VPC エンドポイントに関する考慮事項 ... 581
EBS direct API でインターフェイス VPC エンドポイントを作成する ... 582
CloudTrail ログ ... 583
CloudTrail の EBS direct API データイベント ... 584
CloudTrail の EBS direct API 管理イベント ... 585
EBS direct API イベント例 ... 529
よくある質問 ... 592
ごみ箱 ... 594
サポートされているリソース ... 595
動作の仕組み ... 595
考慮事項 ... 596
クォータ ... 600
関連サービス ... 600
料金 ... 601
アクセスを制御 ... 601
ごみ箱および保持ルールを操作するための許可 ... 602
ごみ箱内のリソースを操作するための許可 ... 603
[Condition keys for Recycle Bin] (ごみ箱の条件キー) ... 604
保持ルールを作成 ... 607
保持ルールの更新 ... 611
保持ルールをロック ... 613
保持ルールのロック解除 ... 615
保持ルールをタグ ... 616
保持ルールのタグを表示する ... 617
保持ルールからタグを削除する ... 618
保持ルールの削除 ... 619
削除されたスナップショットを復元 ... 620

ごみ箱のスナップショットを操作するための権限 ... 620
ごみ箱のスナップショットを表示する ... 621
ごみ箱からスナップショットを復元する ... 623
削除されたボリュームを復旧する ... 624
ごみ箱内のボリュームを操作するためのアクセス許可 ... 624
ごみ箱のボリュームを表示する ... 626
ごみ箱からボリュームを復元する ... 627
削除された AMI の復元 ... 628
ごみ箱内の AMI を操作するための許可 ... 629
ごみ箱内の AMI を表示する ... 630
ごみ箱から AMI を復元する ... 631
EventBridge を使用したモニタリング ... 633
RuleLocked ... 633
RuleChangeAttempted ... 634
RuleUnlockScheduled ... 635
RuleUnlockingNotice ... 635
RuleUnlocked ... 636
CloudTrail を使用してモニタリングする ... 636
CloudTrail でのごみ箱情報 ... 528
ごみ箱ログファイルエントリについて ... 638
サービスエンドポイント ... 651
IPv4 エンドポイント ... 574
デュアルスタック (IPv4 および IPv6) エンドポイント ... 652
FIPS エンドポイント ... 653
エンドポイントの指定 ... 653
インターフェイス VPC エンドポイントを使用 ... 654
ごみ箱のインターフェイス VPC エンドポイントを作成 ... 654
ごみ箱用の VPC エンドポイントポリシーを作成 ... 654
セキュリティ ... 656
データ保護 ... 656
Amazon EBS のデータセキュリティ ... 658
保管中と転送中の暗号化 ... 658
KMS キー管理 ... 658
Identity and Access Management ... 659
オーディエンス ... 659
アイデンティティを使用した認証 ... 660

ポリシーを使用したアクセスの管理 661
EBS と IAM が連携する仕組み 663
IAM ポリシーの例 668
トラブルシューティング 687
コンプライアンス検証 689
データ回復力 690
モニタリング 691
Amazon CloudWatch 692
Amazon EBS ボリュームのメトリクス 692
Amazon EBS スナップショットのメトリクス 715
Nitro インスタンスのメトリクス 715
高速スナップショット復元のメトリクス 720
Amazon EC2 コンソールのグラフ 721
Amazon EventBridge 723
EBS ボリュームイベント 724
EBS ボリューム初期化イベント 730
EBS ボリュームの変更イベント 731
EBS スナップショットイベント 731
EBS スナップショットのアーカイブイベント 740
EBS 高速スナップショット復元イベント 740
AWS Lambda を使用して EventBridge イベントを処理する 741
EBS の詳細なパフォーマンス統計 745
統計 746
統計へのアクセス 747
Amazon GuardDuty 750
請求および使用状況レポート 750
スナップショット 751
ボリュームストレージ 752
プロビジョンドパフォーマンス 753
EBS direct API 754
高速スナップショット復元 (FSR) 755
EBS ボリュームをインベントリに登録する 755
ボリューム 756
I/O パフォーマンス 758
EBS データボリュームの内容 759
スナップショット 759

EBS スナップショットの内容 760
クォータ 762
ドキュメント履歴 783
..... dccxcvi

Amazon Elastic Block Store とは？

Amazon Elastic Block Store (Amazon EBS) は、Amazon Elastic Compute Cloud (Amazon EC2) インスタンスで使用できるスケーラブルな高性能ブロックストレージリソースを提供します。Amazon Elastic Block Store を使用すると、次のブロックストレージリソースを作成および管理できます。

- Amazon EBS ボリューム – Amazon EC2 インスタンスにアタッチするストレージボリュームです。ボリュームをインスタンスにアタッチすると、ファイルの保存やアプリケーションのインストールなど、コンピュータに接続されたローカルハードドライブを使用する場合と同じ方法でボリュームを使用できます。
- Amazon EBS スナップショット – ボリューム自体とは独立して持続する Amazon EBS ボリュームのポイントインタイムバックアップです。スナップショットを作成して、Amazon EBS ボリュームのデータをバックアップできます。その後、それらのスナップショットからいつでも新しいボリュームを復元できます。

トピック

- Amazon EBS の機能
- 関連サービス
- Amazon EBS へのアクセス
- 料金

Amazon EBS の機能

Amazon EBS は次の特徴と利点があります。

- 複数のボリュームタイプ – Amazon EBS では、幅広いアプリケーションのストレージパフォーマンスおよびコストを最適化できる複数のボリュームタイプが用意されています。ボリュームタイプは、トランザクションワークロード用の SSD-backed ストレージと、スループット集約型ワークロード用の HDD-backed ストレージの 2 つの主要なカテゴリに分かれています。
- スケーラビリティ – ニーズに合った容量およびパフォーマンスの仕様を備えた Amazon EBS ボリュームを作成できます。ニーズの変化に応じて、Elastic Volumes オペレーションを使用し、ダウンタイムなしで容量の動的な増加またはパフォーマンスの調整ができます。
- バックアップとリカバリ – Amazon EBS スナップショットを使用し、ボリュームに保存されているデータをバックアップします。その後、これらのスナップショットを使用して、ボリュームを

即座に復元したり、 AWS アカウント、 AWS リージョン、またはアベイラビリティーゾーン間でデータを移行したりできます。

- データ保護 – Amazon EBS 暗号化を使用し、Amazon EBS ボリュームおよび Amazon EBS スナップショットを暗号化します。暗号化オペレーションは、Amazon EC2 インスタンスをホストするサーバーで行われます。これにより、インスタンスとそのアタッチされたボリュームおよび後続のスナップショット間のdata-at-restdata-in-transitの両方のセキュリティが確保されます。
- データの可用性と耐久性 – io2 Block Express ボリュームは、99.999% の耐久性を持っており、年間故障率は 0.001% です。その他のボリュームタイプは 99.8% ~ 99.9% の耐久性を持っており、年間故障率は 0.1% ~ 0.2% です。さらに、ボリュームのデータは、1 つのアベイラビリティーゾーンの複数サーバー間で自動的にレプリケートされ、1 つのコンポーネントに障害が発生したときにデータが失われることを防ぎます。
- データアーカイブ — EBS スナップショットアーカイブは、EBS スナップショットの完全なpoint-in-timeコピーをアーカイブするための低コストのストレージ層を提供します。規制、コンプライアンス、または将来のプロジェクトリリースの目的でスナップショットを 90 日以上保持する必要がある場合は、アーカイブを使用します。

関連サービス

Amazon EBS は次のサービスと連携します。

- Amazon Elastic Compute Cloud — AWS クラウドで仮想マシン (Amazon EC2 インスタンス) を起動および管理できるサービス。EBS ボリュームをそれらのインスタンスにアタッチし、たとえば、ファイルの保存やアプリケーションのインストールなど、ローカルハードドライブを使用する場合と同じ方法で使用できます。詳細については、「Amazon EC2 とは」を参照してください。
- AWS Key Management Service – 暗号化キーを作成および管理をできるようにするマネージドサービス。 AWS KMS 暗号化キーを使用して、Amazon EBS ボリュームと Amazon EBS スナップショットに保存されているデータを暗号化できます。詳細については、「How Amazon EBS uses AWS KMS」を参照してください。
- Amazon Data Lifecycle Manager – EBS スナップショットおよび EBS-backed AMI の作成、保持、削除を自動化するマネージドサービス。Amazon Data Lifecycle Manager を使用し、Amazon EBS ボリュームおよび Amazon EC2 インスタンスのバックアップを自動化できます。詳細については、「Amazon Data Lifecycle Manager でバックアップを自動化」を参照してください。
- EBS direct APIs – EBS スナップショットの作成、スナップショットへデータの直接書き込み、スナップショットのデータの読み取り、2 つのスナップショット間の違いや変更の特定をできるよう

にするサービス。詳細については、「EBS direct API を使用して EBS スナップショットの内容にアクセスする」を参照してください。

- ごみ箱 – 誤って削除した EBS スナップショットおよび EBS バック AMI を復元できるようにするデータリカバリサービス。詳細については、「ごみ箱」を参照してください。

Amazon EBS へのアクセス

次のインターフェイスを使用して Amazon EBS リソースの作成および管理を行うことができます。

Amazon EC2 コンソール

ボリュームおよびスナップショットを作成および管理するウェブインターフェイス。 AWS アカウントにサインアップしている場合は、https://console.aws.amazon.com/ec2/ Amazon EC2 コンソールにアクセスできます。

AWS Command Line Interface

コマンドラインシェルのコマンドを使用して Amazon EBS リソースを管理できるようにするコマンドラインツール。Windows、Mac、Linux でサポートされています。詳細については、「AWS Command Line Interface ユーザーガイド」および ec2 コマンドを参照してください。

AWS Tools for PowerShell

PowerShell のコマンドラインから Amazon EBS リソースの操作をスクリプト処理できるようにする一連の PowerShell モジュール。詳細については、「AWS Tools for PowerShell ユーザーガイド」および「AWS Tools for PowerShell コマンドレットリファレンス」を参照してください。

CloudFormation

AWS リソースを記述する再利用可能な JSON または YAML テンプレートを作成し、それらのリソースをプロビジョニングして設定できるフルマネージド AWS サービス。詳細については、「AWS CloudFormation ユーザーガイド」を参照してください。

Amazon EC2 クエリ API

Amazon EC2 クエリ API は、HTTP 動詞 `GET` または `POST` を使用する HTTP または HTTPS リクエスト、ならびに `Action` という名前のクエリパラメータを提供します。詳細については、「Amazon EC2 API リファレンス」を参照してください。

AWS SDKs

AWS サービスと統合されたアプリケーションを構築できる言語固有の APIs。 AWS SDKsは多くの一般的なプログラミング言語で使用できます。詳細については、「構築するツール AWS」を参照してください。

料金

Amazon EBS については、お客様が利用されたプロビジョンの分のみのお支払いとなります。詳細については、Amazon EBS の料金表を参照してください。

Amazon EBS のセットアップ

このセクションのタスクを完了して、Amazon EBS リソースを使用するためのセットアップを行います。

タスク

- にサインアップする AWS アカウント
- (オプション) Amazon EBS 暗号化用のカスタマーマネージドキーを作成および使用する
- (オプション) Amazon EBS スナップショットのブロックパブリックアクセスを有効にする

にサインアップする AWS アカウント

の使用を開始するには AWS、 が必要です AWS アカウント。の作成の詳細については AWS アカウント、「 AWS アカウント管理 リファレンスガイド」の「 の開始方法 AWS アカウント」を参照してください。

(オプション) Amazon EBS 暗号化用のカスタマーマネージドキーを作成および使用する

Amazon EBS 暗号化は、 AWS KMS 暗号化キーを使用して Amazon EBS ボリュームと Amazon EBS スナップショットを暗号化する暗号化ソリューションです。Amazon EBS は、各リージョンで Amazon EBS 暗号化用の一意の AWS マネージド KMS キーを自動的に作成します。この KMS キーには aws/ebs エイリアスがあります。デフォルトの KMS キーをローテーションしたり、許可を管理したりすることはできません。Amazon EBS 暗号化に使用される KMS キーをより柔軟に制御するには、カスタマーマネージドキーの作成および使用を検討してください。

Amazon EBS 暗号化用のカスタマーマネージドキーの作成および使用方法

1. 対称暗号化 KMS キーを作成します。
2. Amazon EBS 暗号化のデフォルト KMS キーとして KMS キーを選択します。
3. Amazon EBS 暗号化 に KMS キー を使用する許可をユーザーに付与します。

(オプション) Amazon EBS スナップショットのブロックパブリックアクセスを有効にする

スナップショットがパブリックに共有されないようにするために、スナップショットのブロックパブリックアクセスを有効にします。リージョンでスナップショットのブロックパブリックアクセスを有効にすると、そのリージョンでスナップショットをパブリックに共有しようとする試みは自動的にブロックされます。これにより、スナップショットのセキュリティを強化し、スナップショットデータを不正アクセスや意図しないアクセスから保護することができます。

詳細については、「Amazon EBS スナップショットのブロックパブリックアクセス」を参照してください。

Console

スナップショットのパブリックアクセスのブロックを有効にする方法

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. ナビゲーションペインで [EC2 ダッシュボード] を選択し、[アカウントの属性] (右側) で [データ保護とセキュリティ] を選択します。
3. [EBS スナップショットのブロックパブリックアクセス] セクションで [管理] を選択します。
4. [パブリックアクセスをブロック] を選択し、次のオプションのいずれかを選択します。
   - [パブリックアクセスをすべてブロック] – スナップショットのパブリック共有をすべてブロックします。アカウント内のユーザーは、新しいパブリック共有をリクエストできません。また、既にパブリックに共有されていたスナップショットはプライベートとして扱われ、一般公開されなくなります。
   - [新しいパブリック共有をブロック] – スナップショットの新しいパブリック共有のみをブロックします。アカウント内のユーザーは、新しいパブリック共有をリクエストできません。ただし、既にパブリックに共有されていたスナップショットは、引き続き一般公開されます。
5. [更新] を選択します。

AWS CLI

スナップショットのパブリックアクセスのブロックを有効にする方法

enable-snapshot-block-public-access コマンドを使用します。--state に、次のいずれかの値を指定します。

- block-all-sharing – スナップショットのパブリック共有をすべてブロックします。アカウント内のユーザーは、新しいパブリック共有をリクエストできません。また、既にパブリックに共有されていたスナップショットはプライベートとして扱われ、一般公開されなくなります。
- block-new-sharing – スナップショットの新しいパブリック共有のみをブロックします。アカウント内のユーザーは、新しいパブリック共有をリクエストできません。ただし、既にパブリックに共有されていたスナップショットは、引き続き一般公開されます。

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

PowerShell

スナップショットのパブリックアクセスのブロックを有効にするには

Enable-EC2SnapshotBlockPublicAccess コマンドレットを使用します。-State に、次のいずれかの値を指定します。

- block-all-sharing – スナップショットのパブリック共有をすべてブロックします。アカウント内のユーザーは、新しいパブリック共有をリクエストできません。また、既にパブリックに共有されていたスナップショットはプライベートとして扱われ、一般公開されなくなります。
- block-new-sharing – スナップショットの新しいパブリック共有のみをブロックします。アカウント内のユーザーは、新しいパブリック共有をリクエストできません。ただし、既にパブリックに共有されていたスナップショットは、引き続き一般公開されます。

```
Enable-EC2SnapshotBlockPublicAccess -State block-new-sharing
```

Amazon EBS ボリューム

Amazon EBS ボリュームは、耐久性に優れたブロックレベルのストレージボリュームであり、インスタンスにアタッチできます。ボリュームをインスタンスにアタッチすると、他の物理ハードドライブと同じように使用できます。EBS ボリュームには柔軟性があります。現行世代のインスタンスタイプにアタッチされた現行世代のボリュームの場合、サイズの拡張、プロビジョンド IOPS の容量の変更、実稼働ボリュームのボリュームタイプの変更を動的に行うことができます。

EBS ボリュームは、インスタンス用のシステムドライブ、データベースアプリケーションのストレージなど、頻繁に更新する必要があるデータのプライマリストレージとして使用できます。連続ディスクスキャンを実行するスループットが高いアプリケーションにも使用できます。EBS ボリュームは、EC2 インスタンスの運用状況から独立した永続性を持ちます。

複数の EBS ボリュームを 1 つのインスタンスにアタッチできます。ボリュームとそのアタッチ先インスタンスは同じアベイラビリティーゾーンに存在している必要があります。ボリュームとインスタンスタイプによっては、マルチアタッチを使用してボリュームを複数のインスタンスに同時にマウントできます。

Amazon EBS には、次ののボリュームタイプが用意されています。汎用 SSD (gp2 および gp3)、プロビジョンド IOPS SSD (io1 および io2)、スループット最適化 HDD (st1)、Cold HDD (sc1)、磁気 (standard) です。それらはパフォーマンス特性と料金が異なるため、アプリケーションのニーズに応じてストレージのパフォーマンスとコストを調整できます。詳細については、「Amazon EBS ボリュームの種類」を参照してください。

アカウントには、利用できるストレージの合計に制限があります。これらの制限、および制限の引き上げをリクエストする方法についての詳細は、「Amazon EBS エンドポイントとクォータ」を参照してください。

マネージド EBS ボリュームは、Amazon EKS Auto Mode などのサービスプロバイダーによって管理されます。管理された EBS ボリュームの設定を直接変更することはできません。管理された EBS ボリュームは 管理されたフィールドの 正 値によって識別されます。詳細については、「Amazon EC2 マネージドインスタンス」を参照してください。

料金の詳細については、Amazon EBS 料金表を参照してください。

内容

- Amazon EBS ボリュームの特徴と利点
- Amazon EBS ボリュームの種類

- Amazon EBS ボリュームの制約
- Amazon EBS ボリュームと NVMe
- Amazon EBS ボリュームのライフサイクル
- スナップショットを使用した Amazon EBS ボリュームの置き換え
- Amazon EBS ボリュームステータスチェック
- Amazon EBS での障害テスト

Amazon EBS ボリュームの特徴と利点

EBS ボリュームには、インスタンスストアボリュームにはない利点があります。

利点

- データの可用性
- データの永続性
- データの暗号化
- データセキュリティ
- スナップショット
- 柔軟性

データの可用性

EBS ボリュームを作成すると、そのボリュームは同じアベイラビリティーゾーン内で自動的にレプリケートされます。これは、1 つのハードウェアコンポーネントの障害が原因でデータが失われることを防ぐためです。EBS ボリュームは、同じアベイラビリティーゾーン内の任意の EC2 インスタンスにアタッチできます。アタッチしたボリュームは、ハードドライブや他の物理デバイスと同じようなネイティブブロックとして表示されます。その時点で、インスタンスはローカルドライブと同じようにボリュームとやり取りできます。インスタンスに接続してファイルシステム (Linux インスタンスの `Ext4` または Windows インスタンスの NTFS) を持った EBS ボリュームをフォーマットし、アプリケーションをインストールできます。

指定したデバイスに複数のボリュームをアタッチする場合は、ボリュームにまたがってデータをストライプすることで I/O とスループットのパフォーマンスを向上させることができます。

`io1` および `io2` の EBS ボリュームを、最大 16 個の Nitro ベースのインスタンスにアタッチできます。詳細については、マルチアタッチを使用して EBS ボリュームを複数の EC2 インスタンスへア

タッチを参照してください。それ以外の場合は、EBS ボリュームを 1 つのインスタンスにアタッチできます。

EBS ボリュームのモニタリングデータは無料で取得できます (EBS-backed インスタンスのルートデバイスボリュームのデータも含まれます)。メトリクスのモニタリングの詳細については、Amazon EBS の Amazon CloudWatch メトリクスを参照してください。ボリュームのステータスの追跡の詳細については、Amazon EBS 用 Amazon EventBridge イベントを参照してください。

データの永続性

EBS ボリュームは、インスタンスの運用状況に左右されない永続性のあるストレージを提供します。データが維持される限り、ボリュームの使用料が発生します。

EC2 コンソール上で使用する EBS ボリュームを設定するときに [Delete on Termination (終了時に削除)] チェックボックスをオフにした場合、実行中のインスタンスにアタッチされている EBS ボリュームを、インスタンスの終了時にデータがそのままの状態でインスタンスから自動的にデタッチすることができます。デタッチされたボリュームは新しいインスタンスに再アタッチできるので、迅速な復旧が可能です。[Delete on Termination (終了時に削除)] のチェックボックスがオンの場合、ボリュームは EC2 インスタンスの終了後に削除されます。EBS-backed インスタンスを使用している場合は、アタッチしたボリュームに格納されているデータに影響を与えることなく、インスタンスを停止および再起動できます。ボリュームは停止/起動のサイクルを通じてアタッチされたままです。これにより、必要なときに処理リソースとストレージリソースを使用するだけで、ボリュームでのデータの処理と格納を永続的に実行できるようになります。データは、ボリュームを明示的に削除するまでボリュームに保持されます。削除した EBS ボリュームが使用していた物理的なブロックストレージは、別のアカウントに再割り当てされる前に、ゼロで上書きされます。機密データを扱っている場合は、手動によるデータの暗号化や、Amazon EBS 暗号化 で保護されているボリュームへのデータの格納を検討してください。詳細については、「Amazon EBS 暗号化」を参照してください。

デフォルトでは、インスタンスの起動時に作成およびアタッチされた ルート EBS ボリュームは、インスタンスの終了時に削除されます。この動作を変更するには、インスタンスの起動時にフラグ `DeleteOnTermination` の値を `false` に変更します。値を変更すると、インスタンスが終了してもボリュームが保持されるので、そのボリュームを別のインスタンスにアタッチできます。

デフォルトでは、インスタンスの起動時に作成およびアタッチされた 追加の EBS ボリュームは、インスタンスの終了時に削除されません。この動作を変更するには、インスタンスの起動時にフラグ `DeleteOnTermination` の値を `true` に変更します。値の変更により、ボリュームはインスタンスの終了時に削除されます。

データの暗号化

簡素化されたデータの暗号化を使用するには、Amazon EBS 暗号化 機能を使用して、暗号化の対象となる EBS ボリュームを作成できます。暗号化は、すべての EBS ボリュームタイプでサポートされています。暗号化された EBS ボリュームを使用することで、規制/監査されるデータとアプリケーションの保管時の広範な暗号化要件に対応できます。Amazon EBS 暗号化では、256 ビットの Advanced Encryption Standard アルゴリズム (AES-256) と Amazon が管理するキーインフラストラクチャが使用されます。暗号化は EC2 インスタンスをホストするサーバーで行われ、EC2 インスタンスから Amazon EBS ストレージに転送されるデータが暗号化されます。詳細については、「Amazon EBS 暗号化」を参照してください。

Amazon EBS 暗号化は、暗号化されたボリュームと、暗号化されたボリュームから作成されたスナップショットを作成する AWS KMS keys ときに を使用します。リージョンで暗号化された EBS ボリュームを初めて作成すると、デフォルトの AWS マネージド KMS キーが自動的に作成されます。このキーは、ユーザーがカスタマーマネージドキーを作成して使用しない限り、Amazon EBS 暗号化に使用されます。独自のカスタマーマネージドキーを作成すると、アクセスコントロールを作成、ローテーション、無効化、定義できるほか、データの保護に使用される暗号化キーを監査できるなど、より高い柔軟性が得られます。詳細については、AWS Key Management Service デベロッパーガイドを参照してください。

データセキュリティ

Amazon EBS ボリュームは、初期化されていない raw ブロックデバイスとして表示されます。これらのデバイスは、EBS インフラストラクチャ上に作成される論理デバイスであり、Amazon EBS サービスは、お客様による利用または再利用の前に、デバイスが論理的に空になっている (つまり、raw ブロックがゼロになっている、または暗号で擬似ランダムデータが含まれている) ようにします。

DoD 5220.22-M (National Industrial Security Program Operating Manual) や NIST 800-88 (Guidelines for Media Sanitization) に詳述されているような、使用後もしくは使用前 (またはその両方) に特定の方法を使用してすべてのデータを消去する必要がある手順がある場合、Amazon EBS でこれを行うことができます。ブロックレベルのアクティビティは、Amazon EBS サービス内の基盤となるストレージメディアに反映されます。

スナップショット

Amazon EBS は、Amazon S3 ボリュームのスナップショット (バックアップ) を作成し、ボリューム内のデータのコピーを EBS に書き込む機能を備えています。そこで、データは複数のアベイラビリ

ティーゾーンに冗長的に保存されます。スナップショットを作成するために、対象のボリュームが実行中のインスタンスにアタッチされている必要はありません。ボリュームにデータを書き込み続けながら、そのボリュームのスナップショットを定期的に作成して、新しいボリュームのベースラインとして使用できます。このスナップショットは、新しい EBS ボリュームを複数作成したり、アベイラビリティーゾーン間でボリュームを移動したりするときに使用できます。暗号化された EBS ボリュームのスナップショットは自動的に暗号化されます。

スナップショットから新規ボリュームを作成する場合、このボリュームはスナップショット作成時における元のボリュームの正確なコピーになります。暗号化されたスナップショットから作成された EBS ボリュームは、自動的に暗号化されます。別のアベイラビリティーゾーンを指定し、この機能を使用してそのゾーンにボリュームを複製することもできます。スナップショットは、特定の AWS アカウントと共有することも、公開することもできます。スナップショットを作成すると、ソースボリュームのサイズではなく、バックアップされるサイズに基づいて、Amazon S3 で料金が発生します。同じボリュームの後続スナップショットは、増分スナップショットです。このスナップショットには、前回のスナップショット作成以降にボリュームに書き込まれた変更データと新規データのみが含まれ、この変更データと新規データに対してのみ料金が発生します。

スナップショットは増分バックアップです。つまり、最後にスナップショットを作成した時点から、ボリューム上で変更のあるブロックだけが保存されます。例えば、100 GiB のデータが格納されているボリュームがあるとします。最後にスナップショットを作成してから、そのうちの 5 GiB 分のデータしか変更されていない場合は、その変更された 5 GiB のデータだけが Amazon S3に書き込まれます。スナップショットの保存は増分ベースで行われるものの、スナップショット削除プロセスは最新のスナップショットのみ保持するように設計されています。

ボリュームとスナップショットを分類および管理しやすくするため、任意のメタデータでタグ付けすることができます。

ボリュームを自動的にバックアップするには、Amazon Data Lifecycle Manager または AWS Backup を使用できます。

柔軟性

EBS ボリュームは、実稼働環境での設定変更をサポートします。サービスを中断せずに、ボリュームタイプ、ボリュームサイズ、IOPS 容量を変更できます。詳細については、Elastic Volumes オペレーションを使用して Amazon EBS ボリュームを変更するを参照してください。

Amazon EBS ボリュームの種類

Amazon EBS では以下のボリュームタイプを提供しており、これらはパフォーマンス特性と料金が異なるため、アプリケーションのニーズに応じてストレージのパフォーマンスとコストを調整できます。

> **Important**
>
> インスタンスの構成、I/O 特性、ワークロードのデマンドなど、EBS ボリュームのパフォーマンスに影響を与える可能性がある要因は複数存在します。EBS ボリュームにプロビジョニングされた IOPS を最大限に活用するには、EBS に最適化されたインスタンスを使用します。EBS ボリュームを最大限活用するための詳細については、Amazon EBS ボリュームパフォーマンスを参照してください。

料金の詳細については、Amazon EBS 料金表を参照してください。

ボリュームの種類

- ソリッドステートドライブ (SSD) ボリューム
- ハードディスクドライブ (HDD) ボリューム
- 旧世代のボリューム

ソリッドステートドライブ (SSD) ボリューム

SSD-backed のボリュームは、主要なパフォーマンス属性は IOPS である I/O サイズの小さい頻繁な読み取り/書き込み操作を伴うトランザクションワークロード用に最適化されています。SSD-backed のボリュームタイプには、汎用 SSD とプロビジョンド IOPS SSD があります。SSD-Backed ボリュームの使用例と特性の概要を次に示します。

	Amazon EBS 汎用 SSD ボリューム		Amazon EBS プロビジョンド IOPS SSD ボリューム	
ボリュームタイプ	gp3 [6]	gp2	io2 Block Express	io1

	Amazon EBS 汎用 SSD ボリューム		Amazon EBS プロビジョンド IOPS SSD ボリューム	
耐久性	99.8% ~ 99.9% の耐久性 (0.1% ~ 0.2% の年間故障率)		99.999% の耐久性 (0.001% の年間故障率)	99.8% ~ 99.9% の耐久性 (0.1% ~ 0.2% の年間故障率)
ユースケース	• トランザクションワークロード • 仮想デスクトップ • 中規模の単一インスタンスデータベース • 低レイテンシーのインタラクティブなアプリケーション • ブートボリューム • 開発・テスト環境		必要なワークロード • 平均レイテンシーが 500 マイクロ秒未満の一貫してミリ秒未満のレイテンシー [5] • 持続的な IOPS パフォーマンス • 80,000 IOPS または 2,000 MiB/秒を超えるスループット	• 持続的な IOPS パフォーマンスまたは 16,000 IOPS 以上のパフォーマンスを必要とするワークロード • I/O 集約型のデータベースワークロード
ボリュームサイズ	1 GiB ~ 64 TiB	1GiB - 16TiB	4 GiB ~ 64 TiB	4 GiB ~ 16 TiB
最大 IOPS	80,000 [3] (25.6 KiB I/O [4])	16,000 (16 KiB I/O [4])	256,000 [3] (16 KiB I/O [4])	64,000 (16 KiB I/O [4])
最大スループット	2,000 MiB/秒	250 MiB/秒 [1]	4,000 MiB/秒	1,000 MiB/秒 [2]

	Amazon EBS 汎用 SSD ボリューム	Amazon EBS プロビジョンド IOPS SSD ボリューム	
Amazon EBS マルチアタッチ	サポート外	サポート	
NVMe 予約	サポート外	サポート対象	サポート
ブートボリューム	サポート		

[1] スループットの制限は、ボリュームサイズに応じて 128 MiB/秒〜250 MiB/秒です。詳細については、「gp2 ボリュームのパフォーマンス」を参照してください。2018 年 12 月 3 日以前に作成され、作成後に変更されていないボリュームの場合は、そのボリュームを変更しない限り、完全なパフォーマンスには到達しない可能性があります。

[2] 1,000 MiB/秒の最大スループットを実現するには、ボリュームを 64,000 IOPS でプロビジョニングし、Nitro ベースのインスタンスにアタッチする必要があります。2017 年 12 月 6 日以前に作成され、作成後に変更されていないボリュームの場合は、そのボリュームを変更しない限り、完全なパフォーマンスには到達しない可能性があります。

[3] Nitro ベースのインスタンスは、最大 256,000 IOPS でプロビジョニングされたボリュームをサポートします。他のインスタンスタイプは、最大 64,000 IOPS までででプロビジョニングされたボリュームにアタッチできますが、最大 32,000 IOPS までプロビジョニングできます。

[4] ボリュームのスループット上限内で最大 IOPS に達するために必要な I/O サイズを表します。

[5] io2 Block Express ボリュームは、16 KiB の I/O オペレーションで 500 マイクロ秒未満の平均レイテンシーを実現するように設計されています。

[6] Outposts では、gp3 ボリュームは最大 16 TiB のサイズ、最大 16,000 の IOPS、最大 1,000 MiB/秒のスループットをサポートします。

SSD-backed のボリュームタイプの詳細については、以下を参照してください。

- Amazon EBS 汎用 SSD ボリューム
- Amazon EBS プロビジョンド IOPS SSD ボリューム

ハードディスクドライブ (HDD) ボリューム

HDD-backed のボリュームはパフォーマンスの主要な属性がスループットである大規模なストリーミングワークロード用に最適化されています。HDD ボリュームタイプには、スループット最適化 HDD と Cold HDD があります。以下は、HDD-Backed ボリュームのユースケースと特性の概要です。

	スループット最適化 HDD ボリューム	Cold HDD ボリューム
ボリュームタイプ	st1	sc1
耐久性	99.8% ~ 99.9% の耐久性 (0.1% ~ 0.2% の年間故障率)	
ユースケース	• ビッグデータ • データウェアハウス • ログ処理	• アクセス頻度の低いデータ用のスループット指向ストレージ • 低いストレージコストが重視されるシナリオ
ボリュームサイズ	125 GiB ~ 16 TiB	
ボリュームあたりの最大 IOPS (1 MiB I/O)	500	250
ボリュームあたりの最大スループット	500 MiB/秒	250 MiB/秒
Amazon EBS マルチアタッチ	サポートされていません	
ブートボリューム	サポートされていません	

ハードディスクドライブ (HDD) ボリュームの詳細については、「Amazon EBS スループット最適化 HDD ボリュームと Cold HDD ボリューム」を参照してください。

旧世代のボリューム

マグネティック (standard) ボリュームは、磁気ドライブによってバックアップされた旧世代のボリュームです。それらは、データへのアクセス頻度が低く、パフォーマンスが最も重要ではない小規模なデータセットを持つワークロードに適しています。これらのボリュームは、平均約 100 IOPS を実現し、バースト能力は最大約数百 IOPS です。ボリュームのサイズは 1 GiB ~ 1 TiB です。

> **Tip**
>
> 磁気ボリュームは、旧世代のボリュームタイプです。旧世代のボリュームより高いパフォーマンスまたはパフォーマンスの整合性が必要であれば、現行世代のボリュームタイプの使用を検討するようお勧めします。

次の表は、旧世代の EBS ボリュームタイプを示しています。

	マグネティック
ボリュームタイプ	`standard`
ユースケース	データへのアクセス頻度が低いワークロード
ボリュームサイズ	1 GiB ~ 1 TiB
ボリュームあたりの最大 IOPS	40 ~ 200
ボリュームあたりの最大スループット	40 ~ 90 MiB/秒
ブートボリューム	サポート

Amazon EBS 汎用 SSD ボリューム

汎用 SSD (gp2 および gp3) ボリュームは、ソリッドステートドライブ (SSD) によってサポートされます。これらは、さまざまなトランザクションワークロードに対して、料金とパフォーマンスのバランスをとります。これらには、仮想デスクトップ、中規模のシングルインスタンスデータベース、レイテンシーの影響を受けやすいインタラクティブなアプリケーション、開発およびテスト環境、およ

びブートボリュームが含まれます。これらのボリュームは、ほとんどのワークロードに推奨されます。

Amazon EBS は、次のタイプの汎用 SSD ボリュームを提供します。

型

- 汎用 SSD (gp3) ボリューム
- 汎用 SSD (gp2) ボリューム

汎用 SSD (gp3) ボリューム

汎用 SSD (gp3) ボリュームは、最新世代の汎用 SSD ボリュームであり、Amazon EBS が提供する最も低コストの SSD ボリュームです。このボリュームタイプは、ほとんどの用途で料金とパフォーマンスの適切なバランスを提供するのに役立ちます。また、ボリュームサイズにかかわらず、ボリュームのパフォーマンスをスケールするのにも役立ちます。つまり、追加のブロックストレージ容量をプロビジョニングしなくても、必要なパフォーマンスをプロビジョニングできます。さらに、gp3 ボリュームでは、GiB あたりの料金が汎用 SSD (gp2) ボリュームよりも 20% 低くなります。

gp3 ボリュームは、1 桁ミリ秒のレイテンシーと 99.8% から 99.9% のボリューム耐久性を提供し、年間障害率 (AFR) は 0.2% 以下です。これにより、1 年間に実行中のボリューム 1,000 個あたり最大 2 つのボリューム障害が発生します。 は gp3 ボリュームを AWS 設計して、プロビジョニングされたパフォーマンスを 99% 実現します。

> **Tip**
>
> レイテンシーの影響を受けやすいワークロードの場合は、io2 Block Express ボリュームを使用することをお勧めします。io2Block Express ボリュームは、16 KiB の I/O オペレーションで 500 マイクロ秒未満の平均レイテンシーを実現するように設計されています。io2Block Express ボリュームでは、汎用ボリュームと比較して外れ値のレイテンシーも改善され、800 マイクロ秒を超える I/O の頻度が 10 分の 1 以下に減少します。詳細については、「プロビジョンド IOPS SSD (io2) Block Express ボリューム」を参照してください。

内容

- gp3 ボリュームのパフォーマンス

- gp3 ボリュームサイズ
- gp2 から gp3 に移行する

gp3 ボリュームのパフォーマンス

> **Tip**
>
> gp3 ボリュームはバーストパフォーマンスを使用しません。これらは、フルプロビジョンド IOPS とスループットパフォーマンスを無期限に維持できます。

IOPS パフォーマンス

gp3 ボリュームは、ストレージの料金に含まれている 3,000 IOPS の一貫したベースライン IOPS パフォーマンスを提供します。追加料金を支払うことで、ボリュームサイズの GiB あたり 500 IOPS の割合で追加の IOPS (最大 80,000 まで) をプロビジョニングできます。最大 IOPS は、160 GiB 以上のボリューム (500 IOPS/GiB × 160 GiB = 80,000 IOPS) に対してプロビジョニングできます。

スループットパフォーマンス

gp3 ボリュームは、ストレージの料金に含まれている 125 MiB/秒の一貫したベースラインスループットパフォーマンスを提供します。追加料金を支払うことで、プロビジョンド IOPS あたり 0.25 MiB/秒の割合で追加のスループット (最大 2,000 MiB/秒) をプロビジョニングできます。最大スループットは、8,000 IOPS 以上かつ 16 GiB 以上 (8,000 IOPS × IOPS あたり 0.25 MiB/秒 = 2,000 MiB/秒) でプロビジョニングできます。

> **Note**
>
> Outposts では、gp3 ボリュームは最大 16 TiB のサイズ、最大 16,000 の IOPS、最大 1,000 MiB/秒のスループットをサポートします。

gp3 ボリュームサイズ

gp3 ボリュームのサイズは 1 GiB ~ 64 TiB です。

gp2 から gp3 に移行する

現在 gp2 ボリュームを使用している場合は、Elastic Volumes オペレーションを使用して Amazon EBS ボリュームを変更する オペレーションを使用してボリュームを gp3 に移行できます。Amazon

EBS Elastic Volumes オペレーションを使用して、Amazon EC2 インスタンスを中断することなく、既存のボリュームのボリュームタイプ、IOPS、およびスループットを変更できます。コンソールを使用してボリュームを作成したり、スナップショットから AMI を作成したりする場合、ボリュームタイプには、汎用 SSD gp3 がデフォルトで選択されます。それ以外の場合は、gp2 がデフォルトで選択されます。このような場合、gp2 を使用する代わりに、ボリュームタイプとして gp3 を選択できます。

gp2 ボリュームを gp3 に移行することでどの程度のコストを削減できるかを調べるには、Amazon EBS gp2 から gp3 への移行で削減できるコストの計算ツールを使用してください。

汎用 SSD (gp2) ボリューム

これらは、さまざまなトランザクションワークロードに対応できるコスト効率の高いストレージとして使用できます。gp2 ボリュームを使用すると、ボリュームサイズに応じてパフォーマンスがスケールします。

> **Tip**
>
> gp3 ボリュームは、汎用 SSD ボリュームの最新世代です。このボリュームは、より予測可能なパフォーマンススケーリングと、gp2 ボリュームよりも最大 20% 低い料金を提供します。詳細については、「汎用 SSD (gp3) ボリューム」を参照してください。
> gp2 ボリュームを gp3 に移行することでどの程度のコストを削減できるかを調べるには、Amazon EBS gp2 から gp3 への移行で削減できるコストの計算ツールを使用してください。

gp2 ボリュームは、1 桁ミリ秒のレイテンシーと 99.8% から 99.9% のボリューム耐久性を提供し、年間障害率 (AFR) は 0.2% 以下です。これは、1 年間で実行中のボリューム 1,000 個あたり最大 2 つのボリューム障害に変換されます。 は、プロビジョニングされたパフォーマンスを 99% の割合で実現するようにgp2ボリューム AWS を設計します。

内容

- gp2 ボリュームのパフォーマンス
- gp2 ボリュームサイズ

gp2 ボリュームのパフォーマンス

IOPS パフォーマンス

ベースライン IOPS パフォーマンスは、最小 100 から最大 16,000 の間で、ボリュームサイズの GiB あたり 3 IOPS のレートで直線的にスケールします。IOPS パフォーマンスは次のようにプロビジョニングされます。

- 33.33 GiB 以下のボリュームは、最小 100 IOPS でプロビジョニングされます。
- 33.33 GiB を超えるボリュームは、最大 16,000 IOPS (5,334 GiB (3 X 5,334) で到達) まで、ボリュームサイズの GiB あたり 3 IOPS でプロビジョニングされます。
- 5,334 GiB 以上のボリュームは、16,000 IOPS でプロビジョニングされます。

1 TiB 未満の (および 3,000 IOPS 未満でプロビジョニングされた) gp2 ボリュームは、長期間にわたって必要な場合に 3,000 IOPS にバーストできます。ボリュームのバースト機能は I/O クレジットによって制御されます。I/O の需要がベースラインパフォーマンスよりも大きい場合、ボリュームは I/O クレジットを消費して、必要なパフォーマンスレベル (最大 3,000 IOPS) までバーストします。バースト中は、I/O クレジットは累積されず、ベースライン IOPS を上回る IOPS (使用率 = バースト IOPS - ベースライン IOPS) を上回る IOPS の割合で消費されます。ボリュームが蓄積した I/O クレジットが多いほど、バーストパフォーマンスを維持できる時間が長くなります。次のようにバースト期間を計算できます。

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

I/O の需要がベースラインパフォーマンスレベル以下に低下すると、ボリュームは、ボリュームサイズの GiB あたり 3 I/O クレジット/秒のレートで I/O クレジットを獲得し始めます。ボリュームには 540 万 I/O クレジットの I/O クレジットの累積制限があり、これは少なくとも 30 分間で 3,000 IOPS の最大バーストパフォーマンスを維持するのに十分です。

> **Note**
>
> 各ボリュームは、540 万 I/O クレジットの初期 I/O クレジットバランスを受け取ります。これにより、ブートボリュームの高速な初期ブートサイクルと、他のアプリケーションの優れたブートストラップエクスペリエンスが提供されます。

ボリュームサイズとボリュームの関連するベースラインパフォーマンス、バースト期間 (540 万 I/O クレジットから開始)、および空の I/O クレジットバランスを再補充するのにかかる時間の例を次の表に示します。

ボリュームサイズ (GiB)	ベースラインパフォーマンス (IOPS)	3000 IOPS におけるバースト期間 (秒)	空のクレジットバランスを補充する時間 (秒)
1 ~ 33.33	100	1,862	54,000
100	300	2,000	18,000
334 (最大スループットの最小サイズ)	1,002	2,703	5,389
750	2,250	7,200	2,400
1,000	3,000	該当なし*	該当なし*
5,334 (最大 IOPS の最小サイズ) 以上	16,000	該当なし*	該当なし*

* ボリュームのベースラインパフォーマンスが最大バーストパフォーマンスを超えた場合。

Amazon CloudWatch の Amazon EBS `BurstBalance` メトリクスを使用して、ボリュームの I/O クレジットバランスをモニタリングできます。このメトリクスは、残りの gp2 の I/O クレジットの割合を示します。詳細については、「Amazon EBS I/O の特性およびモニタリング」を参照してください。`BurstBalance` の値が一定のレベルまで下がったときに通知するアラームを設定できます。詳細については、「CloudWatch アラームを作成する」を参照してください。

スループットパフォーマンス

gp2 ボリュームは、ボリュームサイズに応じて、128 MiB/秒 ~ 250 MiB/秒のスループットを提供します。スループットパフォーマンスは次のようにプロビジョニングされます。

- 170 GiB 以下のボリュームは、最大スループット 128 MiB/秒を提供します。
- 170 GiB より大きく 334 GiB より小さいボリュームは、250 MiB/秒の最大スループットまでバーストできます。
- 334 GiB 以上のボリュームは、250 MiB/秒を提供します。

gp2 ボリュームのスループットは、250 MiB/秒のスループット制限まで、次の計算式を使用して計算できます。

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

gp2 ボリュームサイズ

gp2 ボリュームのサイズ範囲は、1 GiB ~ 16 TiB です。ボリュームのパフォーマンスはボリュームサイズに比例してスケールすることに注意してください。

Amazon EBS プロビジョンド IOPS SSD ボリューム

プロビジョンド IOPS SSD ボリュームは、ソリッドステートドライブ (SSD) によってサポートされます。これらは、IOPS が高く、スループットが大量で、低レイテンシーを必要とする、重要なワークロード向けに設計された、最高パフォーマンスの Amazon EBS ストレージボリュームです。プロビジョンド IOPS SSD ボリュームは、期間の 99.9 パーセントにわたり、プロビジョニングされた IOPS パフォーマンスを実現します。

Amazon EBS は、2 種類のプロビジョンド IOPS SSD ボリュームを提供します。

- プロビジョンド IOPS SSD (io2) Block Express ボリューム
- Provisioned IOPS SSD (io1) ボリューム

プロビジョンド IOPS SSD (**io2**) Block Express ボリューム

io2 Block Express ボリュームは、次世代の Amazon EBS ストレージサーバーアーキテクチャにビルドされます。Nitro System 上に構築されインスタンスで実行される、最も要求の厳しい I/O 集約型アプリケーションのパフォーマンス要件を満たす目的として構築されています。最高級の耐久性と最小限のレイテンシーを備えた Block Express は、Oracle、SAP HANA、Microsoft SQL Server、SAS Analytics など、パフォーマンス重視のミッションクリティカルなワークロードの実行に最適です。

Block Express アーキテクチャにより、io2 ボリュームのパフォーマンスとスケールが向上します。Block Express サーバーは、Scalable Reliable Datagram (SRD) ネットワークプロトコルを使用して、Nitro ベースのインスタンスと通信します。このインターフェイスは、インスタンスのホストハードウェア上の Amazon EBS I/O 機能専用の Nitro Card に実装されます。I/O 遅延とレイテンシーのバラツキ (ネットワークジッター) を最小限に抑え、より高速で安定したパフォーマンスをアプリケーションに提供します。

io2 Block Express ボリュームは、年間故障率 (AFR) が 0.001% 以下で 99.999% のボリューム耐久性を提供するように設計されています。これは、1 年間で実行中のボリューム 100,000 個あたりにつ

き 1 つのボリュームの故障に相当します。io2Block Express ボリュームは、一貫してミリ秒未満のレイテンシーを実現する、単一ボリュームの恩恵を受けるワークロードに適しており、gp3 ボリュームより高い IOPS およびスループットをサポートします。

Nitro ベースのインスタンスにアタッチする場合、io2 Block Express ボリュームは、16 KiB の I/O オペレーションで 500 マイクロ秒未満の平均レイテンシーを実現するように設計されています。io2Block Express ボリュームでは、汎用ボリュームと比較して外れ値のレイテンシーも改善され、800 マイクロ秒を超える I/O の頻度が 10 分の 1 以下に減少します。

トピック

- 考慮事項
- パフォーマンス

考慮事項

- io2 Block Express ボリュームは、 AWS リージョン AWS GovCloud (US) と中国リージョンを含むすべての リージョンで使用できます。
- 2025 年 4 月 30 日現在、新規および以前に作成されたすべての io2 ボリュームは io2 Block Express ボリュームです。
- Nitro ベースのインスタンスは、最大 256,000 IOPS でプロビジョニングされたボリュームをサポートします。他のインスタンスタイプは、最大 64,000 IOPS までででプロビジョニングされたボリュームにアタッチできますが、最大 32,000 IOPS までプロビジョニングできます。

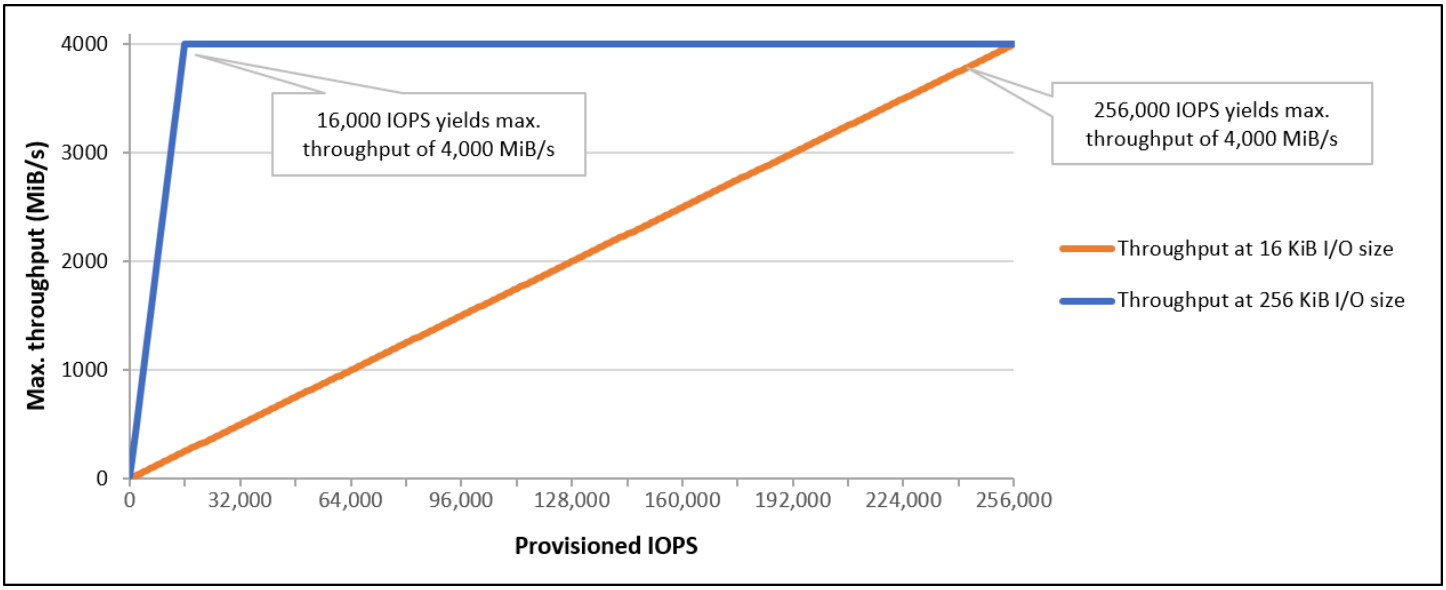
パフォーマンス

io2 Block Express ボリュームには、次の特性があります。

- 16 KiB の I/O サイズで 500 マイクロ秒未満の平均レイテンシー。汎用ボリュームと比較して外れ値のレイテンシーが改善し、800 マイクロ秒を超える I/O の頻度が 10 分の 1 以下に減少します。
- 最大 64 TiB (65,536 GiB) までのストレージ容量。
- IOPS:GiB 比は 1,000:1 の、最大 256,000 のプロビジョンド IOPS。最大 IOPS は、256 GiB 以上でプロビジョニングできます (1,000 IOPS × 256 GiB = 256,000 IOPS)。

> **Note**
>
> Nitro ベースのインスタンスで最大 256,000 IOPS を実現できます。他のインスタンスでは、最大 32,000 IOPS を実現できます。

- 最大 4,000 MiB/秒のボリュームスループット。スループットはプロビジョンド IOPS ごとに 0.256 MiB/秒の割合で比例してスケールします。16,000 IOPS 以上で最大スループットに達します。



Provisioned IOPS SSD (**io1**) ボリューム

プロビジョンド IOPS SSD (io1) ボリュームは、ランダムアクセス I/O スループットにおけるストレージパフォーマンスと整合性が重要な、I/O 集約型ワークロード (特にデータベースワークロード) のニーズを満たすように設計されています。プロビジョンド IOPS SSD ボリュームでは、ボリュームの作成時に指定した、一貫性のある IOPS レートを使用します。Amazon EBS では、プロビジョニングされたパフォーマンスを 99.9% 提供します。

io1 ボリュームは、1 桁ミリ秒のレイテンシーと 99.8～99.9% のボリューム耐久性を提供し、年間故障率 (AFR) は 0.2% 以下です。これは、1 年間の期間において故障するボリュームの数が、実行中の 1,000 個のボリュームあたり最大 2 個であることを意味します。

io1 ボリュームは、すべての Amazon EC2 インスタンスタイプで使用できます。

パフォーマンス

io1 ボリュームのサイズは 4 GiB ~ 16 TiB であり、ボリュームあたり 100 IOPS から最大 64,000 IOPS をプロビジョニングできます。リクエストされたボリュームサイズに対するプロビジョンド IOPS の最大割合 (GiB 単位) は 50:1 です。例えば、100 GiB の io1 ボリュームは最大 5,000 IOPS でプロビジョニングできます。

1,280 GiB 以上のボリュームに対して、最大 IOPS をプロビジョニングできます (50 × 1,280 GiB = 64,000 IOPS)。

- 最大 32,000 IOPS でプロビジョニングされた io1 ボリュームは、最大 256 KiB の I/O サイズをサポートし、最大 500 MiB/秒のスループットを生み出します。最大の I/O サイズでは、ピークのスループットが 2,000 IOPS に達します。
- 32,000 IOPS 超 (最大 64,000 IOPS) でプロビジョニングされた io1 ボリュームでは、プロビジョンド IOPS あたり 16 KiB のレートでスループットが直線的に増加します。例えば、48,000 IOPS でプロビジョニングされたボリュームは、最大 750 MiB/秒のスループット (プロビジョンド IOPS あたり 16 KiB × 48,000 プロビジョンド IOPS = 750 MiB/秒) をサポートできます。
- 1,000 MiB/秒の最大スループットを実現するには、ボリュームを 64,000 IOPS (プロビジョンド IOPS あたり 16 KiB × 64,000 プロビジョンド IOPS = 1,000 MiB/秒) でプロビジョニングする必要があります。
- Nitro ベースのインスタンスでのみ最大 64,000 IOPS を実現できます。他のインスタンスでは、最大 32,000 IOPS を実現できます。

次のグラフは、これらのパフォーマンスの特長を示しています。



発生する I/O あたりのレイテンシーは、プロビジョニングされる IOPS とワークロードプロファイルによって異なります。最適な I/O レイテンシーエクスペリエンスを得るには、ワークロードの I/O プロファイルに合わせて IOPS をプロビジョニングしてください。

Amazon EBS スループット最適化 HDD ボリュームと Cold HDD ボリューム

Amazon EBS によって提供される HDD-Backed ボリュームは、次のカテゴリに分類されます。

- スループット最適化 HDD: 高いスループットを必要とするアクセス頻度の高いワークロード向けの低コストの HDD
- Cold HDD: アクセス頻度の低いワークロード向けの最も低コストの HDD。

トピック

- インスタンスごとのスループット制限
- スループット最適化 HDD ボリューム
- Cold HDD ボリューム
- HDD ボリュームを使用するときのパフォーマンスに関する考慮事項
- ボリュームのバーストバケットバランスのモニタリング

インスタンスごとのスループット制限

`st1` ボリュームと `sc1` ボリュームのスループットは常に、次のいずれか小さい方によって決定されます。

- ボリュームのスループット制限
- インスタンスのスループット制限

ネットワークボトルネックを回避するには、すべての Amazon EBS ボリュームで、EBS 最適化 EC2 インスタンスを選択することをお勧めします。

スループット最適化 HDD ボリューム

スループット最適化 HDD (`st1`) ボリュームは、IOPS ではなくスループットでパフォーマンスを示す、低コストの磁気ストレージに使用できます。このボリュームタイプは、Amazon EMR、ETL、

データウェアハウス、ログ処理など、サイズの大きなシーケンシャルワークロードに適しています。ブート可能な st1 ボリュームはサポートされていません。

スループット最適化 HDD (st1) ボリュームは Cold HDD (sc1) ボリュームに類似していますが、アクセスが頻繁なデータをサポートするように設計されています。

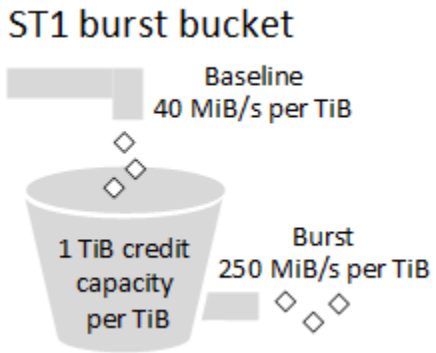
> **Note**
>
> このボリュームタイプは、サイズの大きなシーケンシャル I/O が含まれるワークロードに適しています。サイズの小さなランダム I/O を実行するワークロードのお客様には、Amazon EBS 汎用 SSD ボリューム または Amazon EBS プロビジョンド IOPS SSD ボリューム の使用をお勧めします。詳細については、「HDD に対する読み取り/書き込みサイズが小さい場合の非効率性」を参照してください。

スループット最適化 HDD (st1) ボリュームを EBS 最適化インスタンスにアタッチすると、一貫したパフォーマンスが維持され、1 年で 99% の期間、想定されるスループットパフォーマンスの少なくとも 90% のボリュームが提供されます。

スループットクレジットとバーストパフォーマンス

gp2 と同様、st1 でもパフォーマンスのためにバーストバケットモデルが使用されます。ボリュームのベースラインスループット (ボリュームのスループットクレジットが蓄積されるレート) は、ボリュームサイズによって決まります。ボリュームのバーストスループット (クレジットがある場合に可能な消費レート) もボリュームサイズによって決まります。ボリュームが大きいほど、ベースラインとバーストスループットの値も大きくなります。また、ボリュームのクレジットが多いほど、バーストレベルでドライブ I/O に使用できる時間が長くなります。

次の図は、st1 のバーストバケット動作を示しています。



スループットとスループットクレジットの上限により、st1 ボリュームで使用可能なスループットは、以下の計算式で示されます。

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

1 TiB の st1 ボリュームの場合、バーストスループットは 250 MiB/秒に制限され、バケットのクレジットは 40 MiB/秒で最大 1 TiB 分まで累積されます。

容量が大きいほど、これらの制限はリニアにスケールされ、スループットは最大 500 MiB/秒に制限されます。バケットが枯渇した後は、スループットは TiB あたり 40 MiB/秒のベースラインレートに制限されます。

ボリュームサイズが 0.125 ~ 16 TiB の場合、ベースラインスループットの範囲は 5 MiB/秒 ~ 500 MiB/秒 (上限値) です。次に示すように、この最大値には 12.5 TiB で到達します。

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

バーストスループットの範囲は、31 MiB/秒 ~ 500 MiB/秒 (上限) です。次に示すように、この上限には 2 TiB で到達します。

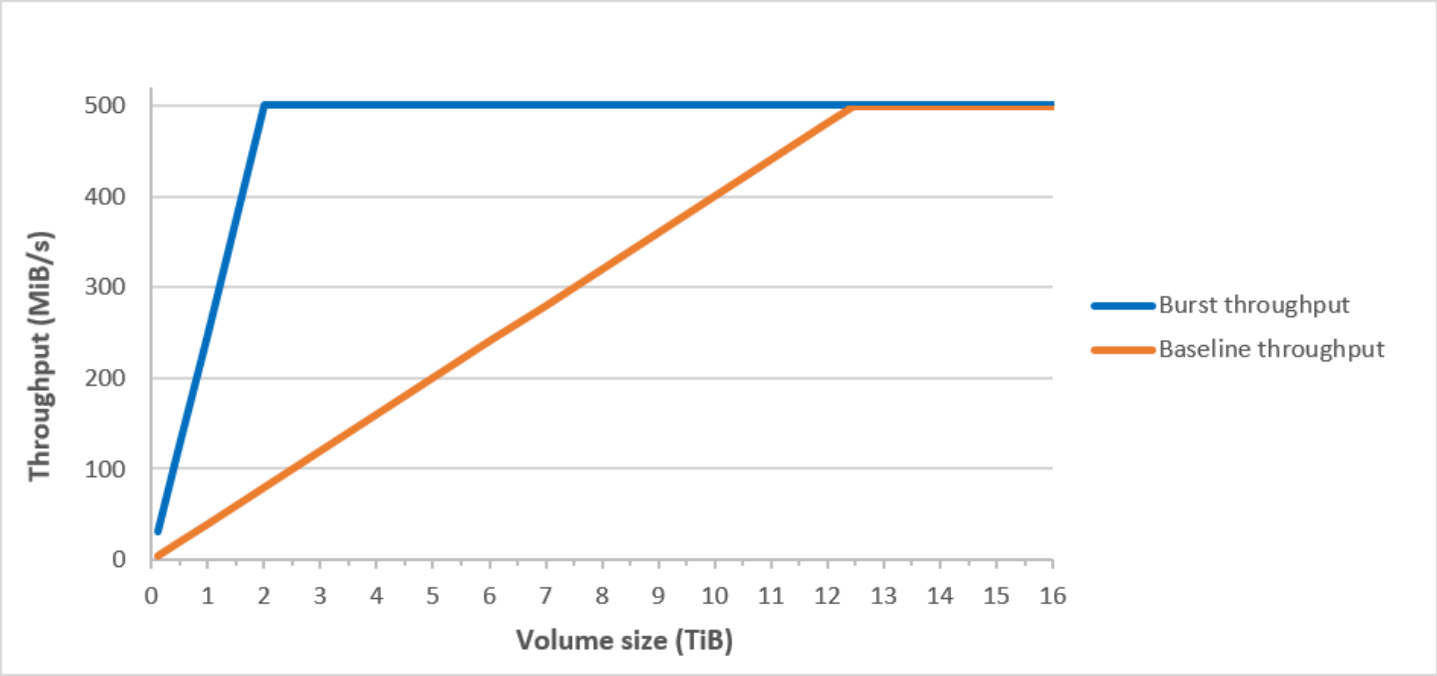
```
         250 MiB/s
2 TiB × ---------- = 500 MiB/s
          1 TiB
```

次の表は、st1 のベーススループット値およびバーストスループット値の範囲を示します。

ボリュームサイズ (TiB)	ST1 ベーススループット (MiB/秒)	ST1 バーストスループット (MiB/秒)
0.125	5	31
0.5	20	125
1	40	250
2	80	500
3	120	500

ボリュームサイズ (TiB)	ST1 ベーススループット (MiB/秒)	ST1 バーストスループット (MiB/秒)
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
13	500	500
14	500	500
15	500	500
16	500	500

次の図は、テーブルの値をグラフで示したものです。



> **Note**
>
> スループット最適化 HDD (st1) ボリュームのスナップショットを作成すると、スナップショットの進行中はボリュームのベースライン値までパフォーマンスが低下します。

CloudWatch メトリクスとアラームを使用してバーストバケットバランスをモニタリングする方法については、ボリュームのバーストバケットバランスのモニタリングを参照してください。

Cold HDD ボリューム

Cold HDD (sc1) ボリュームは、IOPS ではなくスループットでパフォーマンスを示す、低コストの磁気ストレージに使用できます。st1 は、sc1 よりスループット制限が低く、サイズの大きなコールドデータのシーケンシャルワークロードに適しています。データへのアクセス頻度が低く、コストの削減が必要である場合は、低コストなブロックストレージとして sc1 を使用できます。ブート可能な sc1 ボリュームはサポートされていません。

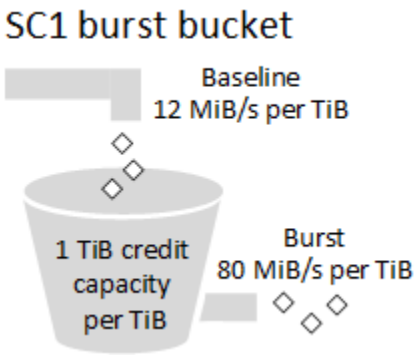
Cold HDD (sc1) ボリュームは、スループット最適化 HDD (st1) ボリュームに類似していますが、アクセス頻度が低いデータをサポートするように設計されています。

> **Note**
>
> このボリュームタイプは、サイズの大きなシーケンシャル I/O が含まれるワークロードに適しています。サイズの小さなランダム I/O を実行するワークロードのお客様には、Amazon EBS 汎用 SSD ボリューム または Amazon EBS プロビジョンド IOPS SSD ボリューム の使用をお勧めします。詳細については、「HDD に対する読み取り/書き込みサイズが小さい場合の非効率性」を参照してください。

Cold HDD (sc1) ボリュームを EBS 最適化インスタンスにアタッチすると、一貫したパフォーマンスが維持され、1 年で 99% の期間、想定されるスループットパフォーマンスの少なくとも 90% のボリュームが提供されます。

スループットクレジットとバーストパフォーマンス

gp2 と同様、sc1 でもパフォーマンスのためにバーストバケットモデルが使用されます。ボリュームのベースラインスループット (ボリュームのスループットクレジットが蓄積されるレート) は、ボリュームサイズによって決まります。ボリュームのバーストスループット (クレジットがある場合に可能な消費レート) もボリュームサイズによって決まります。ボリュームが大きいほど、ベースラインとバーストスループットの値も大きくなります。また、ボリュームのクレジットが多いほど、バーストレベルでドライブ I/O に使用できる時間が長くなります。



スループットとスループットクレジットの上限により、sc1 ボリュームで使用可能なスループットは、以下の計算式で示されます。

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

1 TiB の sc1 ボリュームの場合、バーストスループットは 80 MiB/秒に制限され、バケットのクレジットは 12 MiB/秒で最大 1 TiB 分まで累積されます。

容量が大きいほど、これらの制限はリニアにスケールされ、スループットは最大 250 MiB/秒に制限されます。バケットが枯渇した後は、スループットは TiB あたり 12 MiB/秒のベースラインレートに制限されます。

ボリュームサイズが 0.125 ~ 16 TiB の場合、ベースラインスループットの範囲は 1.5 MiB/秒 ~ 192 MiB/秒 (最大値) です。次に示すように、この最大値には 16 TiB で到達します。

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

バーストスループットの範囲は、10 MiB/秒 ~ 250 MiB/秒 (上限) です。次に示すように、この上限には 3.125 TiB で到達します。

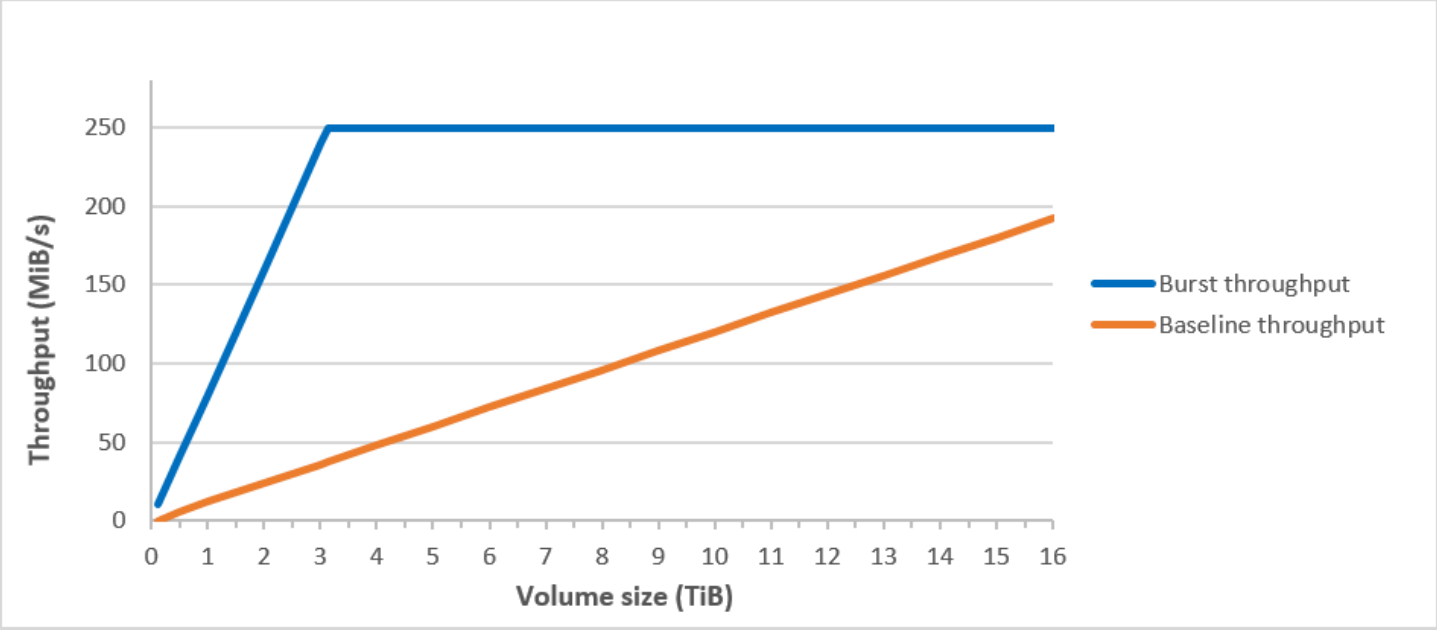
```
              80 MiB/s
3.125 TiB × ----------- = 250 MiB/s
               1 TiB
```

次の表は、sc1 のベーススループット値およびバーストスループット値の範囲を示します。

ボリュームサイズ (TiB)	SC1 ベーススループット (MiB/秒)	SC1 バーストスループット (MiB/秒)
0.125	1.5	10
0.5	6	40
1	12	80
2	24	160
3	36	240
3.125	37.5	250
4	48	250
5	60	250
6	72	250

ボリュームサイズ (TiB)	SC1 ベーススループット (MiB/秒)	SC1 バーストスループット (MiB/秒)
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

次の図は、テーブルの値をグラフで示したものです。



> **Note**
>
> Cold HDD (sc1) ボリュームのスナップショットを作成すると、スナップショットの進行中はボリュームのベースライン値までパフォーマンスが低下します。

CloudWatch メトリクスとアラームを使用してバーストバケットバランスをモニタリングする方法については、ボリュームのバーストバケットバランスのモニタリングを参照してください。

HDD ボリュームを使用するときのパフォーマンスに関する考慮事項

HDD ボリュームを使用して最適なスループットを実現するには、次の考慮事項を念頭に置いてワークロードを計画してください。

スループット最適化 HDD と Cold HDD との比較

st1 と sc1 のバケットサイズはボリュームサイズによって異なり、フルバケットにはフルボリュームスキャンのための十分なトークンが含まれています。ただし、st1 ボリュームと sc1 ボリュームの場合は、サイズが大きくなるほど、インスタンスごとおよびボリュームごとのスループット制限により、ボリュームスキャンの完了にかかる時間が長くなります。ボリュームが小さなインスタンスにアタッチされている場合は、st1 または sc1 のスループット制限よりインスタンスごとのスループットの方に制限されます。

st1 と sc1 のいずれも、全体のうち 99% の時間はバーストスループットの 90% のパフォーマンス安定性を実現できるよう設計されています。毎時間、予測合計スループットの 99% 達成を目標に、準拠しない期間はほぼ均一に分散されています。

スキャン時間は、一般的にこの式で示します。

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

例えば、パフォーマンス安定性の保証と他の最適化を想定すると、5 TiB のボリュームを持つ st1 のお客様は、フルボリュームスキャンが 2.91 ~ 3.27 時間で完了すると予測できます。

- 最適なスキャン時間

```
   5 TiB            5 TiB
----------- = ------------------ = 10,486 seconds = 2.91 hours
```

```
 500 MiB/s     0.00047684 TiB/s
```

- 最大スキャン時間

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

同様に、5 TiB のボリュームを持つ sc1 のお客様は、フルボリュームスキャンが 5.83～6.54 時間で完了すると予測できます。

- 最適なスキャン時間

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

- 最大スキャン時間

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

次の表は、フルバケットと十分なインスタンススループットを前提として、さまざまなサイズのボリュームに関する最も望ましいスキャン時間を示します。

ボリュームサイズ (TiB)	ST1 のスキャン時間、バーストを含む (時間)*	SC1 のスキャン時間、バーストを含む (時間)*
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

ボリュームサイズ (TiB)	ST1 のスキャン時間、バーストを含む (時間)*	SC1 のスキャン時間、バーストを含む (時間)*
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

* これらのスキャン時間では、1 MiB のシーケンシャル I/O を実行する際のキューの平均深度 (整数に四捨五入) として 4 以上を前提としています。

したがって、スキャンを早く (最大 500 MiB/秒) 完了するために必要なスループット指向のワークロードがある場合や、または 1 日に複数のフルボリュームスキャンが必要な場合は、st1 を使用してください。コストを最適化している場合、データのアクセス頻度が比較的低い場合、スキャンのパフォーマンスとして 250 MiB/秒を超える必要がない場合は、sc1 を使用してください。

HDD に対する読み取り/書き込みサイズが小さい場合の非効率性

st1 ボリュームおよび sc1 ボリュームのパフォーマンスモデルは、シーケンシャル I/O 用に最適化され、高スループットのワークロードに適しています。多様な IOPS およびスループットのワークロードに対して許容範囲のパフォーマンスを提供しますが、サイズの小さなランダム I/O のワークロードには向いていません。

例えば、1 MiB 以下の I/O リクエストは、1 MiB の I/O クレジットとしてカウントされます。ただし、I/O がシーケンシャルであれば、1 MiB の I/O ブロックにマージされ、1 MiB の I/O クレジットとしてのみカウントされます。

ボリュームのバーストバケットバランスのモニタリング

st1 および sc1 ボリュームのバーストバケットレベルをモニタリングするには、Amazon CloudWatch の Amazon EBS BurstBalance メトリクスを使用します。このメトリクスは、バーストバケット内の残りの st1 と sc1 のスループットクレジットを示します。BurstBalance メトリクス、および I/O に関連するその他のメトリクスの詳細については、Amazon EBS I/O の特性およびモニタリングを参照してください。CloudWatch では、BurstBalance 値があるレベルまで落ち込んだ時に通知するアラームを設定することもできます。詳細については、「CloudWatch アラームを作成する」を参照してください。

Amazon EBS ボリュームの制約

Amazon EBS ボリュームのサイズは、ブロックデータストレージの物理学と算術、およびオペレーティングシステム (OS) とファイルシステムデザイナーの実装決定によって制約されます。 は、ボリュームサイズに追加の制限 AWS を適用して、サービスの信頼性を保護します。

次のセクションでは、EBS ボリュームの使用可能サイズを制限する最も重要な要素と、EBS ボリュームを設定するための推奨事項について説明します。

コンテンツ

- ストレージキャパシティ
- サービスの制限
- パーティションスキーム
- データブロックサイズ

ストレージキャパシティ

次の表は、Amazon EBS で最も一般的に使用されているファイルシステムに実装された理論的なストレージ容量の概要を示しています (4,096 バイトのブロックサイズと仮定)。

パーティションスキーム	アドレス可能な最大ブロック	理論的な最大サイズ (ブロック x ブロックサイズ)	Ext4 に実装される最大サイズ*	XFS に実装される最大サイズ**	NTFS に実装される最大サイズ	EBS による最大サポート数
MBR	2^{32}	2 TiB	2 TiB	2 TiB	2 TiB	2 TiB
GPT	2^{64}	64 ZiB	1 EiB = 1024 ^{2}TiB (RHEL7 で認証されている 50 TiB)	500 TiB (RHEL7 で認証)	256 TiB	64 TiB†

*「Ext4 Howto」および「What are the file and file system size limitations for Red Hat Enterprise Linux?」

**「What are the file and file system size limitations for Red Hat Enterprise Linux?」

† io2 Block Express ボリュームは、最大 64 TiB のGPT パーティションをサポートします。詳細については、プロビジョンド IOPS SSD (io2) Block Express ボリュームを参照してください。

サービスの制限

Amazon EBS では、データセンターの大規模な分散ストレージを仮想ハードディスクドライブに抽象化しています。EC2 インスタンスにインストールされたオペレーティングシステムにとって、アタッチされた EBS ボリュームは、512 バイトのディスクセクタを含む物理ハードディスクドライブのように見えます。OS は、ストレージ管理ユーティリティを使用して、データブロック (またはクラスター) をその仮想セクタに割り当てます。この割り当ては、マスターブートレコード (MBR) または GUID パーティションテーブル (GPT) などのボリュームパーティションスキームに準拠しており、インストールされているファイルシステム (ext4、NTFS など) の機能の範囲内で行うことができます。

EBS では、仮想ディスクセクタ内のデータは認識されません。セクタの整合性の保護のみ行われます。つまり、 AWS アクションと OS アクションは互いに独立しています。ボリュームサイズを選択する場合は、次のように機能と制限の両方に注意してください。

- EBS では現在、最大 64 TiB のボリュームサイズがサポートされています。つまり、最大 64 TiB の EBS ボリュームを作成することはできますが、OS でその容量が認識されるかどうかは、そのOS自体の設計特性と、ボリュームのパーティションスキームによって異なります。
- ブートボリュームは、MBR または GPT パーティションスキームのいずれかを使用する必要があります。インスタンスを起動する AMI はブートモードを決定し、その後はブートボリュームに使用するパーティションスキームを決定します。

  MBR を使用すると、ブートボリュームのサイズは 2 TiB に制限されます。

  GPT を使用すると、GRUB2 (Linux) または UEFI ブートモード (Windows) と使用した場合、ブートボリュームは最大 64 TiB のサイズにすることができます。

  詳細については、「Amazon EBS ボリュームを使用できるようにする」を参照してください。
- 2 TiB (2048 GiB) 以上の非ブートボリュームは、ボリューム全体にアクセスするには GPT パーティションテーブルを使用する必要があります。

パーティションスキーム

他にも影響がある中で、このパーティションスキームは、単一ボリュームで一意にアドレス解決できる論理データブロックの数を決定します。詳細については、データブロックサイズを参照してください。使用中の一般的なパーティションスキームは、マスターブートレコード (MBR) と GUID パーティションテーブル (GPT) です。これらのパーティションスキームの重要な違いは次のようにまとめることができます。

MBR

MBR では、32 ビットのデータ構造を使用して、ブロックアドレスを格納します。これは、各データブロックが、正の整数 2^{32} のいずれかにマッピングされることを意味します。アドレス可能なボリュームの最大サイズは、次の式により得られます。

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

MBR ボリュームのブロックサイズは、通常 512 バイトに制限されています。したがって、

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

この MBR ボリュームの 2 TiB の制限を増やすための回避策は、一般的に広く普及していません。したがって、Linux と Windows は、 AWS がサイズを大きくしても、MBR ボリュームが 2 TiB より大きいことを検出しません。

GPT

GPT では、64 ビットのデータ構造を使用して、ブロックアドレスを格納します。これは、各データブロックが、正の整数 2^{64} のいずれかにマッピングされることを意味します。アドレス可能なボリュームの最大サイズは、次の式により得られます。

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

GPT ボリュームのブロックサイズは、一般的に 4,096 バイトです。したがって、

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

実際のコンピュータシステムでは、この理論上の最大値のような大きな値はサポートされていません。実装されたファイルシステムのサイズは現在、ext4 では 50 TiB、NTFS では 256 TiB に制限されています。

データブロックサイズ

現代のハードドライブ上のデータストレージは、論理ブロックアドレスや、オペレーティングシステムで基礎となるハードウェアをほとんど把握することなく論理ブロック内のデータを読み書きできる抽象化レイヤーによって管理されています。オペレーティングシステムは、ストレージデバイスを使用してブロックを物理セクターにマッピングし、セクターサイズの倍数であるデータブロックを使用してデータをディスクに読み書きします。

Amazon EBS は、次の要素に応じて、512 バイトまたは 4,096 バイト (4 KiB) の物理セクターをオペレーティングシステムにアドバタイズします。

1. Amazon EC2 インスタンスタイプ
2. オペレーティングシステム
3. NVMe ドライバーのバージョン

Amazon EBS は、すべての要素でサポートされている場合にのみ、4-KiB の物理セクターをアドバタイズします。これらのいずれかで 4-KiB の物理セクターがサポートされていない場合、Amazon EBS は 512 バイトの物理セクターをアドバタイズします。

Amazon EC2 インスタンスタイプのサポート

次の表は、Amazon EBS がさまざまな Amazon EC2 インスタンスタイプに対してアドバタイズするセクターサイズを示しています。

インスタンスタイプ	Linux	Windows
すべての Xen ベースのインスタンスタイプ	Amazon EBS は常に 512 バイトの物理セクターをアドバタイズします	
A1 | C5 | C5a | C5ad | C5d | C5n | C6g | C6gd | DL1 | D3 | D3en | G4ad | G4dn | G5 | G5g | I3 | I3en | Inf1 | M5 | M5a | M5ad | M5d | M5dn | M5n | M5zn | M6g | M6gd | P3dn | P4d | P4de | R5 | R5a | R5ad | R5d | R5dn | R5n | R6g | R6gd | T3 | T3a | T4g | U-12tb1 | U-18tb1 | U-24tb1 | U-3tb1 | U-6tb1 | U-9tb1 | X2gd | X2iezn | VT1 | Z1d	Amazon EBS は常に 512 バイトの物理セクターをアドバタイズします	Amazon EBS は 512 バイトまたは 4-KiB の物理セクターをアドバタイズします [1]
その他すべての Nitro ベースのインスタンス	Amazon EBS は 512 バイトまたは 4-KiB の物理セクターをアドバタイズします [1]	

[1] オペレーティングシステムのサポートによって異なります。次の「」セクションを参照してください。

オペレーティングシステムのサポート

次の表は、オペレーティングシステムの例と、Amazon EBS によってアドバタイズされる対応する物理セクターサイズを示しています。これは網羅的なリストではありません。オペレーティングシス

テムで Amazon EBS によってアドバタイズされた物理セクターのサイズを確認することをお勧めします。

オペレーティングシステム	アドバタイズされた物理セクターサイズ
• カーネルバージョン 4.14 以前を搭載する Amazon Linux • RHEL 7.9 以前 • Ubuntu 20.04 以前 • Windows 7/Windows Server 2008 以前	512 バイト
• カーネルバージョン 5.3 以降の Amazon Linux • RHEL 8.8 以降 • Ubuntu 22.04 以降 • Windows 8/Windows Server 2012 以降 [1]	4 KiB

[1] Windows ワークロードの場合は、最新バージョンの AWS NVMe ドライバーを使用していることを確認してください。Amazon EBS は、 AWS NVMe ドライバーバージョン 1.4.1 以降の 4-KiB 物理セクターをアドバタイズします。

デフォルト以外のブロックサイズ

論理データブロックの一般的なデフォルトサイズは、現在 4 KiB です。ワークロードによっては、ブロックサイズが小さいまたは大きい方がメリットを得られるため、ファイルシステムはデフォルト以外のブロックサイズをサポートしています。このサイズはフォーマット時に指定できます。デフォルト以外のブロックサイズを使用するシナリオ (最適化など) は、このトピックの対象外ですが、指定したブロックサイズによっては、ボリュームのストレージキャパシティに影響を及ぼす場合があります。次の表に、ブロックサイズの機能としての理論上のストレージキャパシティを示します。ただし、EBS で指定されているボリュームサイズ (io2 Block Express だと 64 TiB) の制限は、現在 16 KiB のデータブロックで使用できる最大サイズと同等であることに注意してください。

ブロックサイズ	最大ボリュームサイズ
4 KiB (デフォルト)	16 TiB

ブロックサイズ	最大ボリュームサイズ
8 KiB	32 TiB
16 KiB	64 TiB
32 KiB	128 TiB
64 KiB (最大)	256 TiB

Amazon EBS ボリュームと NVMe

Amazon EBS ボリュームは NVMe ブロックデバイスとして、AWS Nitro System 上に構築された Amazon EC2 インスタンスで公開されます。NVMe ブロックデバイスとして公開されている Amazon EBS ボリュームのパフォーマンスと機能を最大限に活用するには、EC2 インスタンスに AWS NVMe ドライバーがインストールされている必要があります。現行世代のすべての AWS Windows および Linux AMIs、 AWS デフォルトで NVMe ドライバーがインストールされています。

AWS NVMe ドライバーがない AMI を使用する場合は、手動でインストールできます。詳細については、Amazon EC2 ユーザーガイドの AWS NVMe ドライバーを参照してください。

Linux インスタンス

デバイス名は、「/dev/nvme0n1」や「/dev/nvme1n1」などです。ブロックデバイスマッピングで指定したデバイス名は、NVMe デバイス名 (/dev/nvme[0-26]n1) を使用して名称変更されます。ブロックデバイスドライバーは、ブロックデバイスマッピングのボリュームに指定した順序とは異なる順序で NVMe デバイス名を割り当てることができます。

Windows インスタンス

インスタンスにボリュームをアタッチする場合はボリュームのデバイス名を含めます。このデバイス名は Amazon EC2 によって使用されます。インスタンスのブロックデバイスドライバーは、ボリュームのマウント時に実際のボリューム名を割り当て、この割り当てられた名前は Amazon EC2 が使用する名前とは異なる可能性があります。

内容

- Amazon EBS ボリュームを NVMe デバイス名にマッピング
- Amazon EBS ボリュームの NVMe I/O オペレーションタイムアウト

- Amazon EBS ボリュームの NVMe Abort コマンド

Amazon EBS ボリュームを NVMe デバイス名にマッピング

EBS では、シングルルート I/O 仮想化 (SR-IOV) を使用して、NVMe 規格を使用して Nitro ベースのインスタンスにボリュームをアタッチします。これらのデバイスは、オペレーティングシステムの標準 NVMe ドライバーに依存しています。これらのドライバーは、通常、インスタンスのブート時にアタッチ済みのデバイスを検出し、そのデバイスがブロックデバイスマッピングでどのように指定されているかではなく、デバイスが応答する順序に基づいてデバイスノードを作成します。

Linux インスタンス

Linux では、NVMe デバイス名は /dev/nvme<x>n<y> のパターンに従います。ただし、<x> は列挙順序で、EBS の場合の <y> は 1 です。場合によっては、デバイスは後続のインスタンスの開始時に異なる順序で検出に応答することがあり、デバイス名が変更されます。また、ブロックデバイスドライバーによって割り当てられるデバイス名は、ブロックデバイスマッピングで指定される名前と異なる場合があります。

インスタンス内の EBS ボリュームには、次のいずれかのような安定した識別子を使用することをお勧めします。

- Nitro ベースのインスタンスでは、ブロックデバイスマッピングは、EBS ボリュームをアタッチしているとき、または `AttachVolume` か `RunInstances` API コールが、NVMe コントローラー ID のベンダー固有のデータフィールドに取り込まれる際に Amazon EC2 コンソールで指定されます。バージョン 2017.09.01 以降の Amazon Linux AMI で、このデータを読み込んでブロックデバイスマッピングへのシンボリックリンクを作成する udev ルールを提供します。
- EBS ボリューム ID とマウントポイントは、インスタンスの状態が変化しても安定しています。NVMe デバイス名は、インスタンスの起動時にデバイスが応答する順序に応じて変化します。一貫性のあるデバイスを識別するには、EBS ボリューム ID とマウントポイントを使用することをお勧めします。
- NVMe EBS ボリュームには、EBS ボリューム ID がデバイス ID のシリアル番号として設定されています。シリアル番号を一覧表示するには、`lsblk -o +SERIAL` コマンドを使用します。
- NVMe デバイス名の形式は、EBS ボリュームがインスタンスの起動中または起動後にアタッチされたかどうかによって異なります。インスタンスの起動後にアタッチされたボリュームの NVMe デバイス名には、/dev/プレフィクスが含まれますが、インスタンスの起動中にアタッチされたボリュームの NVMe デバイス名には/dev/プレフィクスが含まれません。

- Amazon Linux または FreeBSD AMI の場合、`sudo ebsnvme-id /dev/nvme0n1 -u` コマンドを使用して、一貫した NVMe デバイス名を指定します。
- その他のディストリビューションでは`sudo nvme id-ctrl -V /dev/nvme0n1`コマンドを使用して NVMe デバイス名を指定します。`--vendor-specific` コマンドオプションを含める必要がある場合があります。

- デバイスがフォーマットされると、ファイルシステムの存続期間中、存続する UUID が生成されます。デバイスラベルは同時に指定することができます。詳細については、「Amazon EBS ボリュームを使用できるようにする および 間違ったボリュームからの起動」を参照してください。

Amazon Linux AMI

AMI Amazon Linux 2017.09.01 以降 (Amazon Linux 2 を含む) では、次のように ebsnvme-id コマンドを実行して、NVMe デバイス名をボリューム ID とデバイス名にマップすることができます。

次の例は、インスタンスの起動時にアタッチされたボリュームのコマンドと出力を示しています。NVMe デバイス名には、/dev/プレフィクスが含まれないことに注意してください。

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

次の例は、インスタンスの起動後にアタッチされたボリュームのコマンドと出力を示しています。NVMe デバイス名に、/dev/プレフィクスが含まれることに注意してください。

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

また、Amazon Linux はブロックデバイスマッピング (例えば、/dev/sdf) 内のデバイス名から NVMe デバイス名へのシンボリックリンクを作成します。

FreeBSD AMI

FreeBSD 12.2-RELEASE 以降では、上記のように ebsnvme-id コマンドを実行することができます。NVMe デバイスの名前 (nvme0 など) またはディスクデバイス (nvd0 または nda0) を渡します。FreeBSD は、ディスクデバイスへのシンボリックリンク (/dev/aws/disk/ebs/*volume_id* など) も作成します。

その他の Linux AMI

カーネルバージョン 4.2 以降では、次のように nvme id-ctrl コマンドを実行して、NVMe デバイスをボリューム ID にマップすることができます。最初に、Linux ディストリビューションのパッケージ管理ツールを使用して、NVMe コマンドラインのパッケージ nvme-cli をインストールします。他のディストリビューションのダウンロードおよびインストール手順については、ディストリビューションに固有のドキュメントを参照してください。

次の例では、インスタンスの起動時にアタッチされたボリュームのボリューム ID と NVMe デバイス名を取得します。NVMe デバイス名には、/dev/プレフィクスが含まれないことに注意してください。デバイス名は、NVMe コントローラベンダー固有の拡張子 (コントローラー ID のバイト 384:4095) を介して使用できます。

```
[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..."
```

次の例では、インスタンスの起動後にアタッチされたボリュームのボリューム ID と NVMe デバイス名を取得します。NVMe デバイス名には、/dev/プレフィクスが含まれることに注意してください。

```
[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..."
```

lsblk コマンドは、使用可能なデバイスとそのマウントポイント (該当する場合) をリストします。これは、使用する正しいデバイス名を決定するのに役立ちます。この例では、/dev/nvme0n1p1 がルートデバイスとしてマウントされ、/dev/nvme1n1 はアタッチされていますがマウントされていません。

```
[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
```

Windows インスタンス

ebsnvme-id コマンドを実行して、NVMe デバイスのディスク番号を EBS ボリューム ID およびデバイス名にマッピングできます。デフォルトでは、すべての EBS NVMe デバイスは列挙されません。特定のデバイスの情報を列挙するには、ディスク番号を渡します。このebsnvme-idツールは、にある最新の AWS Windows Server AMIs に含まれていますC:\ProgramData\Amazon\Tools。

AWS NVMe ドライバーパッケージ以降1.5.0,、ebsnvme-idツールの最新バージョンはドライバーパッケージによってインストールされます。最新バージョンはドライバーパッケージでのみ入手可能です。ebsnvme-id ツールのスタンドアロンダウンロードリンクにはアップデートが送信されなくなります。スタンドアロンリンクから入手できる最後のバージョンは 1.1.0 です。このバージョンは ebsnvme-id.zip リンクを使用してコンテンツを Amazon EC2 インスタンスに抽出することでダウンロード可能となり、ebsnvme-id.exe にアクセスできるようになります。

```
PS C:\ProgramData\Amazon\Tools> 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:\ProgramData\Amazon\Tools> ebsnvme-id.exe 4
Disk Number: 4
Volume ID: vol-03e2dbe464b66f0a1
```

```
Device Name: xvdc
```

Amazon EBS ボリュームの NVMe I/O オペレーションタイムアウト

ほとんどのオペレーティングシステムは、NVMe デバイスに送信される I/O オペレーションのタイムアウトを指定します。

Linux インスタンス

Linux では、Nitro ベースのインスタンスにアタッチされた EBS ボリュームは、オペレーティングシステムが提供するデフォルトの NVMe ドライバーを使用します。ほとんどのオペレーティングシステムは、NVMe デバイスに送信される I/O オペレーションのタイムアウトを指定します。デフォルトのタイムアウトは 30 秒で、`nvme_core.io_timeout` ブートパラメータを使用して変更できます。バージョン 4.6 より前の Linux カーネルでは、このパラメータは `nvme.io_timeout` です。

I/O レイテンシーがこの timeout パラメータの値を超えると、Linux NVMe ドライバーは I/O に失敗し、ファイルシステムまたはアプリケーションにエラーを返します。I/O オペレーションに応じて、ファイルシステムまたはアプリケーションはエラーを再試行できます。場合によっては、ファイルシステムを読み取り専用として再マウントすることがあります。

Xen インスタンスに接続された EBS ボリュームに類似するエクスペリエンスのため、`nvme_core.io_timeout` を可能な限り最大値に設定することをお勧めします。現在のカーネルでは、最大値は 4294967295 ですが、以前のカーネルでは最大値は 255 です。Linux のバージョンに応じて、タイムアウトはすでにサポートされる最大値に設定されていることがあります。例えば、Amazon Linux AMI 2017.09.01 以降では、デフォルトでタイムアウトが 4294967295 に設定されています。

Linux ディストリビューションの最大値を確認するには、示されている最大値よりも高い値を `/sys/module/nvme_core/parameters/io_timeout` に書き込み、ファイルを保存する際に範囲外の数値結果エラーがないかどうかをチェックします。

Windows インスタンス

Windows では、デフォルトのタイムアウトは 60 秒で、最大は 255 秒です。`TimeoutValue` ディスククラスのレジストリ設定は、SCSI ミニドライバーのレジストリエントリで説明されている手順を使用して変更できます。

Amazon EBS ボリュームの NVMe Abort コマンド

Abort コマンドは、以前にコントローラーに送信された特定のコマンドを中止するために発行される NVMe Admin コマンドです。このコマンドは、通常、I/O オペレーションのタイムアウトしきい値を超えたストレージデバイスに対して、デバイスドライバーによって発行されます。

デフォルトで Abort コマンドをサポートする Amazon EC2 インスタンスタイプは、Abort コマンドが発行されアタッチされた Amazon EBS デバイスのコントローラーに以前に送信された特定のコマンドを中止します。Abort コマンドをサポートしていない Amazon EC2 インスタンスは、アタッチされた Amazon EBS ボリュームに Abort コマンドが発行されてもアクションを実行しません。

Abort コマンドは、以下でサポートされています。

- NVMe デバイスバージョン 1.4 以降の Amazon EBS デバイス。
- Xen ベースのインスタンスタイプと次の Nitro ベースのインスタンスタイプを除くすべての Amazon EC2 インスタンス:
  - 汎用: A1 | M5 | M5a | M5ad | M5d | M5dn | M5n | M5zn | M6g | M6gd | Mac1 | Mac2 | T3 | T3a | T4g
  - コンピューティングの最適化: C5 | c5a | C5ad | C5d | C5n | C6g | C6gd
  - メモリ最適化: R5 | R5a | R5ad | R5d | R5dn | R5n | R6g | R6gd | U-12tb1 | U-18tb1 | U-24tb1 | U-3tb1 | U-6tb1 | U-9tb1 | X2gd | X2iezn | Z1d
  - ストレージ最適化: D3 | D3en | I3en
  - 高速コンピューティング: DL1 | G4ad | G4dn | G5 | G5g | Inf1 | P3dn | P4d | P4de | VT1

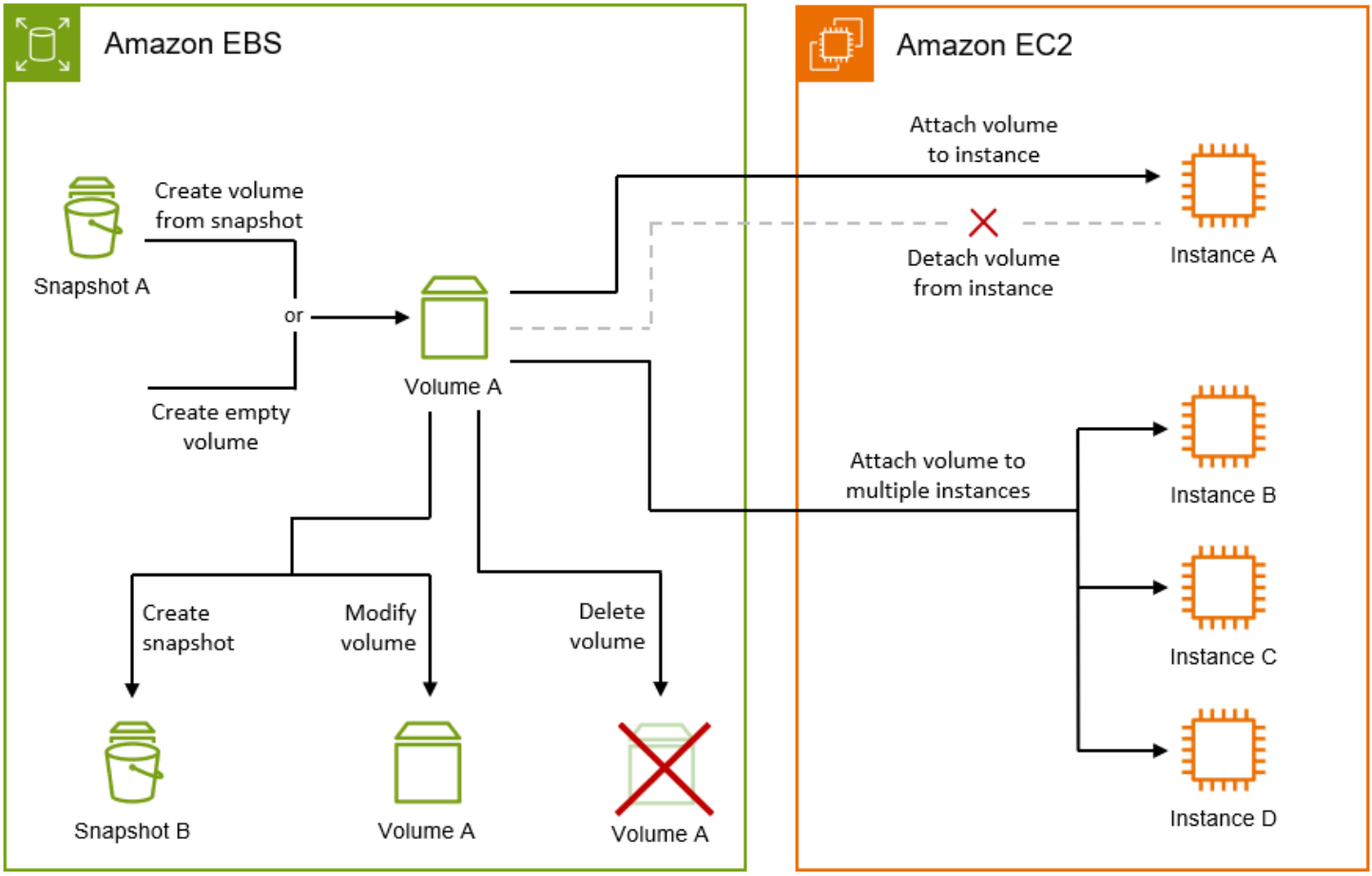
詳細については、「NVM Express の基本仕様」のセクション「5.1 Abort コマンド」を参照してください。

Amazon EBS ボリュームのライフサイクル

Amazon EBS ボリュームのライフサイクルは、作成プロセスから始まります。Amazon EBS スナップショットからボリュームを作成するか、空のボリュームを作成できます。ボリュームを使用する前に、ボリュームと同じアベイラビリティーゾーンにある 1 つ以上の Amazon EC2 インスタンスにボリュームをアタッチする必要があります。複数のボリュームを 1 つのインスタンスにアタッチできます。必要に応じて、1 つのインスタンスからボリュームをデタッチし、別のインスタンスにアタッチできます。ストレージ要件が変更された場合、いつでもボリュームのサイズまたはパフォーマンスを変更できます。Amazon EBS スナップショットを作成することにより、ボリュームのポイントイ

ンタイムバックアップを作成できます。ボリュームが不要になった場合、それを削除して関連するストレージコストの発生を中止できます。

次の図は、ボリュームライフサイクルの一環としてボリュームに実行できるアクションを示しています。インスタンスに接続してオペレーティングシステムのコマンドを実行することにより、実行するタスクもあります。例えば、ボリュームのフォーマット、ボリュームのマウント、パーティションの管理、空きディスク容量の表示などです。



タスク

- Amazon EBS ボリュームの作成
- ボリュームを共有する
- Amazon EBS ボリュームをコピーする
- Amazon EBS ボリュームを Amazon EC2 インスタンスにアタッチ
- マルチアタッチを使用して EBS ボリュームを複数の EC2 インスタンスへアタッチ
- Amazon EBS ボリュームを使用できるようにする
- Amazon EBS ボリュームに関する情報の表示

- Elastic Volumes オペレーションを使用して Amazon EBS ボリュームを変更する
- Amazon EC2 インスタンスから Amazon EBS ボリュームをデタッチ
- Amazon EBS ボリュームの削除

Amazon EBS ボリュームの作成

Amazon EBS ボリュームを作成し、同じアベイラビリティーゾーン内の任意の EC2 インスタンスにアタッチできます。

Amazon EBS スナップショットからボリュームを作成するか、空のボリュームを作成できます。スナップショットに基づいて EBS ボリュームを作成した場合、ボリュームは、スナップショットの作成に使用されたボリュームの完全なレプリカとして開始されます。

ボリュームの初期化

スナップショットからボリュームを作成した場合、それにアクセスする前に、スナップショットのストレージブロックは Amazon S3 からダウンロードされてボリュームに書き込まれる必要があります。このプロセスはボリューム初期化と呼ばれます。この間、ボリュームの I/O レイテンシーは増加しています。すべてのストレージブロックがダウンロードされてボリュームに書き込まれた後にのみ、フルボリュームのパフォーマンスに達します。

デフォルトのボリューム初期化レートは初期化プロセス全体で変動するため、完了時間が予測できなくなる可能性があります。

ボリューム初期化に関連するパフォーマンスへの影響を最小限に抑えるには、ボリューム初期化のための Amazon EBS プロビジョンドレート (ボリューム初期化レート) または高速スナップショット復元を使用できます。詳細については、「Amazon EBS ボリュームの初期化」を参照してください。

ボリュームの暗号化

ボリュームの暗号化状態は、アカウントがデフォルトで暗号化が有効になっているかどうか、そして使用を選択する場合はスナップショットの暗号化状態によって異なります。次の表は、考えられる暗号化の結果をまとめたものです。

デフォルトでの暗号化	スナップショット使用の有無	ボリューム暗号化の結果	メモ
無効	いいえ	オプションの暗号化	暗号化を有効にする場合、使用する KMS キーを指定できます。暗号化を有効にしても KMS キーを指定しない場合、 AWS マネージドキー (aws/ebs) が使用されます。
無効	はい、暗号化されていません	オプションの暗号化	暗号化を有効にする場合、使用する KMS キーを指定できます。暗号化を有効にしても KMS キーを指定しない場合、 AWS マネージドキー (aws/ebs) が使用されます。
無効	はい、暗号化されています	自動暗号化	使用する KMS キーを指定できます。KMS キーを指定しない場合、ボリュームはソーススナップショットと同じ KMS キーを使用して暗号化されます。
有効	いいえ	自動暗号化	使用する KMS キーを指定できます。KMS キーを指定しない場合、暗号化用に指定されたキーがデフォルトで使用されます。
有効	はい、暗号化されていません	自動暗号化	使用する KMS キーを指定できます。KMS キーを指定しない場合、暗号化用に指定されたキーがデフォルトで使用されます。
有効	はい、暗号化されています	自動暗号化	使用する KMS キーを指定できます。KMS キーを指定しない場合、ボリュームはソーススナップショットと同じキー (コンソール) またはデフォルトで暗号化

デフォルトでの暗号化	スナップショット使用の有無	ボリューム暗号化の結果	メモ
			用に指定されたキー (CLI/API) を使用して暗号化されます。

その他の考慮事項

- ボリュームは、インスタンスと同じアベイラビリティーゾーンにアタッチする必要があります。
- ボリュームは、`available` 状態になった後にのみ使用できるようになります。
- コンソールを使用してボリュームを作成する場合、デフォルトのボリュームタイプは gp3 です。コマンドラインツール、API、および SDK では、デフォルトのボリュームタイプは gp2 です。
- Outpost で実行されているインスタンスでボリュームを使用するには、インスタンスと同じ Outpost にボリュームを作成する必要があります。
- Windows インスタンスで使用するボリュームを作成し、そのボリュームが 2048 GiB を超える場合は、GPT パーティションテーブルを使用するようにボリュームを設定してください。詳細については、Amazon EBS ボリュームの制約 と「2 TB を超えるハードディスクの Windows でのサポート」を参照してください。
- また、ボリュームは Amazon EC2 インスタンスを起動することで間接的に作成されます。インスタンスの起動に使用した AMI、またはインスタンス起動リクエスト自体に Amazon EBS ボリュームのブロックデバイスマッピングを含めることができます。詳細については、「ブロックデバイスマッピング」を参照してください。

Console

ボリュームを作成するには

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. ナビゲーションペインで [ボリューム] を選択して、[ボリュームを作成] を選択します。
3. (Outpost のお客様のみ) Outpost ARN の場合は、ボリュームを作成する AWS Outpost の ARN を入力します。

4. [Volume type] (ボリュームタイプ) に、作成するボリュームのタイプを選択します。使用できるボリュームタイプの詳細については、「Amazon EBS ボリュームの種類」を参照してください。
5. [サイズ] に、ボリュームのサイズ (GiB) を入力します。詳細については、「Amazon EBS ボリュームの制約」を参照してください。
6. (*io1*、*io2*、*gp3* のみ) [IOPS] に、ボリュームが提供する IOPS (1 秒あたりの入力/出力オペレーションの数) の最大数を入力します。
7. (*gp3* のみ) [スループット] に、ボリュームが提供すべきスループットを MiB/秒単位で入力します。
8. [アベイラビリティーゾーン] では、ボリュームを作成するアベイラビリティーゾーンを選択します。
9. [スナップショット ID] で、次のいずれかを実行します。
   - 空のボリュームを作成するには、デフォルト値のままにします ([スナップショットからボリュームを作成しない])。
   - スナップショットからボリュームを作成するには、使用するスナップショットを選択します。
10. [ボリューム初期化レート] にスナップショットを選択した場合は、オプションでボリューム初期化レートを MiB/秒で指定できます。スナップショットブロックはこのレートで Amazon S3 から作成後のボリュームにダウンロードされます。詳細については、「ボリューム初期化に Amazon EBS プロビジョンドレートを使用する」を参照してください。デフォルトの初期化レート、または高速スナップショット復元 (選択したスナップショットで有効化されている場合) を使用する場合は、レートを指定しないでください。
11. (*io1* および *io2* のみ) Amazon EBS マルチアタッチのボリュームを有効にするには、[マルチアタッチを有効化] を選択します。詳細については、マルチアタッチを使用して EBS ボリュームを複数の EC2 インスタンスへアタッチを参照してください。
12. ボリュームの暗号化ステータスを設定します。
    - アカウントにおいてデフォルトでの暗号化が有効になっている場合、暗号化は自動的に有効になり、無効にすることはできません。
    - 暗号化されたスナップショットを選択した場合、暗号化は自動的に行われ、無効にすることはできません。
    - アカウントにおいてデフォルトでの暗号化が有効になっていない、かつ選択したスナップショットが暗号化されていない、またはそもそも選択していない場合、暗号化はオプションです。

13. (オプション) ボリュームにカスタムタグを割り当てるには、[タグ] セクションで [タグの追加] を選択し、タグのキーと値のペアを入力します。
14. [Create volume] (ボリュームの作成) を選択します。
15. ボリュームを使用するには、ボリュームが available 状態になるまで待ってから、同じアベイラビリティーゾーンの Amazon EC2 インスタンスにアタッチします。詳細については、「Amazon EBS ボリュームを Amazon EC2 インスタンスにアタッチ」を参照してください。

AWS CLI

ボリュームを作成するには

create-volume コマンドを使用します。次の例では、指定されたアベイラビリティーゾーンにサイズが 100 GiB の空の gp3 ボリュームを作成します。

```
aws ec2 create-volume \
    --volume-type gp3 \
    --size 100 \
    --availability-zone us-east-1a
```

PowerShell

ボリュームを作成するには

New-EC2Volume コマンドレットを使用します。次の例では、指定されたアベイラビリティーゾーンにサイズが 100 GiB の空の gp3 ボリュームを作成します。

```
New-EC2Volume `
    -VolumeType gp3 `
    -Size 100 `
    -AvailabilityZone us-east-1a
```

ボリュームを共有する

AWS Resource Access Manager (AWS RAM) を使用して、Amazon EBS ボリュームを他の AWS アカウントと共有できます。ボリュームを共有すると、消費アカウントはボリューム情報を表示し、共有ボリュームのコピーを同じアベイラビリティーゾーンに作成できます。コピーの作成の詳細については、「」を参照してくださいAmazon EBS ボリュームをコピーする。消費アカウントは、共有ボリュームのスナップショットをアタッチ、変更、削除、作成することはできません。

たとえば、本番稼働用ボリュームを開発アカウントと共有して、アカウントレベルの分離を維持しながら、テスト環境up-to-date本番稼働用データで更新できます。

次のボリュームを共有できます。

- 暗号化されていない
- カスタマーマネージドキーで暗号化

Amazon EBS のデフォルトの AWS マネージドキーで暗号化されたボリュームを共有することはできません。

内容

- アクセス許可
- ボリュームを共有する
- 共有ボリュームを表示する
- ボリュームの共有に関する考慮事項
- 共有アクティビティをモニタリングする
- 共有ボリュームとごみ箱
- 料金

アクセス許可

共有ボリュームのリソース共有を作成するときは、マネージドアクセス許可をリソース共有に関連付ける必要があります。マネージドアクセス許可は、アカウントを消費する がリソース共有の共有ボリュームに対して実行できるアクションを決定します。 は、次の AWS マネージドアクセス許可 AWS RAM を提供します。

- AWSRAMDefaultPermissionEBSVolume (デフォルト) – ボリューム情報を表示するアクセス許可を消費するアカウントに付与します。
- AWSRAMPermissionEBSVolumeCopyAccess – ボリューム情報を表示し、共有ボリュームのコピーを作成するアクセス許可を消費アカウントに付与します。

または、`ec2:Volume`リソースタイプのカスタマー管理アクセス許可を作成することもできます。詳細については、「AWS RAM ユーザーガイド」の「Creating and using customer managed permission」を参照してください。

ボリュームを共有する

Amazon EBS ボリュームを共有するには、次のいずれかの方法を使用します。

Console

ボリュームを共有するには

1. で Amazon Elastic Compute Cloud コンソールを開きますhttps://console.aws.amazon.com/ec2/。
2. ナビゲーションペインの [Volumes] を選択してください。
3. ボリュームを選択し、アクション、ボリュームの共有を選択します。
4. 次のいずれかを行います:
   - ボリュームを既存のリソース共有に追加するには、リソース共有を選択します。
   - 新しいリソース共有を作成するには、新しいリソース共有の作成を選択します。これにより、共有を設定する AWS RAM コンソールが開きます。
5. プリンシパルの場合は、ボリュームを共有するアカウントを追加します。個々のアカウント IDs、組織単位 (OUs)、または組織全体を追加できます。
6. 管理アクセス許可では、次のいずれかを選択します。
   - AWSRAMDefaultPermissionEBSVolume (デフォルト) — 消費アカウントはボリュームメタデータのみを表示できます。
   - AWSRAMPermissionEBSVolumeCopyAccess – 消費アカウントはボリュームメタデータを表示し、共有ボリュームのコピーを作成できます。
   - `ec2:Volume` リソースタイプ用に作成したカスタマー管理アクセス許可。
7. [リソースの共有の作成] を選択します。

AWS CLI

create-resource-share コマンドを使用します。ボリューム ARN、消費アカウント、アクセス許可 ARN を指定します。ボリューム ARN を検索するには、describe-volumes を呼び出します。レスポンスには、各ボリュームの `VolumeArn`フィールドが含まれます。

次の例では、 アカウント `vol-1234567890abcdef0`とボリュームを共有するリソース共有を作成し`222222222222`、 `AWSRAMPermissionEBSVolumeCopyAccess`マネージドアクセス許可を割り当てます。

```
aws ram create-resource-share \
--name my-volume-share \
--resource-arns arn:aws:ec2:us-east-1:111111111111:volume/vol-1234567890abcdef0 \
--principals 222222222222 \
--permission-arns arn:aws:ram::aws:permission/AWSRAMPermissionEBSVolumeCopyAccess \
--region us-east-1
```

デフォルトのアクセス許可 (、説明のみ) で を共有するには`AWSRAMDefaultPermissionEBSVolume`、 `--permission-arns`パラメータを省略します。

```
aws ram create-resource-share \
--name my-volume-share \
--resource-arns arn:aws:ec2:us-east-1:111111111111:volume/vol-1234567890abcdef0 \
--principals 222222222222 \
--region us-east-1
```

リソース共有の詳細については、AWS RAM 「 ユーザーガイド」の「リソース共有の作成」を参照してください。

共有ボリュームを表示する

リソース共有を作成すると、指定されたアカウントは招待を受け取ります。の組織外のアカウントは、共有ボリュームにアクセスする前に招待を受け入れる AWS Organizations 必要があります。詳細については、「AWS RAM ユーザーガイド」の「リソース共有への招待の受け入れと拒否」を参照してください。

アカウントと共有されているボリュームは、次のいずれかの方法を使用して表示できます。

Console

共有ボリュームを表示するには

1. で Amazon Elastic Compute Cloud コンソールを開きますhttps://console.aws.amazon.com/ec2/。
2. ナビゲーションペインの [Volumes] を選択してください。
3. 次のいずれかを行います:

- Shared with me プリセットフィルターを選択して、アカウントと共有されているすべてのボリュームを表示します。
- フィルターの追加、所有者 ID を選択し、共有アカウント ID を入力します。

共有ボリュームは、共有アカウント ID を所有者 ID 列に表示します。

AWS CLI

owner-id フィルターで describe-volumes コマンドを使用して、特定のアカウントによって共有されているボリュームを表示します。

```
aws ec2 describe-volumes \
--filters Name=owner-id,Values=111111111111 \
--region us-east-1
```

共有ボリュームには、共有アカウント ID を示す OwnerIdフィールドがレスポンスに含まれます。

ボリュームの共有に関する考慮事項

ボリュームを他のアカウントと共有する場合は、次の点を考慮してください。

- 暗号化されていないボリュームと、カスタマーマネージドキーで暗号化されたボリュームを共有できます。Amazon EBS のデフォルトの AWS マネージドキーで暗号化されたボリュームを共有することはできません。
- ボリュームを共有するには、ボリューム所有者のアカウントにデフォルトの Amazon EBS 暗号化キーに対するkms:DescribeKeyアクセス許可が必要です。これは、サービスが共有プロセス中にボリュームの暗号化状態を検証するため、暗号化されていないボリュームを含むすべてのボリュームに必要です。
- カスタマーマネージドキーで暗号化されたボリュームを共有するには、KMS キーを消費アカウントと共有する必要があります。詳細については、AWS Key Management Service デベロッパーガイドの「他のアカウントのユーザーに KMS キーの使用を許可する」を参照してください。
- 共有暗号化ボリュームのコピーを作成するには、消費アカウントで KMS キーkms:Decryptに kms:CreateGrant、kms:GenerateDataKeykms:GenerateDataKeyWithoutPlaintext、kms:Reが必要です。
- 消費アカウントは、共有ボリュームのスナップショットをアタッチ、変更、削除、作成することはできません。ボリュームメタデータの表示とコピーの作成のみが可能です。

- ボリュームが共有されているすべてのアカウントで、共有ボリュームに対して一度に進行中のコピーオペレーションは 1 つだけです。追加のコピーリクエストは、現在のオペレーションが完了するまでエラーを返します。
- アベイラビリティーゾーン名 (などus-east-1a) は、異なる AWS アカウントの異なる物理的な場所にマッピングされます。アベイラビリティーゾーン IDs (などuse1-az1) を使用して、アカウント間で同じ物理的な場所を識別します。詳細については、「 AWS RAM ユーザーガイド」のIDs AWS」を参照してください。
- リソース共有からボリュームを削除しても、進行中のコピーオペレーションはキャンセルされません。
- ソースボリュームを削除しても、進行中のコピーオペレーションはキャンセルされません。

共有アクティビティをモニタリングする

AWS CloudTrail API ログ記録

すべてのボリューム共有 API コールとコピー API コールがログインされます AWS CloudTrail。CopyVolumesAPI コールは、コピーを開始するアカウントとボリューム所有者のアカウントの両方についてログに記録されます。詳細については、「 を使用した Amazon Elastic Compute Cloud API コールのログ記録 AWS CloudTrail」を参照してください。

Amazon EventBridge 通知

Amazon EBS は、クロスアカウントコピーオペレーションが完了すると Amazon EventBridge イベントを送信します。ボリューム所有者と消費アカウントの両方がイベントを受け取ります。ボリュームコピーイベントは、イベント名 EBS Volume Notification の詳細タイプを使用しますsharedVolumeCopy。イベントには、ボリューム ID、完了ステータス (completed または failed)、タイムスタンプが含まれます。

通知を設定するには、EBS Volume NotificationイベントをsharedVolumeCopyイベント名と一致する Amazon EventBridge ルールを作成します。詳細については、「Amazon EBS の Amazon EventBridge イベント」を参照してください。

共有ボリュームとごみ箱

ボリュームに適用されるごみ箱保持ルールがある場合、共有ボリュームを削除するとごみ箱に移動します。ボリュームは、保持期間中、 AWS RAM リソース共有に残ります。ごみ箱からボリュームを復元すると、削除前に属していたのと同じリソース共有からアクセスできます。

ごみ箱にあるボリュームでは、次のリソース共有動作に注意してください。

- ボリュームがごみ箱に入っている間にリソース共有からアカウントを削除した場合、ボリュームを復元しても、そのアカウントアクセスは再付与されません。
- ボリュームがごみ箱に入っている間にリソース共有にアカウントを追加した場合、そのアカウントは復元後にボリュームにアクセスできます。

ごみ箱からボリュームを復元したら、AWS RAM コンソールでリソース共有を確認して、目的のアカウントのみがアクセスできることを確認します。

料金

を通じてボリュームを共有するためのコストはかかりません AWS RAM。コピーを作成するアカウントは、新しいボリュームのボリュームコピー料金と通常の Amazon EBS ボリューム料金を支払います。詳細については、Amazon EBS の料金表を参照してください。

Amazon EBS ボリュームをコピーする

ボリュームコピーは、ソースボリュームのクラッシュコンシステントなpoint-in-timeコピーです。これには、ボリュームコピーの初期化開始時にソースボリュームに書き込まれたすべてのデータブロックが含まれます。ボリュームコピーは、独自の一意のボリューム ID を取得します。ボリュームコピーはすぐに作成され、`available` 状態になると Amazon EC2 インスタンスにアタッチできます。ボリュームコピーを使用すると、テスト環境と開発環境のために本番データをすばやくコピーできます。

他のアカウントと共有されているボリュームからコピーを作成することもできます AWS 。別のアカウントからボリュームをコピーするには、ソースボリュームを AWS Resource Access Manager () を通じて共有する必要がありますAWS RAM。詳細については、「ボリュームを共有する」を参照してください。

初期化

ボリュームコピーは作成後に初期化されます。初期化中、データブロックはソースボリュームからコピーされ、バックグラウンドでボリュームコピーに書き込まれます。初期化が完了するまで、ボリュームは `initializing` 状態のままです。

初期化中のパフォーマンス

コピーオペレーションは、ソースボリュームのパフォーマンスには影響しません。コピープロセス中は、ソースボリュームを通常どおり使用し続けることができます。コピーされたボリュームには、

ソースボリュームからデータがコピーされるのを待たずにすぐにアクセスできます。ボリュームコピーは、1 桁ミリ秒のレイテンシーでデータへの即時アクセスを提供します。ただし、実際のレイテンシーはボリュームタイプによって異なる場合があります。初期化中、ボリュームコピーは、次の 3 つの値のうち最も低い値に等しいベースラインパフォーマンスを提供します。

- 3,000 入出力オペレーション/秒 (IOPS) および 125 MiB/秒
- ソースボリュームのプロビジョニングされたパフォーマンス
- ボリュームコピーのプロビジョニングされたパフォーマンス

次の条件が満たされると、ボリュームコピーがベースラインパフォーマンスを超える場合があります。

1. ソースボリュームとボリュームコピーの両方が 3,000 IOPS および 125 MiB/秒以上でプロビジョニングされる。
2. ソースボリュームのパフォーマンスキャパシティが活用されていない (駆動パフォーマンスがプロビジョニングされたパフォーマンスよりも低い) 。

例えば、ソースボリュームが 10,000 IOPS でプロビジョニングされていて、ワークロードが現在 5,000 IOPS しか駆動しておらず、ボリュームコピーが 10,000 IOPS でプロビジョニングされている場合、ボリュームコピーは、ソースボリュームの未使用の 5,000 IOPS を使用して、初期化中に 3,000 IOPS のベースラインパフォーマンスよりも高いパフォーマンスを達成できます。

初期化期間

ボリュームコピーの初期化にかかる時間は、ボリュームコピーの作成時にソースボリュームに書き込まれるブロックデータのサイズに依存します。ボリュームコピーはベストエフォートベースで初期化され、次の一般的なガイドラインが適用されます。データブロックの最初の 1 TiB については、ボリュームの初期化に最大 6 時間かかります。最大 16 TiB のデータブロックの後続の各 1 TiB については、初期化に TiB あたり 1.2 時間かかります。16 TiB を超える書き込みデータについては、初期化に 24 時間かかります。

初期化の進行状況をモニタリングする

describe-volume-status AWS CLI コマンドまたは Amazon EventBridge を使用して、初期化の進行状況をモニタリングできます。詳細については、「Amazon EBS のボリューム初期化のステータスをモニタリングする」を参照してください。

暗号化

ボリュームをコピーする場合、コピーの暗号化は、ソースボリュームの暗号化状態、デフォルト設定によるアカウントの暗号化、コピーリクエストで指定する Encrypted および KmsKeyIdパラメータによって異なります。

暗号化されていないソースボリューム

ソースボリュームが暗号化されておらず、アカウントでデフォルトで暗号化が無効になっている場合、コピーは暗号化されません。コピーを暗号化するには、 Encryptedパラメータを に設定しますtrue。を指定しない場合 KMS key、アカウントのデフォルトの Amazon EBS 暗号化キーが使用されます。

ソースボリュームが暗号化されておらず、アカウントで暗号化がデフォルトで有効になっている場合、コピーは自動的に暗号化されます。を指定しない場合 KMS key、アカウントのデフォルトの Amazon EBS 暗号化キーが使用されます。

暗号化されたソースボリューム

ソースボリュームが暗号化されている場合、コピーは常に暗号化されます。暗号化されたソースボリュームから暗号化されていないコピーを作成することはできません。

同じアカウントのコピーの場合、コピーはデフォルトでソースボリュームと同じ KMS キーを使用します。別のカスタマーマネージドキーを指定して、コピーを再暗号化できます。

クロスアカウントコピーの場合、コピーはデフォルトでアカウントのデフォルトの Amazon EBS 暗号化キーを使用します。KmsKeyId パラメータで別のカスタマーマネージドキーを指定できます。ソースアカウントの KMS キーを使用して、そのキーがアカウントと共有されている場合にボリュームコピーを暗号化できます。

クロスアカウント暗号化の要件

カスタマーマネージドキーで暗号化された共有ボリュームをコピーするには、ボリューム所有者が KMS キーをアカウントと共有する必要があります。詳細については、「 AWS Key Management Service デベロッパーガイド」の「他のアカウントのユーザーに の使用を許可する KMS key」を参照してください。

Amazon EBS のデフォルトの AWS マネージドキーで暗号化されたボリュームは、アカウント間で共有またはコピーできません。

暗号化の結果

暗号化されていないソースボリューム

デフォルトでの暗号化	を指定します。 KMS key	結果
無効	いいえ	コピーが暗号化されていない
無効	はい	指定されたキーで暗号化されたコピー
有効	いいえ	アカウントのデフォルトの Amazon EBS 暗号化キーで暗号化されたコピー
有効	はい	指定されたキーで暗号化されたコピー

暗号化されたソースボリューム (同じアカウント)

を指定します。 KMS key	結果
いいえ	ソースボリュームと同じキーで暗号化されたコピー
はい	指定されたキーで再暗号化されたコピー

暗号化されたソースボリューム (クロスアカウント)

を指定します。 KMS key	結果
いいえ	アカウントのデフォルトの Amazon EBS 暗号化キーで暗号化されたコピー
はい	指定されたキーで暗号化されたコピー

クロスアカウントコピーは、デフォルトではソースアカウントの KMS キーを再利用しません。ソースアカウントのキーを使用する場合は、アカウントと共有する必要があります。

考慮事項

- ボリュームコピーは、ソースボリュームのクラッシュコンシステントなpoint-in-time状態をキャプチャします。これには、コピーリクエスト時にボリュームに書き込まれたデータのみが含まれます。アプリケーションやオペレーティングシステムによってキャッシュされたデータは含まれません。アプリケーション整合性のあるコピーを実現するには、ボリュームへの書き込みを一時停止するか、コピーを作成する前にアプリケーションを休止します。データベースの場合は、適切なメカニズムを使用してダーティページをフラッシュし、I/O をフリーズします (Linux の場合は 、Windows `fsfreeze` の場合は VSS など) 。コピーリクエストが戻ったら、書き込みを再開できます。
- ソースボリュームから一度に作成できるボリュームコピーは 1 つのみです。同じソースボリュームの後続のコピーは、前のボリュームコピーが完全に初期化された後にのみ、作成できます。
- 進行中のボリュームコピーは、リージョンごとに最大 5 つ持つことができます。このクォータを超えた場合、`CopyVolumesLimitExceeded` エラーが発生します。必要に応じて、クォータの引き上げをリクエストできます。
- ソースボリュームと同じアベイラビリティーゾーンにボリュームコピーを作成する必要があります。
- ボリュームコピーのサイズは、ソースボリュームのサイズ以上である必要があります。
- ボリュームコピーを作成または初期化している間は、コピーできません。
- ボリュームコピーを作成するには、ソースボリュームが `available` または `in-use` 状態にあり、ボリュームの変更が `completed` または `optimizing` 状態にある必要があります。
- ボリュームコピーには、通常の Amazon EBS ボリュームと同じアカウント、リージョンストレージ、および IOPS クォータが適用されます。詳細については、Amazon EBS クォータを参照してください。
- コピーオペレーションの進行中にソースボリュームを削除した場合にも、コピーオペレーションは完了します。
- ソースボリュームに割り当てられたタグは、ボリュームコピーに割り当てられません。
- Outposts または Wavelength ゾーンのボリュームからコピーを作成することはできません。

料金

ソースボリュームのコピーを作成すると、コピーオペレーションに対して課金されます。コピーオペレーション料金は、コピーの作成時にソースボリュームに書き込まれたデータブロックのサイズに基づきます。新しく作成されたボリュームは、他の Amazon EBS ボリュームと同じ方法で課金されます。詳細については、Amazon EBS の料金表を参照してください。

ボリュームをコピーする

Amazon EBS ボリュームをコピーするには、次のいずれかの方法を使用します。

Console

ボリュームをコピーするには

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. ナビゲーションペインの [ボリューム] を選択します。
3. コピーするボリュームを選択し、[アクション]、[ボリュームをコピー] の順にクリックします。
4. [ボリュームタイプ] で、コピーのボリュームタイプを選択します。デフォルトのボリュームタイプは gp3 です。
5. [サイズ] に、ボリュームコピーのサイズ (GiB) を入力します。サイズは、ソースボリュームのサイズ以上である必要があります。
6. (*io1*、*io2*、および *gp3* のみ) [IOPS] に、ボリュームコピーの 1 秒あたりの入出力オペレーション (IOPS) の最大数を入力します。
7. (*gp3* のみ) [スループット] に、ボリュームコピーのスループットを MiB/秒単位で入力します。
8. (*io1* および *io2* のみ) Amazon EBS マルチアタッチのボリュームコピーを有効にするには、[マルチアタッチを有効化] を選択します。
9. 暗号化の場合、デフォルトはソースボリュームとアカウント設定によって異なります。同じアカウントのコピーの場合、コピーはソースボリュームの暗号化キーを継承します。クロスアカウントコピーの場合、アカウントのデフォルトの Amazon EBS 暗号化キーが使用されます。別のカスタマーマネージドキーを選択できます。
10. (オプション) ボリュームコピーにカスタムタグを割り当てるには、[タグ] セクションで [タグの追加] を選択し、タグのキーと値のペアを入力します。
11. [ボリュームをコピー] を選択します。
12. コピーされたボリュームは `creating` 状態になり、その後すぐに `available` に移行します。その後、同じアベイラビリティーゾーン内の Amazon EC2 インスタンスにアタッチできます。

AWS CLI

ボリュームをコピーするには

copy-volumes コマンドを使用します。

次の例では、ボリュームタイプが gp3、サイズが 100 GiB、スループットが 250 MiB/秒の vol-01234567890abcdef のボリュームコピーを作成します。

```
aws ec2 copy-volumes \
--source-volume-id vol-01234567890abcdef \
--volume-type gp3 \
--size 100 \
--throughput 250
```

コピー中に暗号化されていないボリュームを暗号化するには

copy-volumes コマンドを --encryptedおよび --kms-key-idパラメータとともに使用します。

```
aws ec2 copy-volumes \
--source-volume-id vol-01234567890abcdef \
--encrypted \
--kms-key-id alias/my-key
```

別のカスタマーマネージドキーで再暗号化するには

copy-volumes コマンドを --kms-key-idパラメータとともに使用して、別のキーを指定します。

```
aws ec2 copy-volumes \
--source-volume-id vol-01234567890abcdef \
--kms-key-id arn:aws:kms:us-east-1:222222222222:key/abcd1234-a123-456a-a12b-
a123b4cd56ef
```

PowerShell

ボリュームをコピーするには

Copy-EC2Volume コマンドレットを使用します。

次の例では、ボリュームタイプが gp3、サイズが 100 GiB、スループットが 250 MiB/秒の vol-01234567890abcdef のボリュームコピーを作成します。

```
Copy-EC2Volume `
-SourceVolumeId vol-01234567890abcdef `
```

```
-VolumeType gp3 `
-Size 100 `
-Throughput 250
```

コピー中に暗号化されていないボリュームを暗号化するには

Copy-EC2Volume コマンドレットを -Encryptedおよび -KmsKeyIdパラメータとともに使用します。

```
Copy-EC2Volume `
-SourceVolumeId vol-01234567890abcdef `
-Encrypted $true `
-KmsKeyId "alias/my-key"
```

別のカスタマーマネージドキーで再暗号化するには

Copy-EC2Volume コマンドレットを -KmsKeyIdパラメータとともに使用して、別のキーを指定します。

```
Copy-EC2Volume `
-SourceVolumeId vol-01234567890abcdef `
-KmsKeyId "arn:aws:kms:us-east-1:222222222222:key/abcd1234-a123-456a-a12b-
a123b4cd56ef"
```

Amazon EBS ボリュームを Amazon EC2 インスタンスにアタッチ

同じアベイラビリティーゾーンに 1 つ以上のインスタンスに、利用可能な EBS ボリュームをボリュームとしてアタッチできます。

起動時に EBS ボリュームをインスタンスに追加する方法については、「インスタンスブロックデバイスマッピング」を参照してください。

考慮事項

- インスタンスにアタッチできる Amazon EBS ボリュームの最大数はインスタンスタイプによって異なります。インスタンスタイプのボリュームアタッチメント制限を超えると、アタッチメントリクエストは AttachmentLimitExceeded エラーで失敗します。詳細については、「インスタンスボリューム数の制限」を参照してください。
- ボリュームは、インスタンスと同じアベイラビリティーゾーンに限りアタッチできます。

- マルチアタッチが有効なボリュームは、最大で 16 個のインスタンスにアタッチできます。詳細については、「マルチアタッチを使用して EBS ボリュームを複数の EC2 インスタンスへアタッチ」を参照してください。
- ボリュームに AWS Marketplace 製品コードがある場合:
  - 停止されたインスタンスのみにアタッチできます。
  - ボリュームにある AWS Marketplace コードをサブスクライブする必要があります。
  - タイプやオペレーティングシステムなどのインスタンスの設定は、その特定の AWS Marketplace コードをサポートしている必要があります。例えば、Windows インスタンスからのボリュームを Linux インスタンスにアタッチすることはできません。
  - AWS Marketplace コードはボリュームからインスタンスにコピーされます。
- 指定するこのデバイス名は Amazon EC2 によって使用されます。ブロックデバイスドライバーは、指定したデバイス名とは異なるデバイス名でデバイスをマウントすることがあります。詳細については、「Amazon EC2 インスタンス上のボリュームのデバイス名」を参照してください。
- 場合によっては、/dev/xvda または /dev/sda にアタッチしたボリューム以外のボリュームが、インスタンスのルートボリュームになることがあります。これは、別のインスタンスのルートボリュームや、ルートボリュームのスナップショットから作成されたボリュームを、既存のルートボリュームのインスタンスにアタッチした場合に起こります。詳細については、「Linux インスタンスが間違ったボリュームから起動する」を参照してください。
- 一部のインスタンスタイプは、複数の EBS カードをサポートしています。EBS カードインデックスを指定することで、アタッチするボリュームの EBS カードを選択できます。インスタンスが複数の EBS カードをサポートしている場合は、「EBS カード」を参照してください。
  - ルートボリュームは EBS カードインデックス にアタッチする必要があります0。
  - 複数の EBS カードをサポートするインスタンスの場合、EBS カードインデックスを指定しない場合、ボリュームは EBS カードインデックス にアタッチされます0。
  - 高性能ワークロード用に EC2 インスタンスを設定する場合、EBS カードのパフォーマンス制限に達しないように、パフォーマンス要件に基づいて EBS カード間で EBS ボリュームのバランスを取ることが重要です。
  - インスタンスタイプのボリュームアタッチメント制限は、各 EBS カードに均等に分散されます。たとえば、2 つの EBS カードを持つ128ボリュームアタッチメントをサポートする EC2 インスタンスでは、各 EBS カードは最大 個の64ボリュームアタッチメントをサポートできます。EBS カードのアタッチメント制限を超えると、リクエストはCardAttachmentLimitExceeded エラーで失敗します。

Console

EBS ボリュームをインスタンスにアタッチするには

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. ナビゲーションペインの [ボリューム] を選択します。
3. アタッチするボリュームを選択し、[Actions] (アクション)、[Attach volume] (ボリュームのアタッチ) の順にクリックします。
4. [Instance] (インスタンス) に、インスタンスの ID を入力するか、オプションのリストからインスタンスを選択します。
5. [デバイス名] で、以下のいずれかを行います。
   - ルートボリュームの場合は、リストの [ルートボリューム用に予約済み] セクションから必要なデバイス名を選択します。通常、AMI に応じて Linux インスタンスは /dev/sda1 または /dev/xvda であり、Windows インスタンスは /dev/sda1 です。
   - データボリュームの場合は、リストの [データボリュームに推奨] セクションから使用可能なデバイス名を選択します。
   - カスタムデバイス名を使用するには、[カスタムデバイス名を指定] を選択し、使用するデバイス名を入力します。
6. [ボリュームのアタッチ] を選択します。
7. インスタンスに接続し、ボリュームをマウントします。詳細については、「Amazon EBS ボリュームを使用できるようにする」を参照してください。

AWS CLI

EBS ボリュームをインスタンスにアタッチするには

attach-volume コマンドを使用します。次の例では、指定されたデバイス名を使用して、指定されたボリュームを指定されたインスタンスにアタッチします。

```
aws ec2 attach-volume \
    --volume-id vol-01234567890abcdef \
    --instance-id i-1234567890abcdef0 \
    --device /dev/sdf
```

PowerShell

EBS ボリュームをインスタンスにアタッチするには

Add-EC2Volume コマンドレットを使用します。次の例では、指定されたデバイス名を使用して、指定されたボリュームを指定されたインスタンスにアタッチします。

```
Add-EC2Volume `
    -VolumeId vol-01234567890abcdef `
    -InstanceId i-1234567890abcdef0 `
    -Device /dev/sdf
```

マルチアタッチを使用して EBS ボリュームを複数の EC2 インスタンスへアタッチ

Amazon EBS マルチアタッチを使用すると、1 つのプロビジョンド IOPS SSD (io1 または io2) ボリュームを、同じアベイラビリティーゾーンにある複数のインスタンスにアタッチできます。複数のマルチアタッチが有効なボリュームを 1 つのインスタンスまたはインスタンスセットにアタッチできます。ボリュームがアタッチされている各インスタンスには、共有ボリュームに対する完全な読み取りおよび書き込みアクセス許可があります。マルチアタッチを使用すると、同時書き込みオペレーションを管理するアプリケーションで、アプリケーションの可用性を高めることが容易になります。

料金と請求

Amazon EBS マルチアタッチの使用に追加料金はかかりません。プロビジョンド IOPS SSD (io1 および io2) ボリュームに適用される標準料金が請求されます。詳細については、Amazon EBS の料金表を参照してください。

内容

- 考慮事項と制限
- マルチアタッチ Amazon EBS ボリュームのパフォーマンス
- Amazon EBS ボリュームのマルチアタッチを有効にする
- Amazon EBS ボリュームのマルチアタッチを無効化
- マルチアタッチが有効になっている Amazon EBS ボリュームで NVMe 予約を使用する

考慮事項と制限

- マルチアタッチが有効なボリュームは、同じアベイラビリティーゾーンにある「Nitro System」に構築された最大 16 のインスタンスにアタッチできます。

- Linux インスタンスは、マルチアタッチが有効な io1 および io2 ボリュームをサポートします。Windows インスタンスは、マルチアタッチが有効な io2 ボリュームのみをサポートします。
- インスタンスにアタッチできる Amazon EBS ボリュームの最大数はインスタンスのタイプとサイズによって異なります。詳細については、「インスタンスボリューム数の制限」を参照してください。
- マルチアタッチは、プロビジョンド IOPS SSD (io1 および io2) ボリュームでのみサポートされます。
- io1 ボリュームのマルチアタッチは、 をサポートするすべてのリージョンで使用できますio1。

  io2 用のマルチアタッチは、io2 をサポートするすべてのリージョンで使用できます。

  > **Note**
  >
  > パフォーマンス、一貫性、耐久性を低コストで向上させるには、io2 ボリュームを使用することをお勧めします。

- XFS や EXT4 などの標準ファイルシステムは、EC2 インスタンスなどの複数のサーバーから同時にアクセスできるように設計されていません。クラスター化されたファイルシステムを使用して、本番環境のワークロードに対してデータの耐障害性と信頼性を確保する必要があります。
- マルチアタッチが有効でio1、 io2ボリュームは I/O フェンシングをサポートします。I/O フェンスプロトコルは、データの一貫性を維持するために、共有ストレージ環境での書き込みアクセスを制御します。アプリケーションは、データの整合性を維持するために、アタッチされたインスタンスの書き込み順序を提供する必要があります。詳細については、「マルチアタッチが有効になっている Amazon EBS ボリュームで NVMe 予約を使用する」を参照してください。
- マルチアタッチが有効なボリュームは、ブートボリュームとして作成できません。
- マルチアタッチ対応のボリュームは、インスタンスあたり 1 つのブロックデバイスマッピングにアタッチできます。
- マルチアタッチは、Amazon EC2 コンソールまたは RunInstances API を使用してインスタンスの起動時に有効にすることはできません。
- Amazon EBS インフラストラクチャレイヤーに問題があるマルチアタッチが有効なボリュームは、アタッチされているすべてのインスタンスで使用できません。Amazon EC2 またはネットワークレイヤーでの問題は、一部のアタッチされたインスタンスにのみ影響する可能性があります。
- 次の表は、作成後にマルチアタッチが有効な io1 および io2 ボリュームに対するボリューム変更サポートを示しています。

	io2 ボリューム	io1 ボリューム
ボリュームタイプの変更	✓ **	✓ **
ボリュームサイズの変更	✓	✓
プロビジョンド IOPS の変更	✓	✓
マルチアタッチの有効化	✓ *	✓ *
マルチアタッチの無効化	✓ *	✓ *

* ボリュームがインスタンスにアタッチされている間は、マルチアタッチを有効または無効にすることはできません。

** マルチアタッチが有効なボリュームのボリュームタイプ変更はio2、ボリュームがアタッチされている間を含め、いずれの方向でも io1と の間でのみサポートされます。その他のボリュームタイプの変更はサポートされていません。

- マルチアタッチが有効なボリュームは、最後にアタッチされたインスタンスが終了し、そのインスタンスが終了時にボリュームを削除するように設定されている場合、インスタンスの終了時に削除されます。ボリュームが複数のインスタンスにアタッチされ、ボリュームブロックデバイスマッピングで終了時の削除設定が異なる場合、最後にアタッチされたインスタンスのブロックデバイスマッピング設定によって、終了時の削除動作が決まります。

  終了時の削除を予測できるようにするには、ボリュームがアタッチされているすべてのインスタンスについて、終了時の削除を有効または無効にします。詳細については、「インスタンスの終了時にデータを保持する」を参照してください。
- Amazon EBS ボリュームの CloudWatch メトリクスを使用して、マルチアタッチが有効なボリュームをモニタリングできます。データは、アタッチされたすべてのインスタンスにわたって集約されます。アタッチされた個々のインスタンスのメトリクスをモニタリングすることはできま

せん。詳細については、「Amazon EBS の Amazon CloudWatch メトリクス」を参照してください。

マルチアタッチ Amazon EBS ボリュームのパフォーマンス

アタッチされた各インスタンスは、ボリュームのプロビジョニングされた最大パフォーマンスまで IOPS の最大パフォーマンスを引き上げます。ただし、アタッチされたすべてのインスタンスの集計パフォーマンスは、ボリュームのプロビジョニングされた最大パフォーマンスを超えることはできません。アタッチされたインスタンスの IOPS に対する需要がボリュームのプロビジョンド IOPS よりも高い場合、ボリュームはプロビジョニングされたパフォーマンスを超えることはありません。

例えば、`80,000` プロビジョンド IOPSで `io2` マルチアタッチ対応のボリュームを作成し、それを `40,000` プロビジョンド IOPS をサポートする `m7g.large` インスタンスと、`60,000` プロビジョンド IOPS をサポートする `r7g.12xlarge` インスタンスにアタッチするとします。各インスタンスは、ボリュームのプロビジョンド IOPS `80,000` を下回るため、最大 IOPS を駆動できます。ただし、両方のインスタンスがボリュームへの I/O を同時に駆動する場合、それらの合計 IOPS は、ボリュームのプロビジョニングされた `80,000` IOPS のパフォーマンスを超えることはできません。

整合性のあるパフォーマンスを実現するには、マルチアタッチが有効なボリュームのセクターにわたって、アタッチされたインスタンスから駆動される I/O のバランスを取ることがベストプラクティスです。

Amazon EC2 インスタンスの IOPS パフォーマンスの詳細については、「Amazon EC2 ユーザーガイド」の「Amazon EBS 最適化インスタンスタイプ」を参照してください。

Amazon EBS ボリュームのマルチアタッチを有効にする

マルチアタッチが有効なボリュームは、他の Amazon EBS ボリュームを管理する場合とほぼ同じ方法で管理できます。ただし、マルチアタッチ機能を使用するには、ボリュームに対してマルチアタッチ機能を有効にする必要があります。

新しいボリュームを作成する場合、マルチアタッチはデフォルトで無効になっています。ボリュームの作成時に、マルチアタッチを有効にできます。

および `io1`io2ボリュームの作成後にマルチアタッチを有効にすることもできますが、インスタンスにアタッチされていない場合に限ります。

ボリュームに対してマルチアタッチを有効にした後は、他の EBS ボリュームへアタッチするのと同じ方法でインスタンスにボリュームをアタッチできます。詳細については、「Amazon EBS ボリュームを Amazon EC2 インスタンスにアタッチ」を参照してください。

Console

ボリューム作成中にマルチアタッチを有効にするには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインの [ボリューム] を選択します。
3. [Create volume] (ボリュームの作成) を選択します。
4. [ボリュームタイプ] で、[プロビジョンド IOPS SSD (**io1**)] または [プロビジョンド IOPS SSD (**io2**)] を選択します。
5. [Size (サイズ)] と [IOPS] で、必要なボリュームサイズとプロビジョニングする IOPS 数を選択します。
6. [アベイラビリティーゾーン] で、インスタンスと同じアベイラビリティーゾーンを選択します。
7. [Amazon EBS Multi-Attach] (Amazon EBS マルチアタッチ) で、[Enable Multi-Attach] (マルチアタッチの有効化) を選択します。
8. (オプション) [Snapshot ID] (スナップショット ID) に、ボリュームの作成元となるスナップショットを選択します。
9. ボリュームの暗号化ステータスを設定します。

   選択したスナップショットが暗号化されている場合、またはアカウントがデフォルトで暗号化を有効にしている場合は、暗号化が自動的に有効になり、無効にすることはできません。ボリュームの暗号化に使用する KMS キーを選択できます。

   選択したスナップショットが暗号化されておらず、アカウントの暗号化がデフォルトで有効になっていない場合、暗号化はオプションです。ボリュームを暗号化するには、[Encryption] (暗号化) で、[Encrypt this volume] (このボリュームを暗号化する) を選択し、次にボリュームの暗号化に使用する KMS キーを選択します。

   暗号化されたボリュームは、Amazon EBS の暗号化をサポートするインスタンスにのみアタッチすることができます。詳細については、「Amazon EBS 暗号化」を参照してください。
10. (オプション) ボリュームにカスタムタグを割り当てるには、タグセクションで [Add tag] (タグの追加) を選択し、タグのキーと値のペアを入力します。
11. [Create volume] (ボリュームの作成) を選択します。

作成後にマルチアタッチを有効にするには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインの [ボリューム] を選択します。
3. ボリュームを選択し、[Actions] (アクション)、[Modify volume] (ボリュームの変更) の順にクリックします。
4. [Amazon EBS Multi-Attach] (Amazon EBS マルチアタッチ) で、[Enable Multi-Attach] (マルチアタッチの有効化) を選択します。
5. [Modify] (変更) を選択します。

AWS CLI

ボリューム作成中にマルチアタッチを有効にするには

create-volume コマンドを使用して、`--multi-attach-enabled` オプションを指定します。

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

作成後にマルチアタッチを有効にするには

modify-volume コマンドを使用して、`--multi-attach-enabled` オプションを指定します。

```
aws ec2 modify-volume \
    --volume-id vol-01234567890abcdef \
    --multi-attach-enabled
```

PowerShell

ボリューム作成中にマルチアタッチを有効にするには

`-MultiAttachEnabled` パラメータで New-EC2Volume コマンドレットを使用します。

```
New-EC2Volume `
    -VolumeType io2 `
    -MultiAttachEnabled $true `
```

```
    -Size 100 `
    -Iops 2000 `
    -Region us-west-2 `
    -AvailabilityZone us-west-2b
```

作成後にマルチアタッチを有効にするには

-MultiAttachEnabled パラメータで Edit-EC2Volume コマンドレットを使用します。

```
Edit-EC2Volume `
    -VolumeId vol-01234567890abcdef `
    -MultiAttachEnabled $true
```

Amazon EBS ボリュームのマルチアタッチを無効化

io1 または io2ボリュームのマルチアタッチは、インスタンスにアタッチされていない場合にのみ無効にできます。

Console

マルチアタッチの作成後に無効にするには

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. ナビゲーションペインの [ボリューム] を選択します。
3. ボリュームを選択し、[Actions] (アクション)、[Modify volume] (ボリュームの変更) の順にクリックします。
4. [Amazon EBS Multi-Attach] (Amazon EBS マルチアタッチ) で、[Enable Multi-Attach] (マルチアタッチを有効化) の選択を解除します。
5. [Modify] (変更) を選択します。

AWS CLI

マルチアタッチの作成後に無効にするには

modify-volume コマンドを使用して、-no-multi-attach-enabled オプションを指定します。

```
aws ec2 modify-volume \
    --volume-id vol-01234567890abcdef \
    --no-multi-attach-enabled
```

PowerShell

マルチアタッチの作成後に無効にするには

`-MultiAttachEnabled` パラメータで Edit-EC2Volume コマンドレットを使用します。

```
Edit-EC2Volume `
    -VolumeId vol-01234567890abcdef `
    -MultiAttachEnabled $false
```

マルチアタッチが有効になっている Amazon EBS ボリュームで NVMe 予約を使用する

マルチアタッチが有効な ボリューム`io1`と `io2`ボリュームは、業界標準のストレージフェンシングプロトコルのセットである NVMe 予約をサポートします。これらのプロトコルにより、複数のインスタンスから共有ボリュームへのアクセスを制御し、調整する予約を、作成および管理できます。共有ストレージアプリケーションは予約を使用して、データの一貫性を維持します。

内容

- 要件
- NVMe 予約のサポートを有効にする
- サポートされている NVMe 予約コマンド
- 料金

要件

NVMe 予約は、マルチアタッチが有効な `io1` および `io2`ボリュームでサポートされています。マルチアタッチが有効なボリュームは、Nitro システムで構築されたインスタンスにのみアタッチできます。

NVMe 予約は、次のオペレーティングシステムでサポートされています。

- SUSE Linux Enterprise 12 SP3 以降
- RHEL 8.3 以降
- Amazon Linux 2 以降
- Windows Server 2016 以降

> **Note**
>
> サポートされている、2023年 9 月 13 日以降の Windows Server AMI には、必要な NVMe ドライバーが含まれています。それ以前の AMI では、NVMe ドライバーバージョンを 1.5.0 以降に更新する必要があります。詳細については、「AWS NVMe ドライバー」を参照してください。

EC2Launch v2 を使用してディスクを初期化する場合は、バージョン 2.0.1521 以降にアップグレードする必要があります。詳細については、「EC2 Windows インスタンスの起動時に EC2Launch v2 エージェントを使用してタスクを実行する」を参照してください。

NVMe 予約のサポートを有効にする

NVMe 予約のサポートは、2023 年 9 月 18 日以降に作成されたすべてのマルチアタッチ対応io2ボリュームと、2026 年 10 月 1 日以降に作成されたすべてのマルチアタッチ対応io1ボリュームでデフォルトで有効になっています。

2023 年 9 月 18 日より前に作成された既存のio2ボリューム、または 2026 年 10 月 1 日より前に作成された既存のio1ボリュームの NVMe 予約のサポートを有効にするには、ボリュームからすべてのインスタンスをデタッチしてから、必要なインスタンスを再アタッチする必要があります。すべてのインスタンスをデタッチした後にアタッチを行うと、NVMe 予約が有効になります。

サポートされている NVMe 予約コマンド

Amazon EBS は、次の NVMe 予約コマンドをサポートしています。

予約登録

予約キーの登録と解除、または置き換えを行います。登録キーによりインスタンスを識別、または認証します。予約キーをボリュームに登録すると、インスタンスとボリュームが関連付けられます。インスタンスが予約を取得するには、インスタンスをボリュームに登録する必要があります。

予約取得

ボリュームの予約の取得、名前空間による先行予約の保持、ボリュームで保持している予約の中止を行います。次の予約タイプを取得できます。

- 排他的書き込み予約
- 排他的アクセス予約

- 排他的書き込み - 登録者限定予約
- 排他的アクセス - 登録者限定予約
- 排他的書き込み - 全登録者用予約
- 排他的アクセス - 全登録者用予約

予約解除

ボリュームに保持されている予約を、解除またはクリアします。

予約レポート

ボリュームの登録状況と予約状況について説明します。

料金

マルチアタッチを有効にする際、または使用する際に、追加料金はかかりません。

Amazon EBS ボリュームを使用できるようにする

Amazon EBS ボリュームをインスタンスにアタッチすると、ブロックデバイスとして公開されます。任意のファイルシステムでボリュームをフォーマットし、マウントできます。EBS ボリュームを使用できるようにすると、他のボリュームと同じようにアクセスできます。このファイルシステムに書き込まれるデータはすべて EBS ボリュームに書き込まれますが、デバイスを使用するアプリケーションには透過的になります。

EBS ボリュームのスナップショットは、バックアップ目的で作成したり、別のボリュームを作成する際のベースラインとして使用したりできます。詳細については、「Amazon EBS スナップショット」を参照してください。

使用準備中の EBS ボリュームが 2 TiB を超える場合は、GPT パーティショニングスキームを使用してボリューム全体にアクセスする必要があります。詳細については、「Amazon EBS ボリュームの制約」を参照してください。

Linux インスタンス

アタッチ済みボリュームのフォーマットとマウント

ルートデバイス用の EBS ボリューム /dev/xvda を持つ EC2 インスタンスがあり、/dev/sdf を使用して空の EBS ボリュームをインスタンスにアタッチしたとします。新たなアタッチ済みボリュームを使用するには、次の手順を使用します。

Linux で EBS ボリュームをフォーマットしてマウントするには

1. SSH を使用してインスタンスに接続します。詳細については、「Linux インスタンスへの接続」を参照してください。
2. ブロックデバイスマッピングで指定したものとは異なるデバイス名を使用して、デバイスをインスタンスにアタッチすることができます。詳細については、「Linux インスタンスでのデバイス名」を参照してください。lsblk コマンドを使用して、使用可能なディスクデバイスとマウントポイント (該当する場合) を表示し、使用する正しいデバイス名を決定します。lsblk の出力は、フルデバイスパスから /dev/ プレフィクスを削除します。

   次の内容は、EBS ボリュームを NVMe ブロックデバイスとして公開する「Nitro System」で構築されたインスタンスの出力例を示します。ルートデバイス/dev/nvme0n1には、nvme0n1p1およびnvme0n1p128という名前の 2 つのパーティションがあります。アタッチされているボリュームは /dev/nvme1n1 パーティションがなく、まだマウントされていません。

   ```
   [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
   ```

   以下は T2 インスタンスの出力例です。ルートデバイス/dev/xvdaには、xvda1という名前のパーティションが 1 つあります。アタッチされているボリュームは /dev/xvdf パーティションがなく、まだマウントされていません。

   ```
   [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. ボリュームにファイルシステムがあるかどうかを確認します。新しいボリュームは未加工のブロックデバイスであるため、マウントして使用する前に、ボリュームにファイルシステムを作成する必要があります。スナップショットから作成されたボリュームは、多くの場合既にファイルシステムがあります。既存のファイルシステムの上に新しいファイルシステムを作成すると、データが上書きされます。

次の方法のいずれか (または両方) を使用して、ボリューム上にファイルシステムがあるかどうかを判断します。

- file -s コマンドを使用して、ファイルシステムタイプなど、特定のデバイスに関する情報を取得します。次の例のように、出力に data だけが表示されている場合は、デバイスにはファイルシステムが存在していません。

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

  デバイスにファイルシステムがある場合は、ファイルシステムの種類に関する情報が表示されます。例えば、次の出力は XFS ファイルシステムを持つルートデバイスを示しています。

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

- lsblk -f コマンドを使用して、インスタンスにアタッチされているすべてのデバイスに関する情報を取得します。

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

  例えば、次の出力例では、3 つのデバイス — nvme1n1、nvme0n1、および nvme2n1 がインスタンスにアタッチされていることを示しています。最初の列には、デバイスとそのパーティションが一覧表示されています。FSTYPE 列には、各デバイスのファイルシステムタイプが表示されています。特定のデバイスについての列が空の場合は、そのデバイスにファイルシステムがないことを意味します。この場合、デバイス nvme1n1 とデバイス nvme0n1 上のパーティション nvme0n1p1 はともに XFS ファイルシステムを使用してフォーマットされていますが、デバイス nvme2n1 とデバイス nvme0n1 上のパーティション nvme0n1p128 にはファイルシステムがありません。

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

これらのコマンドの出力で、デバイス上にファイルシステムがないことが示された場合は、ファイルシステムを作成する必要があります。

4. (条件付き) 前の手順でデバイスにファイルシステムがあることがわかった場合は、このステップをスキップしてください。ボリュームが空の場合は、mkfs -t コマンドを使用し、そのボリューム上にファイルシステムを作成します。

   > **Warning**
   >
   > すでにデータが入っているボリューム (例: スナップショットから作成されたボリューム) をマウントしている場合は、このコマンドを使用しないでください。そうでない場合、ボリュームがフォーマットされ、既存のデータが削除されます。

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

   mkfs.xfs がないというエラーが表示された場合は、次のコマンドを使用して XFS ツールをインストールしてから、前述のコマンドを繰り返します。

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

5. mkdir コマンドを使用して、ボリュームのマウントポイントディレクトリを作成します。マウントポイントとは、ボリュームをマウントした後、ファイルシステムツリー内でボリュームが配置され、ファイルの読み書きが実行される場所です。次の例では、/data という名前のディレクトリが作成されます。

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

6. 前のステップで作成したマウントポイントディレクトリにボリュームまたはパーティションをマウントします。

   ボリュームにパーティションがない場合は、次のコマンドを実行してデバイス名を指定し、ボリューム全体をマウントします。

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

ボリュームにパーティションがある場合は、次のコマンドを実行してパーティション名を指定し、パーティションをマウントします。

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

7. 新しいボリュームマウントのファイルのアクセス許可をプレビューして、ユーザーとアプリケーションがボリュームに書き込みできることを確認します。ファイルのアクセス許可の詳細については、Linux Documentation Project のファイルセキュリティを参照してください。
8. インスタンスを再起動した後にマウントポイントが自動的に保存されることはありません。再起動後にこの EBS ボリュームを自動的にマウントするには、次の手順を参照してください。

再起動後に接続ボリュームを自動的にマウントする

システムブート時に常に、このアタッチ済みの EBS ボリュームをマウントするには、/etc/fstab ファイルにデバイス用のエントリを追加します。

/etc/fstab でシステムの現在のデバイス名 (/dev/xvdf など) は使用できますが、代わりにデバイスの 128 ビット汎用一意識別子 (UUID) を使用することをお勧めします。デバイス名は変更される可能性がありますが、UUID はパーティションの存続期間を通じて持続します。UUID を使用することで、ハードウェアの再構成後にシステムが起動できなくなる可能性を減らすことができます。詳細については、Amazon EBS ボリュームを NVMe デバイス名にマッピングを参照してください。

再起動後に接続ボリュームを自動的にマウントするには

1. (オプション) /etc/fstab ファイルのバックアップコピーを作成すると、編集中に誤って破壊/削除してしまった場合にこのコピーを使用できます。

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

2. blkid コマンドを使用してデバイスの UUID を見つけます。再起動後にマウントするデバイスの UUID を書き留めます。次の手順で必要になります。

   例えば、次のコマンドは、インスタンスにマウントされている 2 つのデバイスがあることを示し、両方のデバイスの UUID を表示します。

   ```
   [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"
```

Ubuntu 18.04 では、lsblk コマンドを使用します。

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

3. nano や vim などのテキストエディタを使用して、/etc/fstab ファイルを開きます。

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

4. 指定されたマウントポイントにデバイスをマウントするために、/etc/fstab に次のエントリを追加します。フィールドは、blkid (Ubuntu 18.04 の場合は lsblk) から返される UUID 値、マウントポイント、ファイルシステム、および推奨されるファイルシステムマウントオプションです。必須フィールドの詳細については、man fstab を実行して fstab マニュアルを開きます。

   次の例では、UUID を使用してデバイスをマウントポイントaebf131c-6957-451e-8d34-ec978d9581aeにマウント/dataし、 xfs ファイルシステムを使用します。また、 フラグdefaultsと nofailフラグも使用します。ファイルシステムがダンプされない0ように を指定し、非ルートデバイスであることを示す2ように を指定します。

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

> **Note**
>
> このボリュームをアタッチしないでインスタンスを起動することを目的としている場合 (例えば、ボリュームを別のインスタンスに移動した後)、nofail マウントオプションを追加し、ボリュームのマウントでエラーが発生してもインスタンスが起動できるようにしてください。また、Debian から派生した OS (16.04 より前の Ubuntu バージョンなど) では、nobootwait マウントオプションを追加する必要があります。

5. 入力内容が正しいことを確認するには、次のコマンドを実行してデバイスをアンマウントし、すべてのファイルシステムを /etc/fstab にマウントします。エラーがなければ、/etc/fstab ファイルは問題ありません。ファイルシステムは再起動後に自動的にマウントされます。

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

エラーメッセージが表示されたら、ファイル内のエラーに対処してください。

> **Warning**
>
> /etc/fstab ファイルにエラーがあると、システムがブート不能になる可能性があります。/etc/fstab ファイルにエラーがあるシステムをシャットダウンしないでください。

/etc/fstab のエラーを修正する方法がわからず、このステップの最初のステップでバックアップファイルを作成した場合は、次のコマンドを使用してバックアップファイルから復元できます。

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

Windows インスタンス

次のいずれかの方法を使用し、Windows インスタンスでボリュームを使用できるようにします。

PowerShell

raw パーティションを持つすべての EBS ボリュームを Windows PowerShell で使用できるようにするには

1. リモートデスクトップを使用して Windows インスタンスにログインします。詳細については、「Windows インスタンスに接続する」を参照してください。
2. タスクバーで、[Start] (スタート) メニューを開き、[Windows PowerShell] を選択します。
3. 開いた PowerShell プロンプト内で、提供されている一連の Windows PowerShell コマンドを使用します。このスクリプトはデフォルトで次のアクションを実行します。
    1. ShellHWDetection サービスを停止します。
    2. パーティションスタイルが raw のディスクを列挙します。
    3. ディスクとパーティションタイプがサポートする最大サイズにまたがる新しいパーティションを作成します。
    4. 使用できるドライブ文字を割り当てます。
    5. 指定されたファイルシステムラベルを使用して、ファイルシステムを NTFS としてフォーマットします。
    6. ShellHWDetection サービスを再起動します。

```
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

DiskPart コマンドラインツールを使用して EBS ボリュームを使用可能にするには

1. リモートデスクトップを使用して Windows インスタンスにログインします。詳細については、「Windows インスタンスに接続する」を参照してください。
2. 使用可能にするディスク番号を決定します。
   1. [Start] (スタート) メニューを開き、[Windows PowerShell] を選択します。
   2. `Get-Disk` コマンドレットを使用して、使用できるディスクのリストを取得します。
   3. コマンドの出力で、使用可能にするディスクに対応する[Number] (番号) を書き留めます。
3. DiskPart コマンドを実行するスクリプトファイルを次のように作成します。
   1. [Start] (スタート) メニューを開き、[File Explorer] (ファイルエクスプローラ) を選択します。
   2. C:\ などのディレクトリに移動し、スクリプトファイルを保存します。
   3. フォルダ内の空のスペースを選択 (右クリック) してダイアログボックスを開き、New にカーソルを合わせてコンテキストメニューにアクセスし、Text Document を選択します。
   4. テキストファイルの名前を `diskpart.txt` にします。
4. 次のコマンドをスクリプトファイルに追加します。ディスク番号、パーティションタイプ、ボリュームラベル、ドライブ文字の変更が必要になる場合があります。このスクリプトはデフォルトで次のアクションを実行します。
   1. 修正するディスク 1 を選択します。
   2. マスターブートレコード (MBR) パーティション構造を使用するようにボリュームを設定します。
   3. ボリュームを NTFS ボリュームとしてフォーマットします。

4. ボリュームラベルを設定します。
5. ボリュームにドライブ文字を割り当てます。

> **Warning**
>
> 既存のデータがあるボリュームをマウントする場合は、ボリュームを再フォーマットしないでください。再フォーマットすると、既存のデータが削除されます。

```
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"
```

詳細については、DiskPart Syntax and Parametersを参照してください。

5. コマンドプロンプトを開き、スクリプトがあるフォルダに移動し、次のコマンドを実行して、指定したディスクでボリュームを使用できるようにします。

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

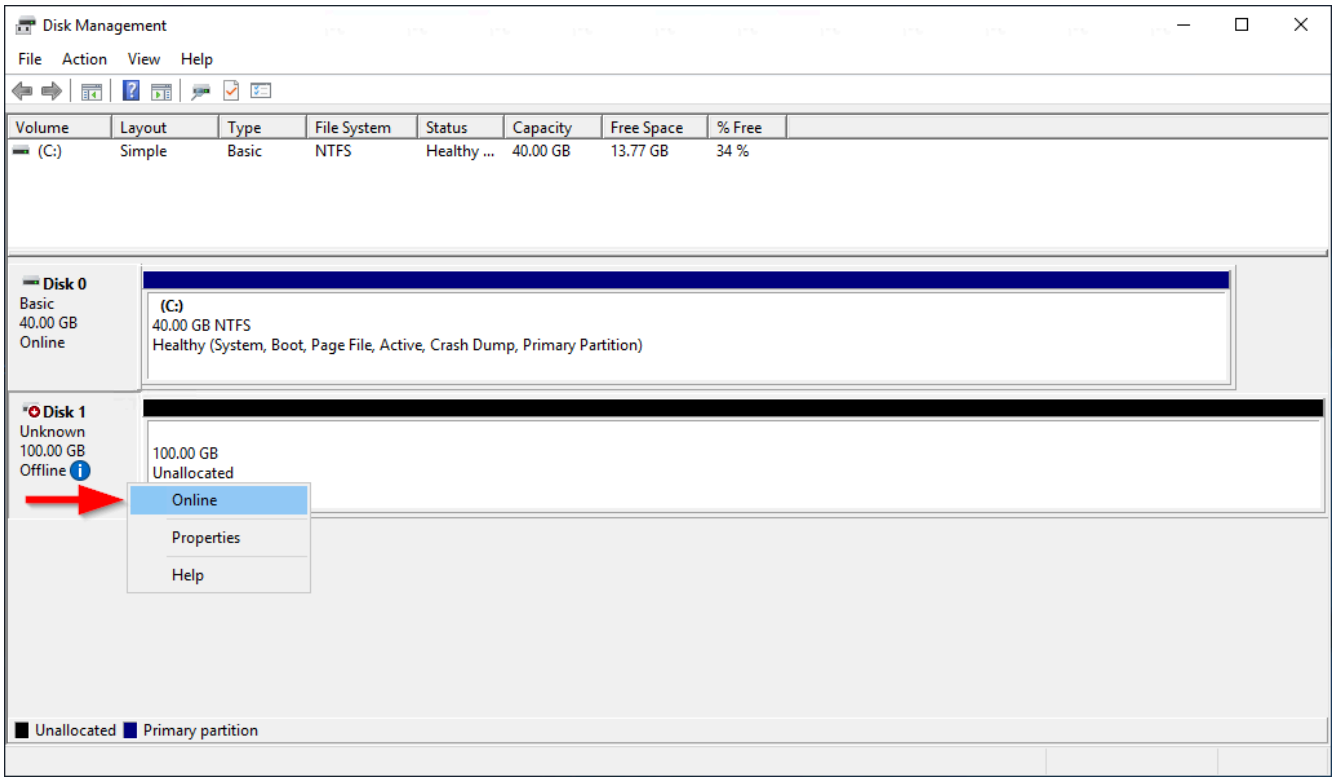
Disk Management utility

ディスク管理ユーティリティで EBS ボリュームを使用可能にするには

1. リモートデスクトップを使用して Windows インスタンスにログインします。詳細については、「Windows インスタンスに接続する」を参照してください。
2. [Disk Management] ユーティリティを起動します。タスクバーで、Windows ロゴのコンテキスト (右クリック) メニューを開き、[Disk Management] (ディスクの管理) を選択します。

> **Note**
>
> Windows Server 2008 では、[スタート]、[管理ツール]、[コンピュータの管理]、[ディスクの管理] の順に選択します。

3. ボリュームをオンライン状態にします。下のペインで、EBS ボリューム用のディスクの左パネルのコンテキスト (右クリック) メニューを開きます。[Online] を選択します。

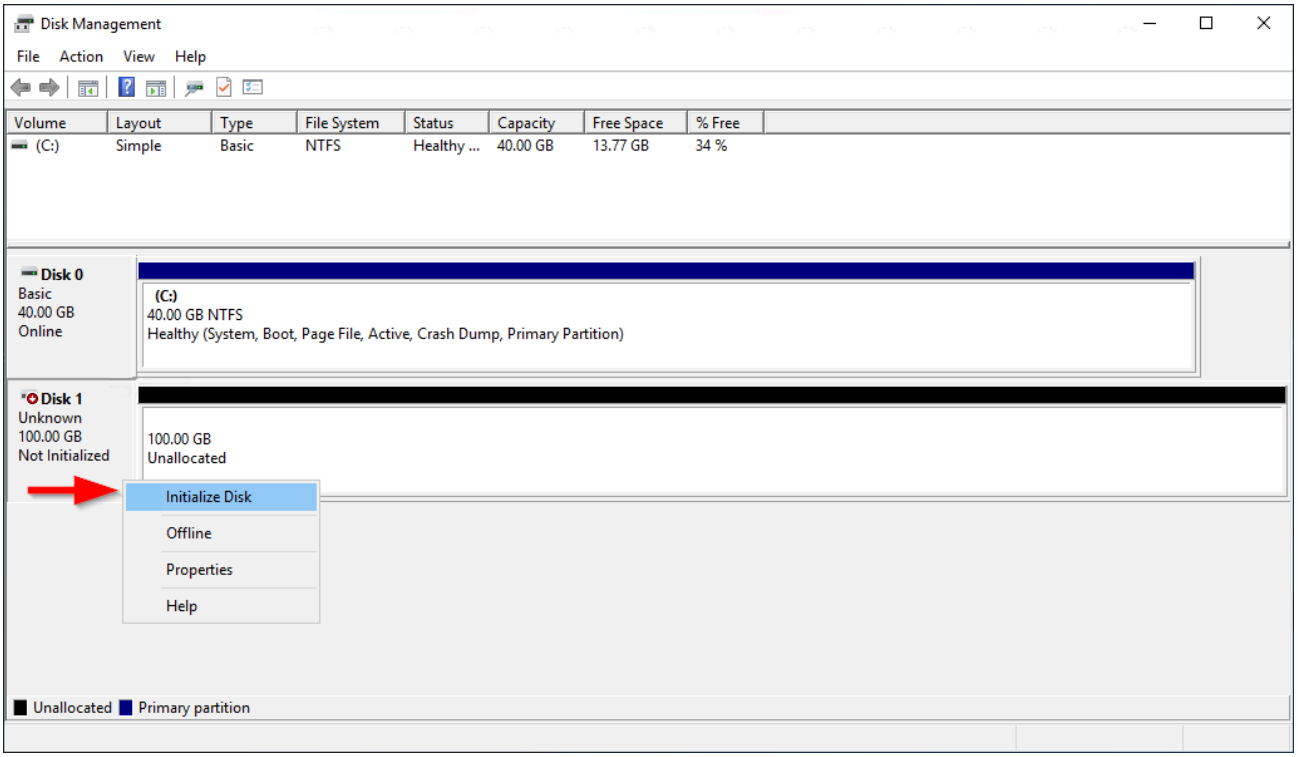


4. (条件に応じて) ディスクが初期化されていない場合は、使用する前に初期化する必要があります。ディスクが既に初期化されている場合は、このステップをスキップします。

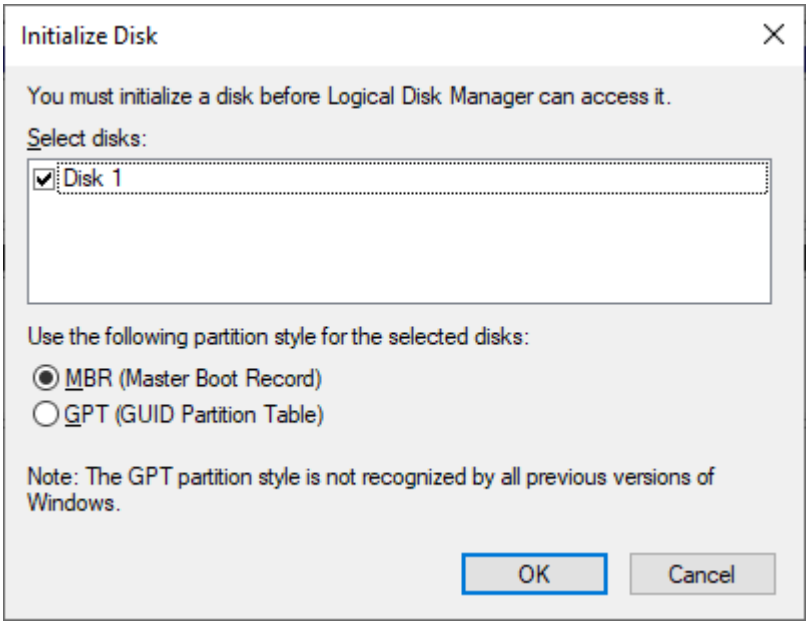
> **Warning**
>
> すでにデータが含まれるボリューム (パブリックデータセット、またはスナップショットから作成したボリュームなど) をマウントする場合は、ボリュームを再フォーマットしないように注意してください。再フォーマットすると、既存のデータが削除されます。

ディスクが初期化されていない場合は、次のように初期化します。

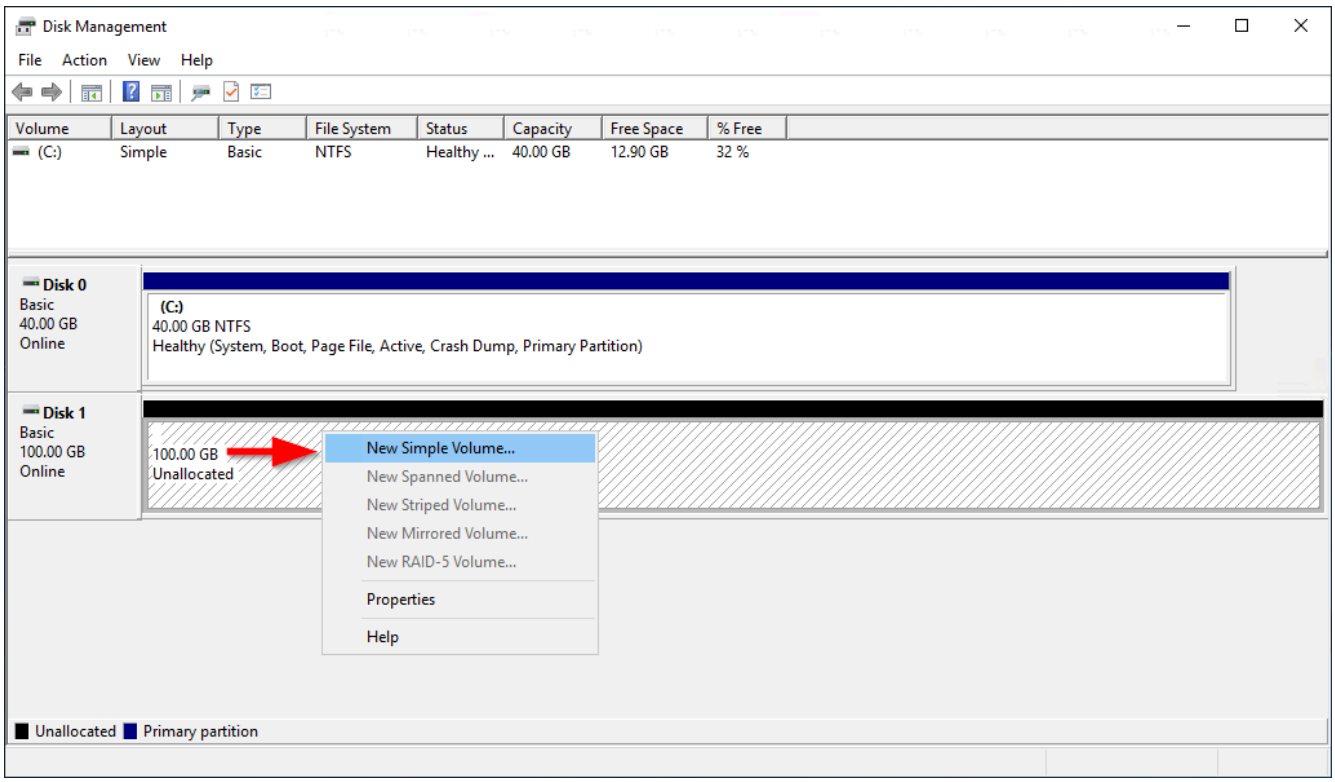
1. ディスクの左側パネルのコンテキスト (右クリック) メニューを開き、[Initialize Disk] (ディスクの初期化) を選択します。



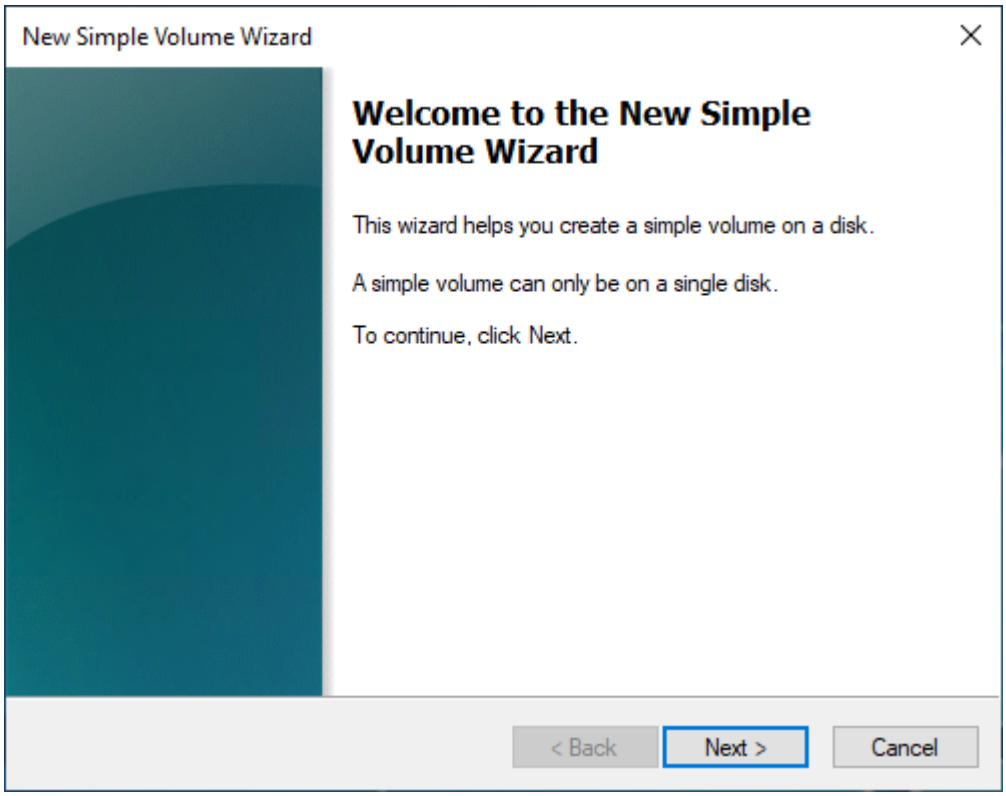
2. [Initialize Disk] (ディスクの初期化) ダイアログボックスで、パーティションスタイルを選択し、[OK] をクリックします。



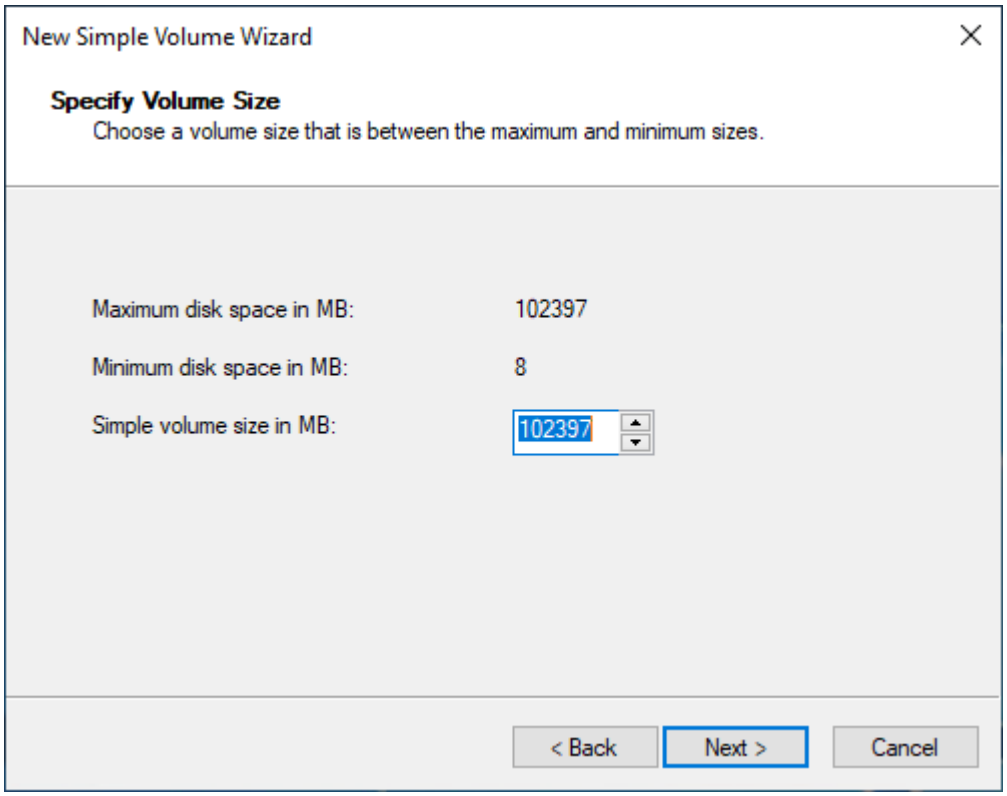
5. ディスクの右パネルのコンテキスト (右クリック) メニューを開き、[New Simple Volume] (新しいシンプルボリューム) を選択します。



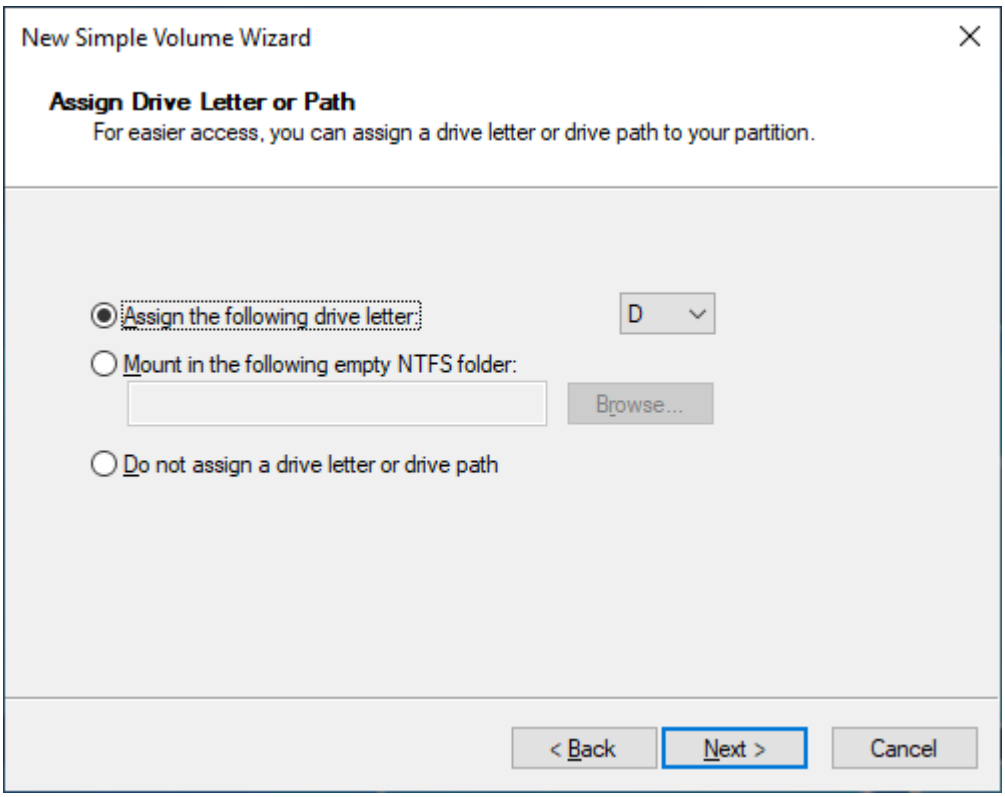
6. [New Simple Volume Wizard] (新しいシンプルボリュームウィザード) で、[Next] (次へ) をクリックします。



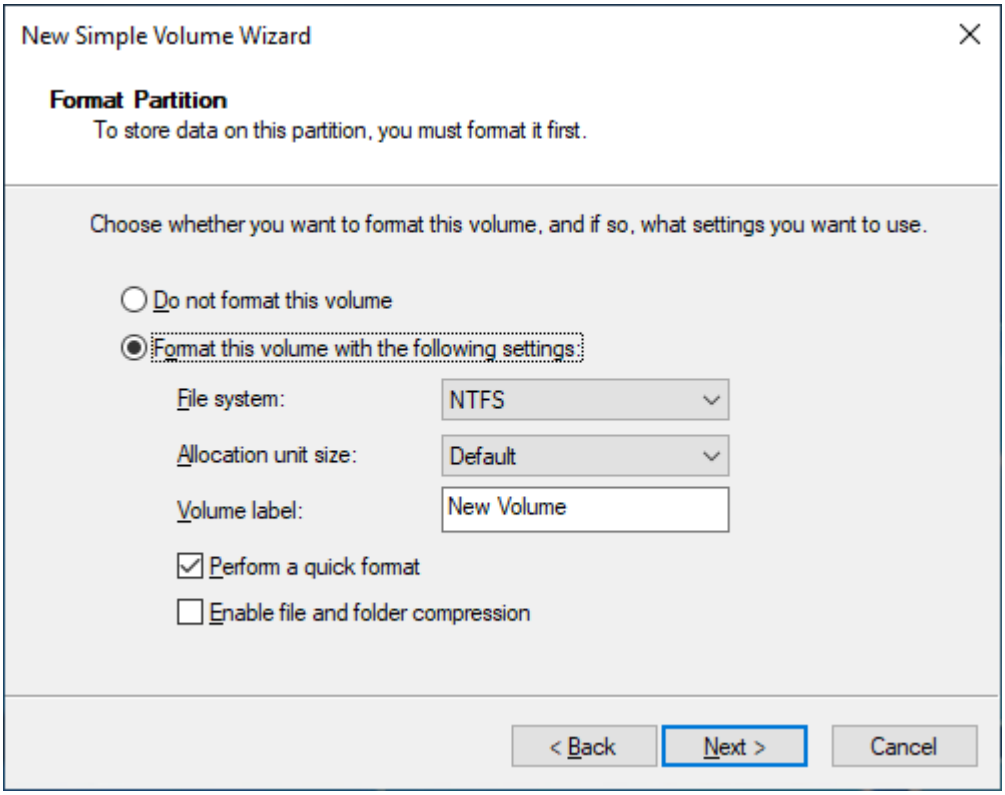
7. デフォルトの最大値を変更する場合は、[Simple volume size in MB] (シンプルボリュームサイズ (MB)) を選択してから、[Next] (次へ) をクリックします。



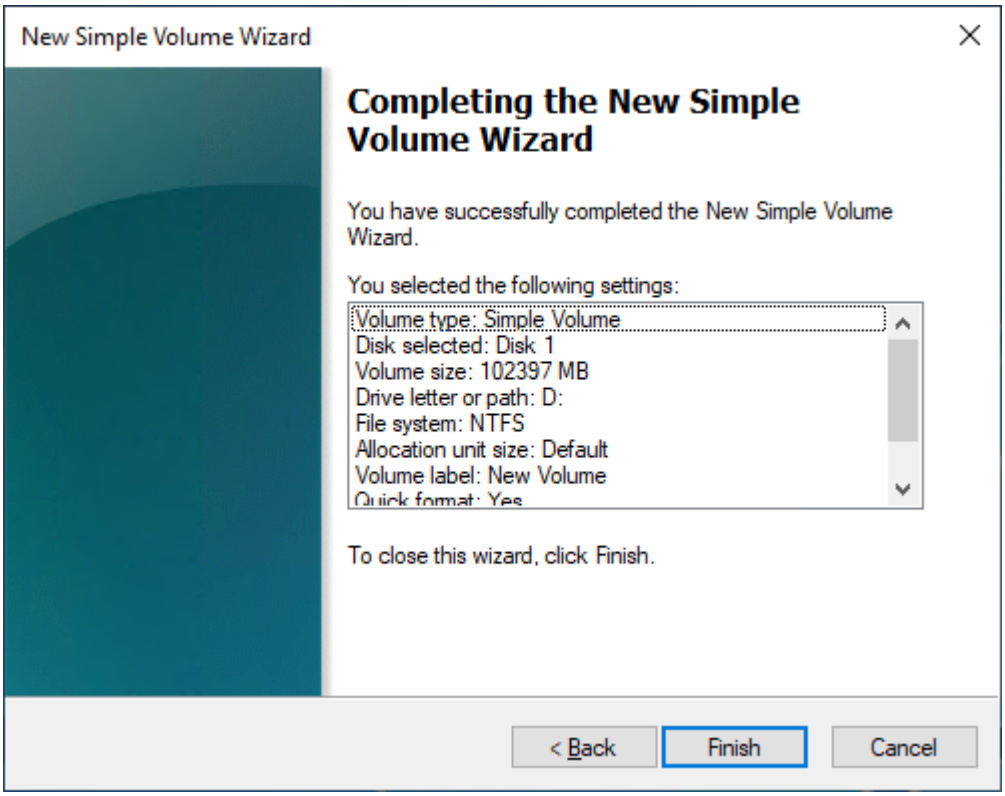
8. 必要に応じて、使用したいドライブ文字を [Assign the following drive letter] (次のドライブ文字を割り当る) ドロップダウンの中に指定し、[Next] (次へ) をクリックします。



9. [Volume Label] (ボリュームラベル) を指定し、必要に応じてデフォルト設定を調整し、[Next] (次へ) をクリックします。



10. 設定を確認してから、[Finish] (終了) をクリックして変更を適用し、[New Simple Volume] (新しいシンプルボリューム) ウィザードを閉じます。



Amazon EBS ボリュームに関する情報の表示

EBS ボリュームに関する詳細情報を表示できます。例えば、特定のリージョンの全てのボリュームの情報を表示したり、単一ボリュームの詳細 (サイズ、ボリュームタイプ、ボリュームが暗号化されているかどうか、ボリュームを暗号化するために使用した KMS キー、ボリュームがアタッチされている特定のインスタンスなど) を表示することができます。

インスタンスのオペレーティングシステムから、どのくらいのディスク容量が使用可能かなどの EBS ボリュームの詳細情報を取得できます。

内容

- ボリューム情報の表示
- ボリューム状態
- ボリュームメトリクスの表示
- 空きディスク容量の表示

ボリューム情報の表示

EBS ボリュームに関する情報を表示できます。

Console

ボリュームに関する情報を表示するには

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. ナビゲーションペインの [ボリューム] を選択します。
3. リストを減らすには、タグとボリューム属性を使用してボリュームをフィルターできます。フィルターフィールドを選択し、タグまたはボリューム属性を選択し、フィルタ値を選択します。
4. ボリュームの詳細情報を表示するには、そのボリュームを選択します。

インスタンスにアタッチされている EBS ボリュームを表示するには

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. ナビゲーションペインで、[インスタンス] を選択してください。
3. インスタンスを選択します。
4. [Storage] (ストレージ) タブの [Block devices] (ブロックデバイス) セクションでは、インスタンスにアタッチされているボリュームの一覧が表示されます。特定のボリュームに関する情報を表示するには、[Volume ID] (ボリューム ID) 列で該当する ID を選択します。

Amazon EC2 Global View

Amazon EC2 Global View を使用して、 AWS アカウントが有効になっているすべてのリージョンでボリュームを表示できます。

すべてのリージョンの EBS ボリュームの概要を表示するには

1. Amazon EC2 Global View コンソール (https://console.aws.amazon.com/ec2globalview/home) を開きます。
2. [リージョンエクスプローラー] タブの [概要] で、[ボリューム] のリソース数を確認します。これには、ボリューム数とリージョン数が含まれており、下線付きのテキストを選択すると、ボリューム数がリージョン間でどのように分散されるかが表示されます。

3. [グローバル検索] タブで、クライアントフィルター [リソースタイプ = ボリューム] を選択します。リージョンまたはタグを指定することで、結果をさらにフィルタリングできます。

AWS CLI

EBS ボリュームに関する情報を表示するには

describe-volumes コマンドを使用します。次の例では、現在のリージョンのボリュームをカウントします。

```
aws ec2 describe-volumes --query "length(Volumes[*])"
```

次の例では、指定されたインスタンスにアタッチされたボリュームを一覧表示します。

```
aws ec2 describe-volumes \
    --filters "Name=attachment.instance-id,Values=i-1234567890abcdef0" \
    --query Volumes[*].VolumeId \
    --output text
```

次の例では、指定したボリュームについて説明します。

```
aws ec2 describe-volumes --volume-ids vol-01234567890abcdef
```

以下は出力の例です。

```
{
    "Volumes": [
        {
            "Iops": 3000,
            "VolumeType": "gp3",
            "MultiAttachEnabled": false,
            "Throughput": 125,
            "Operator": {
                "Managed": false
            },
            "VolumeId": "vol-01234567890abcdef",
            "Size": 8,
            "SnapshotId": "snap-0abcdef1234567890",
            "AvailabilityZone": "us-west-2b",
            "State": "in-use",
            "CreateTime": "2024-05-17T23:23:00.400000+00:00",
```

```
            "Attachments": [
                {
                    "DeleteOnTermination": true,
                    "VolumeId": "vol-01234567890abcdef",
                    "InstanceId": "i-1234567890abcdef0",
                    "Device": "/dev/xvda",
                    "State": "attached",
                    "AttachTime": "2024-05-17T23:23:00+00:00"
                }
            ],
            "Encrypted": false
        }
    ]
}
```

PowerShell

EBS ボリュームに関する情報を表示するには

Get-EC2Volume コマンドレットを使用します。次の例では、現在のリージョンのボリュームをカウントします。

```
(Get-EC2Volume).Count
```

次の例では、指定されたインスタンスにアタッチされたボリュームを一覧表示します。

```
(Get-EC2Volume `
    -Filters @{Name="attachment.instance-id";Values="i-1234567890abcdef0"}).VolumeId
```

次の例では、指定したボリュームについて説明します。

```
Get-EC2Volume -VolumeId vol-01234567890abcdef
```

以下は出力の例です。

```
Attachments        : {i-1234567890abcdef0}
AvailabilityZone   : us-west-2b
CreateTime         : 5/17/2024 11:23:00 PM
Encrypted          : False
FastRestored       : False
Iops               : 3000
```

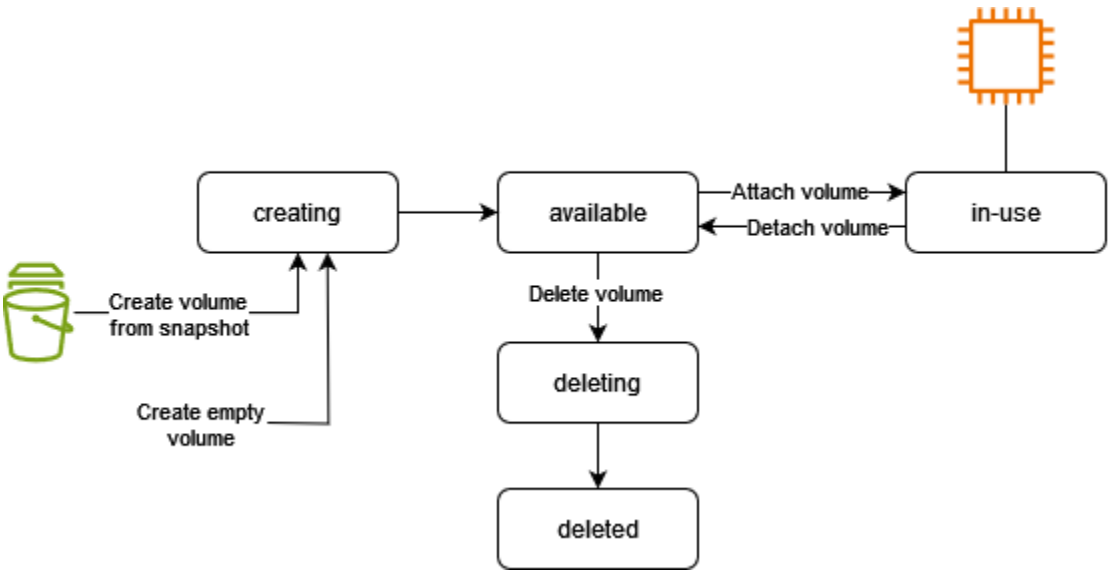
```
KmsKeyId           :
MultiAttachEnabled : False
Operator           : Amazon.EC2.Model.OperatorResponse
OutpostArn         :
Size               : 8
SnapshotId         : snap-0abcdef1234567890
SseType            :
State              : in-use
Tags               : {}
Throughput         : 125
VolumeId           : vol-01234567890abcdef
VolumeType         : gp3
```

ボリューム状態

ボリューム状態は、Amazon EBS ボリュームの可用性を示します。ボリュームの状態は、コンソールのボリュームページのステート列で表示することも、describe-volumes AWS CLI コマンドを使用して表示することもできます。

Amazon EBS ボリュームは、作成されてから削除されるまで、さまざまな状態に移行します。

次の図は、ボリュームの状態間の移行を示しています。Amazon EBS スナップショットからボリュームを作成するか、空のボリュームを作成できます。ボリュームを作成すると、`creating` 状態になります。ボリュームが使用可能になると、`available` 状態になります。ボリュームと同じアベイラビリティーゾーンにあるインスタンスに、利用可能なボリュームをアタッチできます。ボリュームを別のインスタンスにアタッチまたは削除する前に、ボリュームをデタッチする必要があります。不要になったボリュームは削除できます。



次の表はボリューム状態をまとめたものです。

State	説明
creating	ボリュームは作成中です。
available	ボリュームはインスタンスにアタッチされていません。
in-use	ボリュームはインスタンスにアタッチされています。
deleting	ボリュームは削除中です。
deleted	ボリュームは削除されました。
error	EBS ボリュームに関連する基となるハードウェアに障害が発生し、ボリュームに関連付けられているデータを回復できません。ボリュームを復元する方法またはボリューム上のデータを回復する方法については、「EBS ボリュームのステータスが「エラー」になるのはなぜですか?」を参照してください。

ボリュームメトリクスの表示

EBS ボリュームに関する詳細は、Amazon CloudWatch から入手できます。詳細については、Amazon EBS の Amazon CloudWatch メトリクスを参照してください。

空きディスク容量の表示

インスタンスのオペレーティングシステムから、どのくらいのディスク容量が使用可能かなどの EBS ボリュームの詳細情報を取得できます。

Linux インスタンス

df -hT コマンドを使用して、デバイス名を指定します。

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

Windows インスタンス

エクスプローラーを開き、[この PC] を選択して、空きディスク容量を表示します。

次の dir コマンドを使用して、出力の最後の行を確認することで空きディスク容量を表示することもできます。

```
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
```

次の fsutil コマンドを使用して、空きディスク容量を表示することもできます。

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

> **Tip**
>
> CloudWatch エージェントを使用して、インスタンスに接続せずに Amazon EC2 インスタンスからディスクスペース使用量のメトリクスを収集することもできます。詳細については、「Amazon CloudWatch ユーザーガイド」の「CloudWatch エージェント設定ファイルを作成する」および「CloudWatch エージェントのインストール」を参照してください。複数のインスタンスのディスクスペース使用量をモニタリングする必要がある場合は、Systems Manager を使用してそれらのインスタンスに CloudWatch エージェントをインストールして設定できます。詳細については、「Installing the CloudWatch agent using Systems

Manager」(Systems Manager を使用した CloudWatch エージェントのインストール) を参照してください。

Elastic Volumes オペレーションを使用して Amazon EBS ボリュームを変更する

Amazon EBS Elastic Volumes では、EBS ボリュームのボリュームサイズの増加、ボリュームタイプの変更、パフォーマンスの調整を行うことができます。インスタンスで Elastic Volumes をサポートしている場合は、ボリュームのデタッチやインスタンスの再起動を行うことなく、これらの操作を行うことができます。したがって、変更の適用中でも、アプリケーションを引き続き使用できます。

ボリュームの設定を変更するための料金は発生しません。ボリューム変更を開始すると、新しいボリューム設定料金が発生します。詳細については、Amazon EBS 料金表ページを参照してください。

内容

- 考慮事項
- 制限事項
- Amazon EBS ボリューム変更の要件
- Amazon EBS ボリュームの変更をリクエスト
- Amazon EBS ボリューム変更の進行状況のモニタリング
- Amazon EBS ボリュームのサイズ変更後にファイルシステムを拡張

考慮事項

- ボリュームの変更を開始したら、その変更が `completed`状態になるまで待ってから、同じボリュームに対して別の変更を開始する必要があります。ボリュームが `in-use`または `available`状態であり、そのボリュームの以前のすべての変更が である限り、ローリング 24 時間以内に最大 4 回ボリュームを変更できます`completed`。この制限を超えると、次の変更を実行できるタイミングを示すエラーメッセージが表示されます。
- ボリュームの変更はベストエフォートベースで実行され、リクエストされたボリューム設定によっては完了までに数分から数時間かかる場合があります。通常、1-TiB ボリュームの変更には最大 6 時間かかることがあります。ただし、時間は常にボリュームサイズに合わせて直線的にスケールするとは限りません。ボリュームが大きいほど時間がかかり、ボリュームが小さいほど時間が長くなる可能性があります。

- サイズの増加は、ボリュームの変更が optimizing状態になると有効になり、通常は数秒かかります。
- 完全に初期化されていないボリュームでは、変更時間が長くなります。詳細については、「作成後にボリュームを手動で初期化する」を参照してください。
- IOPS またはスループットパフォーマンスを指定gp2gp3せずにボリュームタイプを から に変更すると、Amazon EBS は自動的にパフォーマンスをプロビジョニングします。ソースgp2ボリュームと同等のパフォーマンスまたはベースラインgp3パフォーマンスのいずれか高い方を使用します。

  例えば、IOPS またはスループットパフォーマンスを指定せずに、250 MiB/秒のスループットと 1,500 IOPS の 500 GiB gp2 ボリュームを gp3 に変更すると、Amazon EBS は 3,000 IOPS (ベースライン gp3 IOPS) と 250 MiB/秒 (ソース gp2 ボリュームスループットに一致するように) の gp3 ボリュームを自動的にプロビジョニングします。
- EBS ボリュームを変更する際にエラーメッセージが表示された場合や前世代のインスタンスタイプにアタッチされた EBS ボリュームを変更する場合は、以下のいずれかのステップを行ってください。
  - ルート以外のボリュームの場合は、ボリュームをインスタンスからデタッチして、変更を適用した後で、ボリュームを再アタッチします。
  - ルートボリュームの場合は、インスタンスを停止し、変更を適用した後で、インスタンスを再起動します。

制限事項

- ボリューム変更リクエストの送信後は、キャンセルできません。
- ボリュームサイズを増やす必要があります。ボリュームサイズを小さくすることはできません。ただし、より小さなボリュームを作成し、そのボリュームに対して rsync (Linux インスタンス) または robocopy (Windows インスタンス) などのアプリケーションレベルのツールを使用してデータを移行することができます。
- ボリュームの変更でリクエストできる集計ストレージの最大数には制限があります。詳細については、「Amazon Web Services 全般のリファレンス 」の「Amazon EBS service quotas」を参照してください。
- 変更後のボリュームサイズは、そのファイルシステムとパーティション設定スキームでサポートされる容量を超えることはできません。詳細については、「Amazon EBS ボリュームの制約」を参照してください。

- ボリュームタイプを変更しない場合は、ボリュームサイズとパフォーマンスを、現在のボリュームタイプの制限内であれば変更できます。ボリュームタイプを変更する場合は、ターゲットとなるボリュームタイプの制限内であれば、ボリュームサイズとパフォーマンスを変更することが可能です。詳細については、Amazon EBS ボリュームの種類を参照してください。
- Nitro ベースのインスタンスは、最大 256,000 IOPS でプロビジョニングされたボリュームをサポートします。他のインスタンスタイプは、最大 64,000 IOPS までプロビジョニングされたボリュームにアタッチできますが、最大 32,000 IOPS までプロビジョニングできます。
- マルチアタッチが有効なボリュームの場合、ボリュームがアタッチされている間を含め、ボリュームタイプを双方向io2で io1から の間で変更できます。その他のボリュームタイプの変更はサポートされていません。

  マルチアタッチが有効なボリュームio2のボリュームタイプを io1から に変更すると、ボリュームの NVMe 予約の有効化状態はターゲットボリュームタイプに引き継がれます。
- タイプ io1、io2、gp2、gp3、または standard のルート ボリュームは、インスタンスからデタッチされていても、st1 または sc1 ボリュームに変更できません。
- ボリュームが 2016 年 11 月 3 日 23:40 (UTC) 以前にアタッチされていた場合は、Elastic Volumes サポートを初期化する必要があります。詳細については、Elastic Volumes サポートの初期化を参照してください。
- m3.medium インスタンスはボリュームの変更を完全にサポートしていますが、m3.large、m3.xlarge、および m3.2xlarge インスタンスは、すべてのボリューム変更機能をサポートしていない場合があります。

Amazon EBS ボリューム変更の要件

Amazon EBS ボリュームを変更すると、以下の要件と制限が適用されます。EBS ボリュームの一般的な要件についての詳細は、Amazon EBS ボリュームの制約を参照してください。

トピック

- サポートされるインスタンスタイプ
- オペレーティングシステム

サポートされるインスタンスタイプ

Elastic Volumes は、次のインスタンスでサポートされています。

- すべての現行世代のインスタンス

- 旧世代のインスタンス: C1、C3、C4、G2、I2、M1、M3、M4、R3 および R4

インスタンスタイプが Elastic Volumes をサポートしていない場合は、Elastic Volumes がサポートされていない場合の EBS ボリュームの変更を参照してください。

オペレーティングシステム

次のオペレーティングシステム要件が適用されます。

Linux

Linux AMI では、2 TiB (2048 GiB) 以上のブートボリュームについて GUID パーティションテーブル (GPT) と GRUB 2 が必要です。現在の多くの Linux AMI は依然として MBR パーティションスキームを使用しており、2 TiB までのブートボリュームのみをサポートしています。インスタンスが 2 TiB を超えるブートボリュームで起動しない場合、使用中の AMI は、2 TiB のブートボリュームサイズに制限されている可能性があります。ブートボリューム以外のボリュームには、Linux インスタンスでこの制限はありません。

2 TiB より大きな値にブートボリュームのサイズを変更する前に、ボリュームが MBR と GPT のどちらのパーティション分割を使用しているのか確認します。それには、インスタンス上で、コマンドを実行します。

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

GPT パーティション分割を使用している Amazon Linux インスタンスでは、次の情報が返ります。

```
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.
```

MBR パーティション分割を使用している SUSE インスタンスは、次の情報を返します。

```
GPT fdisk (gdisk) version 0.8.8

  Partition table scan:
```

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

Windows

デフォルトでは、Windows はマスターブートレコード (MBR) パーティションテーブルを使用してボリュームを初期化します。MBR は 2 TiB (2048 GiB) 以下のボリュームのみをサポートするため、Windows はこのサイズ制限以上の MBR ボリュームのサイズ変更を阻止します。この場合、[Extend Volume (ボリュームの拡張)] のオプションは Windows の [Disk Management (ディスクの管理)] ユーティリティで無効になっています。 AWS マネジメントコンソール または を使用してサイズ制限を超える MBR パーティションボリューム AWS CLI を作成する場合、Windows は追加のスペースを検出または使用することはできません。

この制限を回避するには、GUID パーティションテーブル (GPT) でより大きな新しいボリュームを作成し、元の MBR ボリュームからデータをコピーします。

GPT ボリュームを作成するには

1. EC2 インスタンスのアベイラビリティーゾーンに必要な容量の新しい空のボリュームを作成し、インスタンスにアタッチします。

   > **Note**
   >
   > 新しいボリュームには、スナップショットから復元したボリュームは使用できません。

2. Windows システムにログインし、[Disk Management (ディスク管理)] (diskmgmt.exe) を開きます。
3. 新しいディスクのコンテキスト (右クリック) メニューを開き、[Online] を選択します。
4. [Initialize Disk] ウィンドウで、新規のディスクを選択し、続いて [GPT (GUID Partition Table)]、[OK] の順に選択します。
5. 初期化が完了したら、robocopy または teracopy などのツールを使用して元のボリュームから新しいボリュームにデータをコピーします。
6. [Disk Management] で、ドライブ文字を適切な値に変更し、古いボリュームをオフラインにします。
7. Amazon EC2 コンソールで、インスタンスから古いボリュームをデタッチ後、インスタンスを再起動して正常に稼働することを確認したら、古いボリュームを削除します。

Amazon EBS ボリュームの変更をリクエスト

Amazon EBS の伸縮自在なボリュームでは、そのサイズを増やしたり、パフォーマンスを増減したり、ボリュームタイプを変更したりなどが、ボリュームをデタッチすることなく動的に行えます。

プロセスの概要

1. (オプション) 重要なデータを含むボリュームを変更する前に、変更をロールバックする必要がある場合に備えて、ボリュームのスナップショットを作成するのがベストプラクティスです。詳細については、Amazon EBS スナップショットの作成を参照してください。
2. ボリュームの変更をリクエストします。
3. ボリューム変更の進行状況をモニタリングします。詳細については、Amazon EBS ボリューム変更の進行状況のモニタリングを参照してください。
4. ボリュームのサイズが変更された場合、増加されたストレージ容量を利用するには、ボリュームのファイルシステムを拡張します。詳細については、「Amazon EBS ボリュームのサイズ変更後にファイルシステムを拡張」を参照してください。

コンテンツ

- Elastic Volumes を使用して EBS ボリュームを変更する
- Elastic Volumes がサポートされていない場合の EBS ボリュームの変更
- Elastic Volumes サポートの初期化 (必要な場合)

Elastic Volumes を使用して EBS ボリュームを変更する

開始する前に、以下を参照してください。

- 考慮事項
- 制限事項
- 要件

Console

EBS ボリュームを変更するには

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. ナビゲーションペインの [ボリューム] を選択します。

3. ボリュームを選択し、[Actions] (アクション)、[Modify volume] (ボリュームの編集) の順にクリックします。
4. [Modify Volume] (ボリュームの編集) 画面に、ボリューム ID とボリュームの現在の設定 (タイプ、サイズ、IOPS、スループットなど) が表示されます。新しい設定値を以下のように設定します。
   - タイプを変更するには、[Volume Type] (ボリュームタイプ) の値を選択します。
   - サイズを変更するには、[Size] に新しい値を入力します。
   - (gp3,io1, および io2のみ) IOPS を変更するには、IOPS に新しい値を入力します。
   - (gp3 のみ) スループットを変更するには、[Throughput] (スループット) に新しい値を入力します。
5. ボリューム設定を変更したら、[変更] を選択します。確認を求めるメッセージが表示されたら、[Modify] (変更) を選択します。
6. ボリュームのサイズを大きくした場合は、追加のストレージ容量を利用するために、ボリュームのパーティションも拡張する必要があります。詳細については、「Amazon EBS ボリュームのサイズ変更後にファイルシステムを拡張」を参照してください。
7. (Windows インスタンスのみ) NVMe ドライバーがないインスタンスで AWS NVMe ボリュームのサイズを増やす場合は、インスタンスを再起動して Windows が新しいボリュームサイズを表示できるようにする必要があります。 AWS NVMe ドライバーのインストールの詳細については、AWS NVMe ドライバー」を参照してください。

AWS CLI

EBS ボリュームを変更するには

modify-volume コマンドを使用します。例えば、サイズが 100 GiB で、タイプが gp2 のボリュームがある場合は、次の例でこの設定を 10,000 IOPS、サイズが 200 GiB のタイプ io1 のボリュームに変更します。

```
aws ec2 modify-volume \
    --volume-id vol-01234567890abcdef \
    --volume-type io1 \
    --iops 10000 \
    --size 200
```

以下は出力の例です。

```
{
    "VolumeModification": {
        "TargetSize": 200,
        "TargetVolumeType": "io1",
        "ModificationState": "modifying",
        "VolumeId": "vol-01234567890abcdef",
        "TargetIops": 10000,
        "StartTime": "2022-01-19T22:21:02.959Z",
        "Progress": 0,
        "OriginalVolumeType": "gp2",
        "OriginalIops": 300,
        "OriginalSize": 100
    }
}
```

ボリュームのサイズを大きくした場合は、追加のストレージ容量を利用するために、ボリュームのパーティションも拡張する必要があります。詳細については、「Amazon EBS ボリュームのサイズ変更後にファイルシステムを拡張」を参照してください。

PowerShell

EBS ボリュームを変更するには

Edit-EC2Volume コマンドレットを使用します。例えば、サイズが 100 GiB で、タイプが gp2 のボリュームがある場合は、次の例でこの設定を 10,000 IOPS、サイズが 200 GiB のタイプ io1 のボリュームに変更します。

```
Edit-EC2Volume `
    -VolumeId vol-01234567890abcdef `
    -VolumeType io1 `
    -Iops 10000 `
    -Size 200
```

ボリュームのサイズを大きくした場合は、追加のストレージ容量を利用するために、ボリュームのパーティションも拡張する必要があります。詳細については、「Amazon EBS ボリュームのサイズ変更後にファイルシステムを拡張」を参照してください。

Elastic Volumes がサポートされていない場合の EBS ボリュームの変更

サポートされているインスタンスタイプを使用している場合は、Elastic Volumes を使用して、Amazon EBS ボリュームのサイズ、パフォーマンス、およびボリュームのタイプを動的に変更することができます。それらをデタッチする必要はありません。

Elastic Volumes は使用できないが、ルート (ブート) ボリュームの変更が必要になった場合は、インスタンスを停止し、ボリュームを変更してから、インスタンスを再起動する必要があります。

インスタンスが起動したら、ファイルシステムのサイズを確認して、拡大したボリュームスペースをインスタンスが認識しているかどうか表示できます。Linux では、df -h コマンドを使用してファイルシステムのサイズを確認します。

```
[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
```

新しく拡張したボリュームがサイズに反映されていない場合は、デバイスのファイルシステムを拡張して、インスタンスで新しいスペースを使えるようにします。詳細については、「Amazon EBS ボリュームのサイズ変更後にファイルシステムを拡張」を参照してください。

Windows インスタンスを使用すると、ボリュームを使用するためにオンラインにする必要がある場合があります。詳細については、「Amazon EBS ボリュームを使用できるようにする」を参照してください。ボリュームを再フォーマットする必要はありません。

Elastic Volumes サポートの初期化 (必要な場合)

2016 年 11 月 3 日 23:40 (UTC) 以前にインスタンスにアタッチされたボリュームを変更する前に、次のいずれかのアクションを使用してボリュームのサポートを初期化する必要があります。

- ボリュームをデタッチしてアタッチする
- インスタンスの停止と起動

Console

インスタンスの準備が完了していることを確認するには

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. ナビゲーションペインで、[インスタンス] を選択します。

3. [列の表示/非表示] アイコン (歯車) を選択します。[Launch time (起動時刻)] 属性列を選択し、[Confirm (確認)] を選択します。
4. [起動時刻] 列でインスタンスの一覧をソートします。カットオフ日より前に開始されたインスタンスごとに、[Storage (ストレージ)] タブを選択し、[Attachment time (アタッチ時刻)] 列をチェックして、ボリュームがいつアタッチされたかを確認します。

AWS CLI

インスタンスの準備が完了していることを確認するには

ボリュームが 2016 年 11 月 3 日 23:40 (UTC) 以前にアタッチされたかどうかを確認するには、次の describe-instances コマンドを使用します。

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

各インスタンスの出力の最初の行は、その ID と、カットオフ日前に開始されたかどうか (True または False) を示します。その最初の行の後に、各 EBS ボリュームがカットオフ日前にアタッチされたかどうかを示す 1 つ以上の行が続きます。次の出力例では、最初のインスタンスのボリューム変更を初期化する必要があります。これはカットオフ日よりも前に開始され、カットオフ日より前にそのルートボリュームがアタッチされていたためです。他のインスタンスはカットオフ日以降に開始されたため、準備は完了しています。

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

PowerShell

インスタンスの準備が完了していることを確認するには

Get-EC2Instance コマンドレットを使用して、ボリュームが 2016 年 11 月 3 日 23:40 UTC より前にアタッチされたかどうかを確認します。

```
(Get-EC2Instance `
    -InstanceId i-1234567890abcdef0).Instances.BlockDeviceMappings | `
     Format-Table @{Name="VolumeId";Expression={$_.Ebs.VolumeId}}, `
                  @{Name="AttachTime";Expression={$_.Ebs.AttachTime}}
```

以下は出力の例です。

```
VolumeId              AttachTime
--------              ----------
vol-0b243c8d927752d2b 3/23/2020 12:21:14 AM
vol-043eadbeb4a8387c3 9/5/2020 7:39:22 PM
vol-0c3f0c4e55c082753 4/23/2019 4:07:40 PM
```

Amazon EBS ボリューム変更の進行状況のモニタリング

EBS ボリュームを変更すると、次のステータスになります。ボリュームの状態は modifying、optimizing、completed の順に変わります。この時点で、ボリュームは追加の変更を適用できる状態になります。

ボリュームが optimizing 状態である場合、ボリュームのパフォーマンスはソースとターゲットの設定仕様の中間にあります。過渡的なボリュームのパフォーマンスは、ソースボリュームのパフォーマンスより劣ることはありません。IOPS をダウングレードする場合、過渡的なボリュームのパフォーマンスは、ターゲットボリュームと同程度のパフォーマンスになります。

ボリュームの変更による影響は次のとおりです。

- サイズの増加は、ボリュームの変更が optimizing状態になると有効になり、通常は数秒かかります。
- リクエストされたボリューム設定によっては、パフォーマンス (IOPS とスループット) の変更が完了するまでに数分から数時間かかる場合があります。通常、完全に使用された 1-TiB ボリュームは、新しいパフォーマンス設定に移行するまでに約 6 時間かかることがあります。場合によっては、ボリュームが完全に初期化されていない場合など、新しいパフォーマンス設定が有効になるまでに 24 時間以上かかることがあります。

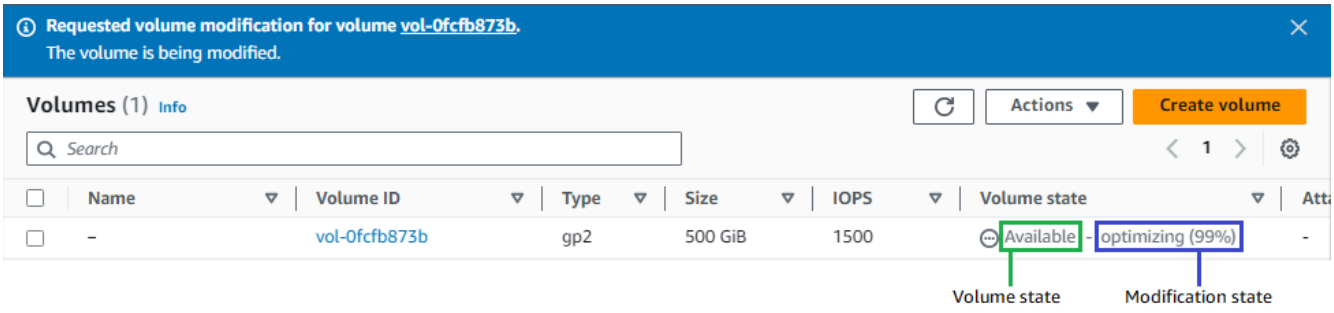
ボリュームの状態には、creating、available、in-use、deleting、deleted、error があります。

修正の状態には、modifying、optimizing、completed があります。

Console

変更の進行状況をモニタリングするには

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. ナビゲーションペインの [ボリューム] を選択します。
3. ボリュームを選択します。
4. [詳細] タブの [ボリュームのステータス] 列と [ボリュームステータス] フィールドには、*volume-state* - *modification-state* — *Modification state* という形式の情報が含まれます。次の画像に、ボリュームとボリュームの変更状態を示します。



変更が完了すると、ボリュームの状態のみが表示されます。変更の状態と進行状況は表示されなくなります。

または、Amazon EventBridge を使用して、ボリューム変更イベントの通知ルールを作成できます。詳細については、「Getting started with Amazon EventBridge」を参照してください。

AWS CLI

変更の進行状況をモニタリングするには

1 つ以上のボリュームの変更の進行状況を表示するには、describe-volumes-modifications コマンドを使用します。次の例では、2 つのボリュームのボリューム変更を示します。

```
aws ec2 describe-volumes-modifications \
```

```
    --volume-ids vol-1111111111111111 vol-2222222222222222
```

次の出力例では、これらのボリュームの変更の状態は、引き続き modifying になっています。進捗状況は、パーセンテージで報告されます。

```
{
    "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-2222222222222222",
            "StartTime": "2017-01-19T22:23:22.158Z",
            "Progress": 0,
            "OriginalVolumeType": "gp2",
            "OriginalIops": 300,
            "OriginalSize": 1000
        }
    ]
}
```

次の例では、変更の状態が optimizing または completed であるすべてのボリュームを示し、その結果をフィルタリングおよびフォーマットして 2017 年 2 月 1 日以降に開始された変更のみを表示します。

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

2 つのボリュームに関する情報を含む出力例を以下に示します。

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

PowerShell

変更の進行状況をモニタリングするには

Get-EC2VolumeModification コマンドレットを使用します。次の例では、2 つのボリュームのボリューム変更を示します。

```
Get-EC2VolumeModification `
    -VolumeId vol-1111111111111111 vol-2222222222222222
```

> **Note**
>
> まれに、一時的な AWS 障害によって failed状態が発生することがあります。これは、ボリュームのヘルスステータスを示すものではなく、ボリュームの変更に失敗したことを単に示しています。この場合は、再度ボリュームの変更を行います。

Amazon EBS ボリュームのサイズ変更後にファイルシステムを拡張

EBS ボリュームのサイズを増やしたら、パーティションおよびファイルシステムを新しいより大きなサイズに拡張する必要があります。ボリュームが optimizing 状態に入るとすぐにこれを実行できます。

[開始する前に]

- 変更をロールバックする必要がある場合に備えて、ボリュームのスナップショットを作成します。詳細については、「Amazon EBS スナップショットの作成」を参照してください。

- ボリュームの変更が成功し、`optimizing` または `completed` 状態になっていることを確認します。詳細については、「Amazon EBS ボリューム変更の進行状況のモニタリング」を参照してください。
- ボリュームがインスタンスにアタッチされ、フォーマットおよびマウントされていることを確認します。詳細については、「アタッチ済みボリュームのフォーマットとマウント」を参照してください。
- (Linux インスタンスのみ) Amazon EBS ボリュームで論理ボリュームを使用している場合、Logical Volume Manager (LVM) を使用して論理ボリュームを拡張する必要があります。これを行う方法については、「LVM を使用して EBS ボリュームのパーティションに論理ボリュームを作成する方法を教えてください」の記事の「LV を拡張する」のセクションを参照してください。

Linux インスタンス

> **Note**
>
> 次の手順では、Linux の XFS および Ext4 ファイルシステムを拡張するプロセスについて説明します。別のファイルシステムの拡張に関する詳細ついては、そのドキュメントを参照してください。

ボリュームにパーティションがある場合、Linux でファイルシステムを拡張する前にパーティションを拡張する必要があります。

EBS ボリュームのファイルシステムを拡張する

サイズを変更したボリュームのファイルシステムを拡張するには、以下の手順を使用します。

Xen インスタンスおよび Nitro System 上に構築されたインスタンスでは、デバイスおよびパーティションの命名が異なることに注意してください。インスタンスが Xen ベースか Nitro ベースかを確認するには、「ハイパーバイザーのタイプ」を参照してください。

EBS ボリュームのファイルシステムを拡張するには

1. インスタンスに接続します。
2. 必要に応じて、パーティションのサイズを変更します。そのためには、次の操作を行います。

    a. ボリュームにパーティションがあるかどうかを確認します。lsblk コマンドを使用します。

Nitro instance example

次の出力例では、ルートボリューム (nvme0n1) には 2 つのパーティション (nvme0n1p1 および nvme0n1p128) がありますが、追加のボリューム (nvme1n1) にはパーティションがありません。

```
[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

次の出力例では、ルートボリューム (xvda) にはパーティション (xvda1) がありますが、追加のボリューム (xvdf) にはパーティションがありません。

```
[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
```

- ボリュームにパーティションがある場合は、次のステップ (2b) に進みます。
- ボリュームにパーティションがない場合は、ステップ 2b、2c、2d をスキップして、ステップ 3 に進みます。

> **トラブルシューティングのヒント**
>
> コマンド出力にボリュームが表示されない場合は、ボリュームがインスタンスにアタッチされ、フォーマットおよびマウントされていることを確認します。

b. パーティションを拡張する必要があるかどうかを確認します。前のステップの lsblk コマンド出力で、パーティションサイズとボリュームサイズを比較します。

- パーティションサイズがボリュームサイズよりも小さい場合は、次のステップ (2c) に進みます。
- パーティションサイズがボリュームサイズと等しい場合、パーティションを拡張する必要はありません。ステップ 2c および 2d をスキップし、ステップ 3 に進みます。

> **トラブルシューティングのヒント**
>
> ボリュームがまだ元のサイズを反映している場合は、ボリュームの変更が成功したことを確認します。

c. パーティションを拡張します。growpart コマンドを使用して、デバイス名とパーティション番号を指定します。

Nitro instance example

パーティション番号は、p の後の番号です。例えば、nvme0n1p1 の場合、パーティション番号は 1 です。nvme0n1p128 の場合、パーティション番号は 128 です。

例えば、nvme0n1p1 という名前のパーティションを拡張するには、次のコマンドを使用します。

> **Important**
>
> デバイス名 (nvme0n1) とパーティション番号 (1) の間のスペースに注意してください。

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

Xen instance example

パーティション番号は、デバイス名の後の番号です。例えば、xvda1 の場合、パーティション番号は 1 です。xvda128 の場合、パーティション番号は 128 です。

例えば、xvda1 という名前のパーティションを拡張するには、次のコマンドを使用します。

> **Important**
>
> デバイス名 (xvda) とパーティション番号 (1) の間のスペースに注意してください。

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

> **トラブルシューティングのヒント**
>
> - `mkdir: cannot create directory ‘/tmp/growpart.31171’: No space left on device FAILED: failed to make temp dir`: サイズ変更の実行に必要な一時ディレクトリを growpart が作成するのに十分な空きディスク容量がボリュームにないことを示します。ディスク容量を少し解放してから、もう一度お試しください。
> - `must supply partition-number`: 正しくないパーティションが指定されたことを示します。lsblk コマンドを使用してパーティション名を確認し、デバイス名とパーティション番号の間にスペースを入力してください。
> - `NOCHANGE: partition 1 is size 16773087. it cannot be grown`: パーティションが既にボリューム全体を拡張しており、拡張できないことを示します。ボリュームの変更が成功したことを確認します。

d. パーティションが拡張されたことを確認します。lsblk コマンドを使用します。これで、パーティションのサイズはボリュームサイズと同じになります。

Nitro instance example

次の出力例は、両方のボリューム (nvme0n1) とパーティション (nvme0n1p1) が同じサイズ (16 GB) であることを示しています。

```
[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

次の出力例は、両方のボリューム (xvda) とパーティション (xvda1) が同じサイズ (16 GB) であることを示しています。

```
[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. ファイルシステムを拡張します。

   a. 拡張する必要があるファイルシステムの名前、サイズ、タイプ、およびマウントポイントを取得します。df -hT または lsblk -f コマンドを使用します。

   Nitro instance example

   df -hT コマンドの次の出力例では、/dev/nvme0n1p1 ファイルシステムのサイズが 8 GB、タイプが xfs、マウントポイントが / であることを示しています。

```
[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

   df -hT コマンドの次の出力例では、/dev/xvda1 ファイルシステムのサイズが 8 GB、タイプが ext4、マウントポイントが / であることを示しています。

```
[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
...
```

- ファイルシステムのサイズがボリュームサイズよりも小さい場合は、次のステップ (3b) に進みます。
- ファイルシステムのサイズがボリュームサイズと等しい場合、拡張する必要はありません。この場合、残りのステップをスキップします。パーティションとファイルシステムは新しいボリュームサイズに拡張されています。

b. ファイルシステムを拡張するコマンドは、ファイルシステムのタイプによって異なります。前のステップで書き留めたファイルシステムのタイプに基づいて、次の正しいコマンドを選択します。

- [XFS ファイルシステム] xfs_growfs コマンドを使用して、前のステップで書き留めたファイルシステムのマウントポイントを指定します。

  Nitro and Xen instance example

  例えば、/ にマウントされているファイルシステムを拡張するには、次のコマンドを使用します。

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

  トラブルシューティングのヒント

  - `xfs_growfs: /data is not a mounted XFS filesystem`: 正しくないマウントポイントが指定されたか、ファイルシステムが XFS でないことを示します。マウントポイントとファイルシステムタイプを確認するには、df -hT コマンドを使用します。
  - `data size unchanged, skipping`: ファイルシステムが既にボリューム全体を拡張していることを示します。ボリュームにパーティションがない場合は、ボリュームの変更が成功したことを確認します。ボリュームにパーティションがある場合は、ステップ 2 の説明に従ってパーティションが拡張されていることを確認してください。

- [Ext4 ファイルシステム] resize2fs コマンドを使用して、前のステップで書き留めたファイルシステムの名前を指定します。

Nitro instance example

例えば、マウントされた /dev/nvme0n1p1 という名前のファイルシステムを拡張するには、次のコマンドを使用します。

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

Xen instance example

例えば、マウントされた /dev/xvda1 という名前のファイルシステムを拡張するには、次のコマンドを使用します。

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

トラブルシューティングのヒント

- `resize2fs: Bad magic number in super-block while trying to open /dev/xvda1`: ファイルシステムが Ext4 ではないことを示します。ファイルのシステムタイプを確認するには、df -hT コマンドを使用します。
- `open: No such file or directory while opening /dev/xvdb1`: 正しくないパーティションが指定されたことを示します。パーティションを検証するには、df -hT コマンドを使用します。
- `The filesystem is already 3932160 blocks long. Nothing to do!`: ファイルシステムが既にボリューム全体を拡張していることを示します。ボリュームにパーティションがない場合は、ボリュームの変更が成功したことを確認します。ボリュームにパーティションがある場合は、ステップ 2 で説明されているように、パーティションが拡張されていることを確認してください。

- [その他のファイルシステム] 手順については、ファイルシステムのドキュメントを参照してください。

c. ファイルシステムが拡張されたことを確認します。df -hT コマンドを使用して、ファイルシステムのサイズがボリュームサイズと等しいことを確認します。

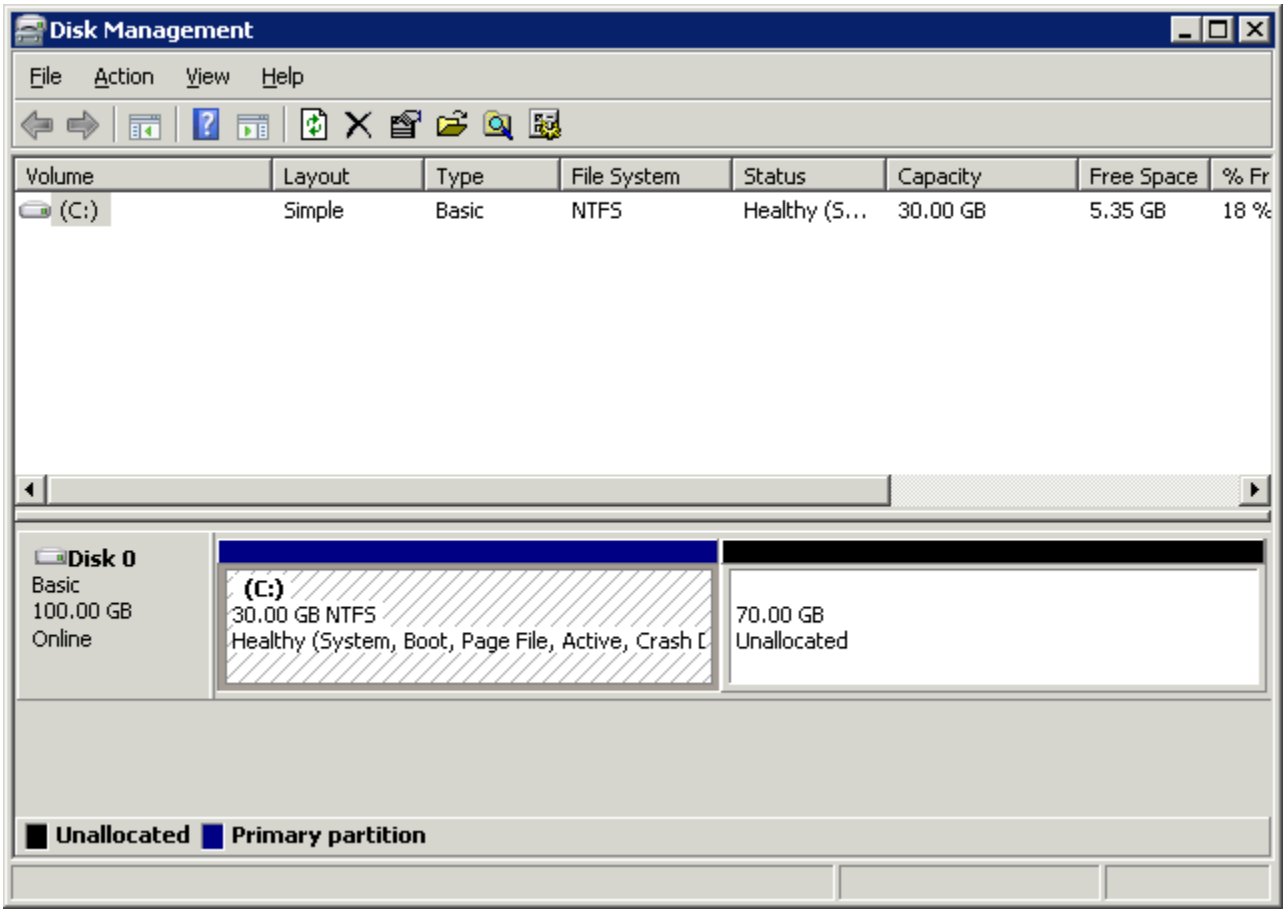
Windows インスタンス

次のいずれかの方法を使用して Windows インスタンスでファイルシステムを拡張します。

Disk Management utility

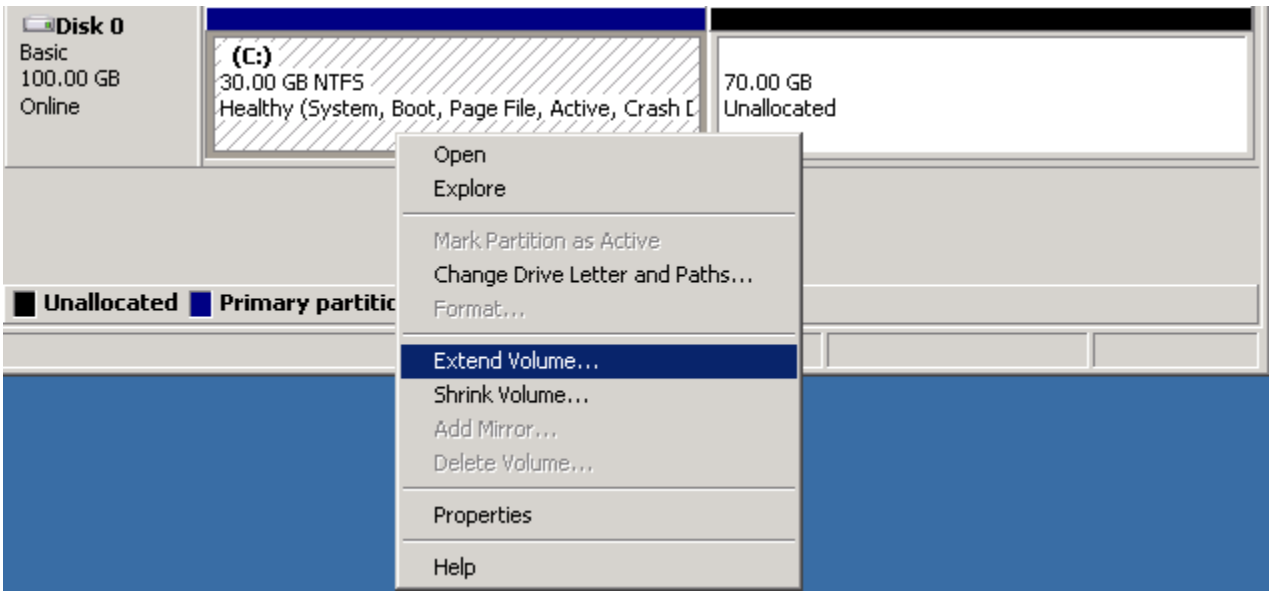
ディスクの管理を使用してファイルシステムを拡張するには

1. 重要なデータを含むファイルシステムを拡張する前に、変更をロールバックする必要がある場合に備えて、ファイルシステムを含むボリュームのスナップショットを作成するのがベストプラクティスです。詳細については、Amazon EBS スナップショットの作成を参照してください。
2. リモートデスクトップを使用して Windows インスタンスにログインします。
3. [実行] ダイアログボックスにdiskmgmt.mscと入力し、Enter キーを押します。ディスクの管理ユーティリティが表示されます。

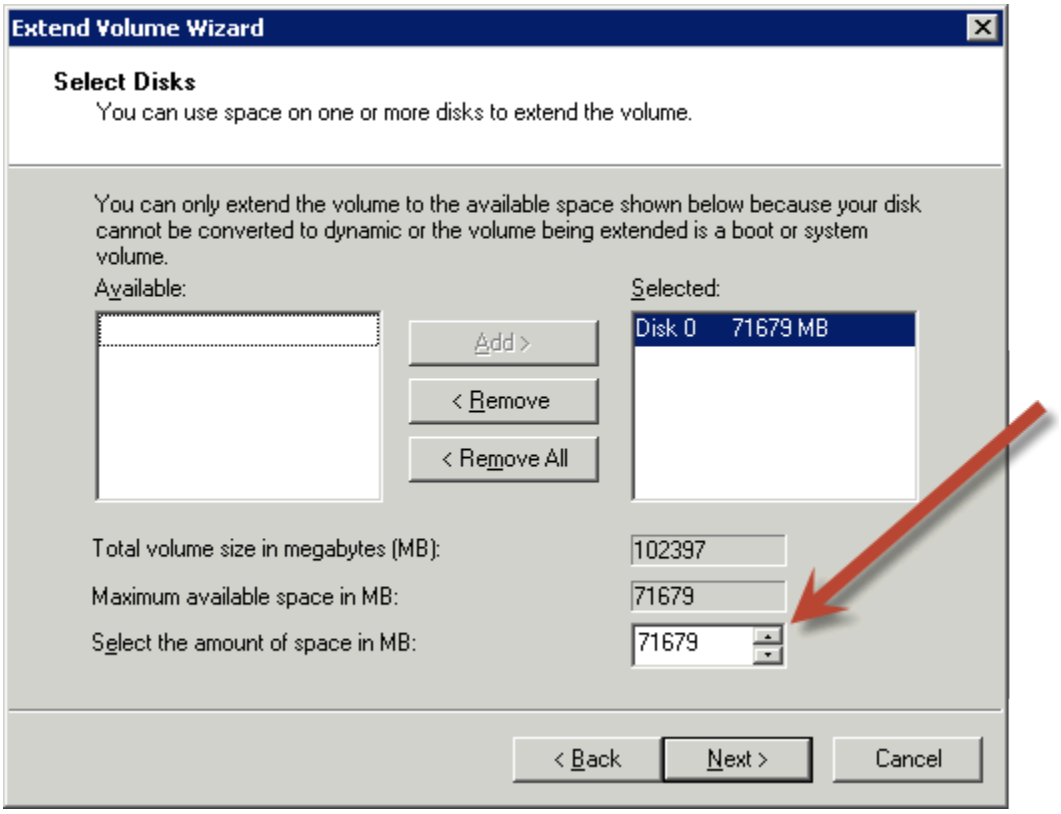


4. [ディスクの管理] メニューで、[操作]、[ディスクの再スキャン] の順に選択します。
5. 拡張したドライブのコンテキスト (右クリック) メニューを開き、[Extend Volume (ボリュームの拡張)] を選択します。

> **Note**
>
> 次の場合には、[ボリュームを拡張] が無効化されている (グレー表示される) ことがあります。
>
> - 未割り当ての領域が、ドライブに隣接していない。未割り当て領域は、拡張するドライブの右側に隣接している必要があります。
> - ボリュームが、マスターブートレコード (MBR) パーティションスタイルを使用しており、そのサイズが既に 2 TB に達している。MBR を使用するボリュームのサイズは 2 TB を超えることはできません。



6. [Extend Volume (ボリュームの拡張)] ウィザードで、[Next (次へ)] を選択します。[Select the amount of space in MB] で、ボリュームを拡張するメガバイト数を入力します。通常は、使用可能な最大領域を指定します。[Selected] のハイライト表示されたテキストは、ボリュームの最終的なサイズではなく、追加分の容量を表します。ウィザードを終了します。



7. AWS NVMe ドライバーがないインスタンスで NVMe ボリュームのサイズを増やす場合、インスタンスを再起動して Windows を有効にし、新規のボリュームサイズを表示する必要があります。 AWS NVMe ドライバーのインストールの詳細については、AWS NVMe ドライバー」を参照してください。

PowerShell

PowerShell を使用して Windows ファイルシステムを拡張するには、以下の手順に従います。

PowerShell を使用してファイルシステムを拡張するには

1. 重要なデータを含むファイルシステムを拡張する前に、変更をロールバックする必要がある場合に備えて、ファイルシステムを含むボリュームのスナップショットを作成するのがベストプラクティスです。詳細については、Amazon EBS スナップショットの作成を参照してください。
2. リモートデスクトップを使用して Windows インスタンスにログインします。
3. 管理者として PowerShell を実行します。

4. Get-Partition コマンドを実行します。PowerShell は、各パーティションに対応するパーティション番号、ドライブ文字、オフセット、サイズ、タイプを返します。拡張するパーティションのドライブ文字をメモします。
5. 以下のコマンドを実行して、ディスクを再スキャンします。

```
"rescan" | diskpart
```

6. **<drive-letter>** の代わりに手順 4 でメモしたドライブ文字を使用して、以下のコマンドを実行します。PowerShell から、許可されているパーティションの最小サイズと最大サイズがバイト単位で返されます。

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

7. 指定された規模にパーティションを拡張するには、**<size>** の代わりにボリュームの新しいサイズを入力しながら、次のコマンドを実行します。サイズは、KB、MB、GB で入力できます (50GB など)。

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

使用可能な最大サイズにパーティションを拡張するには、次のコマンドを実行します。

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

特定のサイズにファイルシステムを拡張するための、完全な PowerShell コマンドと、そのレスポンスフローを以下に示します。

```
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
```

使用可能な最大サイズにファイルシステムを拡張するための、完全な PowerShell コマンドと、そのレスポンスフローを以下に示します。

```
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&34763423&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&34763423&0&000000#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}

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

Amazon EC2 インスタンスから Amazon EBS ボリュームをデタッチ

Amazon Elastic Block Store (Amazon EBS) ボリュームをインスタンスからデタッチしてから、別のインスタンスにアタッチまたは削除する必要があります。ボリュームをデタッチしても、ボリュームのデータには影響しません。

トピック

- 考慮事項
- ボリュームのアンマウントとデタッチ

- トラブルシューティング

考慮事項

- インスタンスから Amazon EBS ボリュームをデタッチするには、明示的にデタッチするか、インスタンスを終了します。ただし、インスタンスが実行中の場合、最初にインスタンスからボリュームをアンマウントする必要があります。
- EBS ボリュームがインスタンスのルートデバイスである場合、ボリュームをデタッチする前に、インスタンスを停止する必要があります。
- (アンマウントせずに) 切り離したボリュームを再接続することはできますが、同じマウントポイントが得られない可能性があります。進行中のボリュームがデタッチされたときにそのボリュームへの書き込みがあった場合、ボリューム上のデータは同期していない可能性があります。
- ボリュームをデタッチした後も、ストレージ量が AWS 無料利用枠の制限を超えている限り、ボリュームストレージに対して課金されます。不要な料金の発生を防ぐために、ボリュームを削除する必要があります。詳細については、Amazon EBS ボリュームの削除を参照してください。

ボリュームのアンマウントとデタッチ

ボリュームをインスタンスからアンマウントおよびデタッチするには、次の手順を使用します。これは、ボリュームを別のインスタンスにアタッチする必要がある場合や、ボリュームを削除する必要がある場合に便利です。

ステップ

- 手順 1: ボリュームをアンマウントする
- 手順 2: ボリュームをインスタンスからデタッチする
- ステップ 3: (Windows インスタンスのみ) オフラインデバイスのロケーションをアンインストールする

手順 1: ボリュームをアンマウントする

Linux インスタンス

Linux インスタンスから、次のコマンドを使用して /dev/sdh デバイスのマウントを解除します。

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

Windows インスタンス

Windows インスタンスから、次のようにボリュームをマウント解除します。

1. [Disk Management] ユーティリティを起動します。
   - (Windows Server 2012 以降) タスクバーで、Windows ロゴを選択し (右クリック) 、ディスク管理を選択します。
   - (Windows Server 2008) [Start]、[Administrative Tools]、[Computer Management]、[Disk Management] の順に選択します。
2. ディスクを選択し (たとえば、ディスク 1 を選択 (右クリック) 、オフラインを選択します。ディスクのステータスが [オフライン] に変わるまで待ってから Amazon EC2 コンソールを開きます。

手順 2: ボリュームをインスタンスからデタッチする

インスタンスからボリュームをデタッチするには、次のいずれかの方法を使用します。

Console

EBS ボリュームをデタッチするには

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. ナビゲーションペインの [ボリューム] を選択します。
3. ボリュームを選択します。
4. [アクション]、[ボリュームのデタッチ] を選択します。
5. 確認を求められたら、[デタッチ] を選択してください。

AWS CLI

EBS ボリュームをインスタンスからデタッチするには

ボリュームをアンマウントしたら、detach-volume コマンドを使用します。

```
aws ec2 detach-volume --volume-id vol-01234567890abcdef
```

PowerShell

EBS ボリュームをインスタンスからデタッチするには

ボリュームをアンマウントしたら、Dismount-EC2Volume コマンドレットを使用します。

```
Dismount-EC2Volume -VolumeId vol-01234567890abcdef
```

ステップ 3: (Windows インスタンスのみ) オフラインデバイスの口ケーションをアンインストールする

ボリュームをインスタンスからアンマウントおよびデタッチすると、Windows はデバイスの場所にオフラインのフラグを立てます。インスタンスの再起動と停止および再起動後、デバイスの場所はオフラインのままになります。インスタンスを再起動すると、Windows はオフラインのデバイスの場所に残りのボリュームの 1 つをマウントする可能性があります。これにより、ボリュームが Windows で使用できなくなります。これを回避し、次回 Windows を起動するときにすべてのボリュームがオンラインデバイス口ケーションにアタッチされていることを確認するには、次の手順を実行します。

1. インスタンスで、デバイスマネージャーを開きます。
2. デバイスマネージャーで、[View]、[Show hidden devices] の順に選択します。
3. デバイスのリストで、[Storage controllers] ノードを展開します。

   デタッチされたボリュームがマウントされたデバイスの場所に `AWS NVMe Elastic Block Storage Adapter` という名前が付けられ、グレー表示されます。
4. という名前のグレー表示された各デバイスの場所を選択 (右クリック) し`AWS NVMe Elastic Block Storage Adapter`、デバイスをアンインストール を選択し、アンインストール を選択します。

   > **Important**
   >
   > [Delete the driver software for this device] チェックボックスはオンにしないでください。

トラブルシューティング

ボリュームをデタッチする場合に発生する一般的な問題と、それらを解決する方法は、次のとおりです。

> **Note**
>
> データ損失の可能性に対する保護を許可するには、ボリュームのスナップショットを作成してからアンマウントを試みます。スタックしたボリュームの強制デタッチを行うと、インスタンスを再起動しない限り、ファイルシステムまたはファイルシステムに含まれるデータに損害を与えたり、同じデバイス名を使用して新しいボリュームをアタッチできなくなったりする可能性があります。

- Amazon EC2 コンソールからボリュームを切り離しているときに問題が発生した場合は、describe-volumes CLI コマンドを使用して問題を診断すると便利なことがあります。詳細については、describe-volumesを参照してください。
- ボリュームの状態がdetaching状態のまま変わらない場合は、[Force Detach] を選択して、強制的にアタッチ解除することもできます。障害が発生したインスタンスからボリュームをアタッチ解除するための最後の手段として、またはボリュームを削除するためにデタッチする場合のみ、このオプションを使用してください。インスタンスは、ファイルシステムキャッシュやファイルシステムメタデータをフラッシュする機会を失います。このオプションを使用する場合は、ファイルシステムのチェックと修復の手順を手動で実行する必要があります。
- ボリュームを数分間何度も強制的に切断しようとしたが、detaching 状態のままになっている場合は、AWS re:Post へのヘルプのリクエストを送信できます。迅速に解決できるようにするため、ボリューム ID と、これまでに実行した手順を記述してください。
- まだマウントされたボリュームをデタッチしようとすると、ボリュームはデタッチを実行しようとして busy 状態でスタックする可能性があります。describe-volumes からの次の出力は、この状態の例を示しています。

```
"Volumes": [
    {
        "AvailabilityZone": "us-west-2b",
        "Attachments": [
            {
                "AttachTime": "2022-07-21T23:44:52.000Z",
                "InstanceId": "i-1234567890abcdef0",
                "VolumeId": "vol-01234567890abcdef",
                "State": "busy",
                "DeleteOnTermination": false,
                "Device": "/dev/sdf"
            }
        ...
```

```
    }
]
```

この状態が発生した場合、ボリュームのアンマウント、デタッチの強制、インスタンスの再起動、またはそれら 3 つをすべて行うまで、デタッチは無期限に遅れる可能性があります。

Amazon EBS ボリュームの削除

Amazon EBS ボリュームが不要になったら、それを削除することができます。削除後、ボリュームに含まれるデータは消去され、ボリューム自体はどのインスタンスにもアタッチできなくなります。削除前にボリュームのスナップショットを保存できるので、それを使用すれば後でボリュームを再作成できます。

インスタンスにアタッチされているボリュームは削除できません。ボリュームを削除するには、まずボリュームをデタッチする必要があります。詳細については、「Amazon EC2 インスタンスから Amazon EBS ボリュームをデタッチ」を参照してください。

ごみ箱の保持ルールに一致するボリュームを削除すると、そのボリュームはすぐに削除されるのではなく、ごみ箱に保持されます。詳細については、「ごみ箱」を参照してください。

Console

EBS ボリュームを削除するには

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. ナビゲーションペインの [ボリューム] を選択します。
3. ボリュームを選択します。ボリュームが [使用可能] 状態であることを確認します。
4. [アクション]、[ボリュームの削除] を選択します。

   このオプションが無効になっている場合、ボリュームはインスタンスにアタッチされ、削除できません。
5. 確認を求められたら、「**delete**」と入力してから、[削除] を選択します。

AWS CLI

EBS ボリュームが使用中かどうかを確認するには

describe-volumes コマンドを使用します。ボリュームが使用中の場合、状態は `in-use` になります。それ以外の場合は、`available` です。

```
aws ec2 describe-volumes \
    --volume-id vol-01234567890abcdef \
    --query Volumes[*].State \
    --output text
```

EBS ボリュームを削除するには

delete-volume コマンドを使用します。

```
aws ec2 delete-volume --volume-id vol-01234567890abcdef
```

PowerShell

EBS ボリュームが使用中かどうかを確認するには

Get-EC2Volume コマンドレットを使用します。ボリュームが使用中の場合、状態は in-use になります。それ以外の場合は、available です。

```
(Get-EC2Volume `
    -VolumeId vol-01234567890abcdef).State.Value
```

EBS ボリュームを削除するには

Remove-EC2Volume コマンドレットを使用します。

```
Remove-EC2Volume -VolumeId vol-01234567890abcdef
```

スナップショットを使用した Amazon EBS ボリュームの置き換え

Amazon EBS スナップショットは、速度、利便性、コストに優れるため、Amazon EC2 で推奨されるバックアップツールです。スナップショットからボリュームを作成すると、すべてのデータをそのままの状態で、過去の特定時点の状態が再作成されます。スナップショットから作成されたボリュームをインスタンスにアタッチすることで、リージョン間でのデータの複製、テスト環境の作成、損傷または破損した本稼働ボリュームの完全な置換、特定のファイルとディレクトリの取得とアタッチされた別のボリュームへの転送を行うことができます。詳細については、「Amazon EBS スナップショット」を参照してください。

Amazon EBS ボリュームを、そのボリュームの以前のスナップショットから作成された別のボリュームに置き換えるには、次の手順のいずれかを使用できます。

要件

インスタンスと同じアベイラビリティーゾーンに新しい EBS ボリュームを作成する必要があります。ボリュームは、インスタンスと同じアベイラビリティーゾーンにアタッチする必要があります。

Console

ボリュームを置き換えるには

1. スナップショットからボリュームを作成し、新しいボリュームの ID を書き留めます。詳細については、「Amazon EBS ボリュームの作成」を参照してください。
2. [インスタンス] ページで、ボリュームを置き換えるインスタンスを選択し、インスタンス ID を書き留めます。

   インスタンスが選択された状態で、[Storage] (ストレージ) タブを選択します。[Block devices] (ブロックデバイス) セクションで、置き換えるボリュームを検索し、ボリュームのデバイス名を書き留めます (例: `/dev/sda1`)。
3. [ストレージ] タブで、ボリューム ID を選択し、インスタンスからボリュームをアンマウントおよびデタッチします。
4. ステップ 1 で作成した新しいボリュームを選択し、[Actions] (アクション)、[Attach volume] (ボリュームのアタッチ) を選択します。

   [Instance] (インスタンス) および [Device name] (デバイス名) に、ステップ 2 で書き留めたインスタンス ID とデバイス名を入力し、[Attach volume] (ボリュームのアタッチ) を選択します。
5. インスタンスに接続し、ボリュームをマウントします。詳細については、「Amazon EBS ボリュームを使用できるようにする」を参照してください。

AWS CLI

ボリュームを置き換えるには

1. スナップショットから新しいボリュームを作成します。create-volume コマンドを使用して、`--snapshot-id` オプションを指定します。`--availability-zone` には、インスタンスと同じアベイラビリティーゾーンを指定します。出力で、新しいボリュームの ID を書き留めます。

```
aws ec2 create-volume \
```

```
    --volume-type gp3 \
    --snapshot-id snap-0abcdef1234567890 \
    --availability-zone us-east-1a
```

2. 置き換えるボリュームのデバイス名を取得します。describe-instances コマンドを使用します。--instance-ids には、ボリュームを置き換えるインスタンスの ID を指定します。置き換えるボリュームのデバイス名とボリューム ID を書き留めます。

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0 \
    --query Reservations[].Instances[].BlockDeviceMappings
```

3. 交換するボリュームをインスタンスからデタッチします。detach-volume コマンドを使用します。

```
aws ec2 detach-volume --volume-id vol-xxxxxxxxxxxxxxxxx
```

4. 置換ボリュームをインスタンスにアタッチします。attach-volume コマンドを使用します。--volume-id には、置き換えるボリュームの ID を指定します。--instance-id には、ボリュームをアタッチするインスタンスの ID を指定します。--device には、先ほどメモしたものと同じデバイス名を指定します。

```
aws ec2 attach-volume \
    --volume-id vol-01234567890abcdef \
    --instance-id i-1234567890abcdef0 \
    --device /dev/sdf
```

5. インスタンスに接続し、ボリュームをマウントします。詳細については、「Amazon EBS ボリュームを使用できるようにする」を参照してください。

PowerShell

ボリュームを置き換えるには

1. スナップショットから新しいボリュームを作成します。-SnapshotId オプションで New-EC2Volume コマンドレットを使用します。-AvailabilityZone には、インスタンスと同じアベイラビリティーゾーンを指定します。出力で、新しいボリュームの ID を書き留めます。

```
New-EC2Volume `
```

```
    -VolumeType gp3 `
    -SnapshotId snap-0abcdef1234567890 `
    -AvailabilityZone us-east-1a
```

2. 置き換えるボリュームのデバイス名を取得します。Get-EC2Instance コマンドレットを使用します。-InstanceId には、ボリュームを置き換えるインスタンスの ID を指定します。置き換えるボリュームのデバイス名とボリューム ID を書き留めます。

```
(Get-EC2Instance `
    -InstanceId i-1234567890abcdef0).Instances.BlockDeviceMappings | `
     Format-Table DeviceName, @{Name="VolumeId";Expression={$_.Ebs.VolumeId}}
```

3. 交換するボリュームをインスタンスからデタッチします。Dismount-EC2Volume コマンドレットを使用します。

```
DismountEC2Volume -VolumeId vol-xxxxxxxxxxxxxxxxx
```

4. 置換ボリュームをインスタンスにアタッチします。Add-EC2Volume コマンドレットを使用します。-VolumeId には、置き換えるボリュームの ID を指定します。-InstanceId には、ボリュームをアタッチするインスタンスの ID を指定します。-Device には、先ほどメモしたものと同じデバイス名を指定します。

```
Add-EC2Volume`
    -VolumeId vol-01234567890abcdef `
    -InstanceId i-1234567890abcdef0 `
    -Device /dev/sdf
```

5. インスタンスに接続し、ボリュームをマウントします。詳細については、「Amazon EBS ボリュームを使用できるようにする」を参照してください。

Amazon EBS ボリュームステータスチェック

ボリュームステータスチェックを利用すると、Amazon EBS ボリュームのデータの潜在的な不整合を容易に理解、追跡、および管理できます。これらのチェックは、Amazon EBS ボリュームに障害が発生しているかどうかを判断するために必要な情報を提供し、潜在的に不整合なボリュームの処理方法を制御できるように設計されています。

ボリュームステータスチェックは 5 分ごとに自動的に試行され、成功または失敗のステータスを返します。すべてのチェックが成功した場合、ボリュームのステータスは ok です。チェックが失敗した場合、ボリュームのステータスは impaired です。ステータスが insufficient-data の場

合、ボリュームのチェックがまだ実行中である可能性があります。ボリュームステータスチェックの結果を表示して、障害のあるボリュームを特定し、必要なアクションを行うことができます。

ボリュームのデータが潜在的に不整合であると Amazon EBS が判断した場合、デフォルトでは、アタッチされたすべての EC2 インスタンスからそのボリュームへの I/O が無効になります。これにより、データの破損を防ぐことができます。I/O が無効になると、次のボリュームステータスチェックが失敗し、ボリュームステータスは `impaired` になります。さらに、I/O が無効になったこと、およびボリュームへの I/O を有効にすることによってボリュームの障害ステータスを解決できることを伝えるイベントが表示されます。EBS は、I/O を有効にして、インスタンスにボリュームを引き続き使用させるか、 fsck (Linux インスタンス) や chkdsk (Windows インスタンス) などのコマンドを使用して整合性チェックを実行するかを決定する機会が得られるまで待機します。

> **Note**
>
> ボリュームステータスはボリュームステータスチェックに基づいており、ボリューム状態を反映していません。従って、ボリュームステータスではボリュームが `error` 状態 (例えば、I/O を受け付けできない) であることは判りません。ボリュームの状態についての詳細は、ボリューム状態を参照してください。

あるボリュームの整合性について心配しているわけではなく、そのボリュームに障害が発生した際にそのボリュームをすぐに利用できるようにしたい場合は、デフォルトの動作を上書きして、I/O を自動的に有効にするようにボリュームを設定することができます。Auto-Enable IO 属性 (API の `autoEnableIO`) を有効にしている場合は、ボリューム状態のチェックは引き続きパスされます。また、ボリュームに潜在的な障害があると判断されたが、そのボリュームの I/O が自動的に有効になったことを伝えるイベントも表示されます。これにより、ボリュームの整合性を確認したり、後でボリュームを交換したりすることが可能になります。

I/O パフォーマンスステータスチェックは、実際のボリュームパフォーマンスと予想されるボリュームパフォーマンスを比較します。ボリュームパフォーマンスが想定未満である場合は、警告が表示されます。このステータスチェックは、インスタンスにアタッチされている、プロビジョンド IOPS SSD (`io1` および `io2`) ボリュームと、汎用 SSD (gp3) ボリュームに対してのみ使用できます。ステータスチェックは、汎用 SSD (gp2)、スループット最適化 HDD (`st1`)、Cold HDD (`sc1`)、または磁気 (`standard`) ボリュームでは使用できません。I/O パフォーマンスステータスチェックは 1 分ごとに 1 回実行され、CloudWatch は 5 分ごとにこのデータを収集します。`io1` または `io2` ボリュームをインスタンスにアタッチしてから、ステータスチェックが I/O パフォーマンスステータスを報告するまでに、最長で 5 分かかる場合があります。

> **Important**
>
> スナップショットから復元された Provisioned IOPS SSD ボリュームを初期化している間は、ボリュームのパフォーマンスが想定レベルの 50% を下回る場合があります。このため、ボリュームの [I/O Performance (I/O パフォーマンス)] ステータスチェックでは `warning` 状態が表示されます。これは想定の動作です。初期化中の Provisioned IOPS SSD ボリュームの `warning` 状態は無視してかまいません。詳細については、作成後にボリュームを手動で初期化するを参照してください。

次の表に、Amazon EBS ボリュームのステータスを示します。

ボリュームのステータス	I/O 有効ステータス	I/O パフォーマンスステータス (**io1**、**io2**、および **gp3** ボリュームのみ)
ok	Enabled (I/O Enabled または I/O Auto-Enabled)	Normal (ボリュームパフォーマンスは想定どおり)
`warning`	Enabled (I/O Enabled または I/O Auto-Enabled)	Degraded (ボリュームのパフォーマンスが想定を下回っている) Severely Degraded (ボリュームのパフォーマンスが想定をかなり下回っている)
`impaired`	Enabled (I/O Enabled または I/O Auto-Enabled) 無効 (ボリュームがオフラインで復旧が保留中であるか、I/O を有効にするのを待っています)	Stalled (ボリュームのパフォーマンスは致命的な影響を受けている) Not Available (I/O が無効なため、I/O パフォーマンスの判定不能)
`insufficient-data`	Enabled (I/O Enabled または I/O Auto-Enabled)	Insufficient Data

ボリュームのステータス	I/O 有効ステータス	I/O パフォーマンスステータス (**io1**、**io2**、および **gp3** ボリュームのみ)
	Insufficient Data	

Console

ステータスチェックを表示するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインの [ボリューム] を選択します。

   [Volume Status] (ボリュームのステータス) 列に、各ボリュームの動作状況が表示されます。
3. 特定のボリュームのステータスの詳細を表示するには、グリッドからボリュームを選択して、[Status Checks]] (ステータスチェック) タブを選択します。
4. ステータスチェックが失敗したボリュームがある場合 (ステータスが `impaired` と示されている場合) は、障害のある Amazon EBS ボリュームを操作する を参照してください。

ナビゲータで [Events (イベント)] を選択して、インスタンスとボリュームのすべてのイベントを表示することもできます。詳細については、Amazon EBS ボリュームイベントを参照してください。

AWS CLI

ボリュームステータス情報を表示するには

describe-volume-status コマンドを使用します。

```
aws ec2 describe-volume-status --volume-ids vol-01234567890abcdef
```

次の例を使用して、障害のあるボリュームを特定します。

```
aws ec2 describe-volume-status --filters Name=volume-status.status,Values=impaired
```

PowerShell

ボリュームステータス情報を表示するには

Get-EC2VolumeStatus コマンドレットを使用します。

```
Get-EC2VolumeStatus -VolumeId vol-01234567890abcdef
```

次の例を使用して、障害のあるボリュームを特定します。

```
Get-EC2VolumeStatus -Filter @{Name="volume-status.status"; Values="impaired"}
```

Amazon EBS ボリュームイベント

ボリュームのデータが潜在的に不整合であると Amazon EBS によって判断された場合、デフォルトでは、アタッチされているすべての EC2 インスタンスからそのボリュームへの I/O が無効になります。これにより、ボリュームステータスチェックが失敗し、障害の原因を示すボリュームステータスイベントが作成されます。

データが潜在的に不整合であるボリュームで I/O を自動的に有効にするには、Auto-Enabled IO ボリューム属性 (API の autoEnableIO) の設定を変更します。この属性の変更の詳細については、障害のある Amazon EBS ボリュームを操作するを参照してください。

各イベントには、イベントが発生した時刻を示す開始時刻と、そのボリュームに対する I/O が無効になった時間を示す継続時間が含まれています。ボリュームに対する I/O が有効になると、イベントに終了時刻が追加されます。

ボリュームステータスイベント

`Awaiting Action: Enable IO`

ボリュームデータに整合性がない可能性があります。ボリュームに対する I/O は、ユーザーが明示的に有効にするまで無効になります。I/O を明示的に有効にすると、イベントの説明が IO Enabled に変更されます。

`IO Enabled`

このボリュームに対する I/O 操作が明示的に有効にされました。

`IO Auto-Enabled`

イベントの発生後に、このボリュームで I/O 操作が自動的に有効になりました。データを引き続き使用する前に、データの整合性を確認することをお勧めします。

`Normal`

io1、io2、および gp3 ボリュームのみ。ボリュームのパフォーマンスは想定どおりです。

`Degraded`

io1、io2、および gp3 ボリュームのみ。ボリュームのパフォーマンスは想定を下回っています。

`Severely Degraded`

io1、io2、および gp3 ボリュームのみ。ボリュームのパフォーマンスは想定をはるかに下回っています。

`Stalled`

io1、io2、および gp3 ボリュームのみ。ボリュームのパフォーマンスは致命的な影響を受けています。

I/O が無効になっているボリュームがある場合は、障害のある Amazon EBS ボリュームを操作するを参照してください。I/O パフォーマンスが通常を下回るボリュームがある場合、実行したアクション (ピーク時の使用時にボリュームのスナップショットを作成する、必要な I/O 帯域幅をサポートできないインスタンスでボリュームを実行する、ボリュームのデータに初めてアクセスするなど) が原因で一時的な状態になる可能性があります。

Console

ボリュームイベントを表示するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインの [Events] を選択します。イベントを含むすべてのインスタンスおよびボリュームがリストされています。
3. ボリュームでフィルタリングして、ボリュームステータスのみを表示できます。特定のタイプのステータスでフィルタリングすることもできます。
4. ボリュームを選択して、その特定のイベントを表示します。

AWS CLI

ボリュームイベントを表示するには

describe-volume-status コマンドを使用します。

```
aws ec2 describe-volume-status --volume-ids vol-01234567890abcdef
```

PowerShell

ボリュームイベントを表示するには

Get-EC2VolumeStatus コマンドレットを使用します。

```
Get-EC2VolumeStatus -VolumeId vol-01234567890abcdef
```

障害のある Amazon EBS ボリュームを操作する

ボリュームのデータが整合していない可能性があるためにボリュームに障害がある場合は、以下のオプションを使用します。

Options

- オプション 1: インスタンスにアタッチされたボリュームで整合性チェックを実行する
- オプション 2: 別のインスタンスを使用してボリュームで整合性チェックを実行する
- オプション 3: 不要なボリュームを削除する

オプション 1: インスタンスにアタッチされたボリュームで整合性チェックを実行する

もっとも単純なオプションは、ボリュームが Amazon EC2 にアタッチされているときに、I/O を有効にしてから、ボリュームでデータの整合性チェックを実行するオプションです。

アタッチされたボリュームで整合性チェックを実行するには

1. アプリケーションによるボリュームの使用を停止します。
2. ボリュームの I/O を有効にします。次のいずれかの方法を使用します。

   Console

   ボリュームの I/O を有効にするには

   1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
   2. ナビゲーションペインの [Events] を選択してください。
   3. ボリュームを選択します。

4. [Actions] (アクション)、[Enable I/O] (I/Oを有効化) を選択します。

AWS CLI

ボリュームの I/O を有効にするには

enable-volume-io コマンドを使用します。

```
aws ec2 enable-volume-io --volume-id vol-01234567890abcdef
```

PowerShell

ボリュームの I/O を有効にするには

Enable-EC2VolumeIO コマンドレットを使用します。

```
Enable-EC2VolumeIO -VolumeId vol-01234567890abcdef
```

3. ボリュームのデータを確認します。

a. fsck (Linux インスタンス) または chkdsk (Windows インスタンス) コマンドを実行します。

b. (オプション) 関連するエラーメッセージがないか、使用可能なアプリケーションログまたはシステムログを確認します。

c. ボリュームに 20 分以上障害が発生した場合は、 AWS サポートセンターにお問い合わせください。[Troubleshoot (トラブルシューティング)] をクリックしてから、[Troubleshoot Status Checks (ステータスチェックのトラブルシューティング)] ダイアログボックスの [Contact Support (サポートに問い合わせる)] を選択してサポートケースを送信します。

オプション 2: 別のインスタンスを使用してボリュームで整合性チェックを実行する

実動環境外部のボリュームをチェックするには、次の手順に従います。

> **Important**
>
> この手順を実行すると、ボリューム I/O を無効にしたときに停止された書き込み I/O が失われる場合があります。

分離されたボリュームで整合性チェックを実行するには

1. アプリケーションによるボリュームの使用を停止します。
2. ボリュームをインスタンスからデタッチします。詳細については、Amazon EC2 インスタンスから Amazon EBS ボリュームをデタッチを参照してください。
3. ボリュームの I/O を有効にします。次のいずれかの方法を使用します。

   Console

   ボリュームの I/O を有効にするには

   1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
   2. ナビゲーションペインの [Events] を選択してください。
   3. 前の手順でデタッチしたボリュームを選択します。
   4. [Actions] (アクション)、[Enable I/O] (I/Oを有効化) を選択します。

   AWS CLI

   ボリュームの I/O を有効にするには

   enable-volume-io コマンドを使用します。

   ```
   aws ec2 enable-volume-io --volume-id vol-01234567890abcdef
   ```

   PowerShell

   ボリュームの I/O を有効にするには

   Enable-EC2VolumeIO コマンドレットを使用します。

   ```
   Enable-EC2VolumeIO -VolumeId vol-01234567890abcdef
   ```

4. ボリュームを別のインスタンスにアタッチします。詳細については、「インスタンスの起動」および「Amazon EBS ボリュームを Amazon EC2 インスタンスにアタッチ」を参照してください。
5. ボリュームのデータを確認します。

   a. fsck (Linux インスタンス) または chkdsk (Windows インスタンス) コマンドを実行します。

b. (オプション) 関連するエラーメッセージがないか、使用可能なアプリケーションログまたはシステムログを確認します。
c. ボリュームに 20 分以上障害が発生した場合は、 AWS サポートセンターにお問い合わせください。[Troubleshoot (トラブルシューティング)] を選択し、トラブルシューティングのダイアログボックスで [Contact Support (サポートに問い合わせる)] を選択して、サポートケースを送信します。

オプション 3: 不要なボリュームを削除する

環境からボリュームを削除するには、単にそれを削除します。ボリュームの削除の詳細については、Amazon EBS ボリュームの削除を参照してください。

ボリュームのデータをバックアップするスナップショットを最近作成した場合、そのスナップショットから新しいボリュームを作成できます。詳細については、「Amazon EBS ボリュームの作成」を参照してください。

障害のある Amazon EBS ボリュームの I/O の自動有効化

ボリュームのデータが潜在的に不整合であると Amazon EBS によって判断された場合、デフォルトでは、アタッチされているすべての EC2 インスタンスからそのボリュームへの I/O が無効になります。これにより、ボリュームステータスチェックが失敗し、障害の原因を示すボリュームステータスイベントが作成されます。あるボリュームの整合性について心配しているわけではなく、そのボリュームに障害が発生した際にそのボリュームをすぐに利用できるようにしたい場合は、デフォルトの動作を上書きして、I/O を自動的に有効にするようにボリュームを設定することができます。Auto-Enable IO 属性 (API の `autoEnableIO`) を有効にしている場合は、ボリュームとインスタンスとの間の I/O が自動的に有効になり、ボリュームのステータスチェックはパスされます。また、ボリュームが潜在的に不整合な状態であること、ただしそのボリュームの I/O が自動的に有効になったことを伝えるイベントも表示されます。このイベントが発生した場合は、ボリュームの整合性をチェックし、必要に応じて置き換えます。詳細については、「Amazon EBS ボリュームイベント」を参照してください。

Console

ボリュームの、 IO の自動有効化 属性を表示するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインの [ボリューム] を選択します。
3. ボリュームを選択して、[Status Checks] (ステータスチェック) を選択します。

[Auto-Enabled I/O] (自動有効化された I/O) には、ボリュームの現在の設定 ([Enabled] (有効) または [Disabled] (無効)) が表示されます。

ボリュームの、IO の自動有効化属性を変更するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインの [ボリューム] を選択します。
3. ボリュームを選択し、[Actions] (アクション)、[Manage auto-enabled I/O] (自動有効化 I/O の管理) を選択します。
4. 障害のあるボリュームの I/O を自動的に有効にするには、[Auto-Enable Volume I/O for impaired volumes] (障害のあるボリューム I/O の自動有効化) チェックボックスをオンにします。この機能を無効にするには、チェックボックスをクリアします。
5. [更新] を選択します。

AWS CLI

ボリュームの autoEnableIO 属性を表示するには

describe-volume-attribute コマンドを使用します。

```
aws ec2 describe-volume-attribute \
    --attribute autoEnableIO \
    --volume-id vol-01234567890abcdef
```

以下は出力の例です。

```
{
    "AutoEnableIO": {
        "Value": true
    },
    "VolumeId": "vol-01234567890abcdef"
}
```

ボリュームの autoEnableIO 属性を変更するには

modify-volume-attribute コマンドを使用します。

```
aws ec2 modify-volume-attribute \
```

```
    --auto-enable-io \
    --volume-id vol-01234567890abcdef
```

PowerShell

ボリュームの autoEnableIO 属性を表示するには

Get-EC2VolumeAttribute コマンドレットを使用します。

```
(Get-EC2VolumeAttribute `
    -Attribute autoEnableIO `
    -VolumeId vol-01234567890abcdef).AutoEnableIO
```

以下は出力の例です。

```
True
```

ボリュームの autoEnableIO 属性を変更するには

Edit-EC2VolumeAttribute コマンドレットを使用します。

```
Edit-EC2VolumeAttribute `
    -AutoEnableIO $true `
    -VolumeId vol-01234567890abcdef
```

Amazon EBS での障害テスト

AWS Fault Injection Service (AWS FIS) は、 AWS ワークロードでフォールトインジェクション実験を実行するのに役立つフルマネージドサービスです。EBS アクションを使用すると AWS FIS、アプリケーションが I/O の中断やボリュームのパフォーマンスの低下につながるストレージ障害にどのように応答するかをテストできます。この制御されたテスト環境により、アプリケーションが中断にどのように応答するかを観察できるため、アーキテクチャの弱点を特定し、アプリケーションの全体的な耐障害性を向上させることができます。I/O 一時停止アクションとレイテンシーインジェクションアクションを使用して、Amazon CloudWatch アラームとフェイルオーバーワークフローなどのモニタリングおよび復旧メカニズムをテストし、ミッションクリティカルなアプリケーションのストレージ障害に対する耐障害性を向上させることができます。詳細については AWS FIS、AWS Fault Injection Service 「 ユーザーガイド」を参照してください。

利用可能な実験

Amazon EBS は現在、2 つの AWS FIS フォールトインジェクションをサポートしています。

- I/O フォールトインジェクションを一時停止する
- レイテンシーインジェクション

考慮事項

以下の考慮事項に注意してください。

- すべての Amazon EBS ボリュームタイプがサポートされています。ルートボリュームとデータボリュームの両方がサポートされます。インスタンスストアボリュームはサポートされていません。
- ボリュームは Nitro ベースの EC2 インスタンスにアタッチする必要があります。
- 期間に基づいて実験が完了すると、ボリュームは元の I/O パフォーマンスを再開します。実行中の実験は、完了する前に停止することもできます。または、CloudWatch アラームで定義したしきい値に達した場合に実験を停止するように停止条件を作成することもできます。
- マルチアタッチが有効なボリューム AWS FIS では、 を使用できます。アタッチされたインスタンスのすべてが影響を受けます。実験用に特定のボリュームとインスタンスのアタッチメントを選択することはできません。
- FIS は現在、ローカルゾーン、Outposts、または Wavelength ゾーンでは使用できません。
- コンソールでボリューム ARN を指定するとき、同じアベイラビリティーゾーンで最大 5 つのボリュームを同時にテストできます。
- AWS FIS 、 ゾーンOutpost、またはローカル AWS Wavelength ゾーンで作成されたボリュームでを使用することはできません。

I/O フォールトインジェクションを一時停止する

AWS Fault Injection Service および I/O 一時停止アクションを使用して、Amazon EBS ボリュームとそれがアタッチされているインスタンス間の I/O を一時的に停止し、ワークロードが I/O 中断を処理する方法をテストします。

詳細については AWS FIS、AWS Fault Injection Service 「 ユーザーガイド」を参照してください。

考慮事項

ボリュームの I/O を一時停止する場合は、以下の考慮事項に留意してください。

- I/O の一時停止は、すべての Nitro ベースのインスタンスタイプでサポートされています。
- OS タイムアウト設定をテストするには、実験時間を nvme_core.io_timeout に指定された値以上に設定します。詳細については、「Amazon EBS ボリュームの NVMe I/O オペレーションタイムアウト」を参照してください。
- I/O が一時停止しているボリュームに I/O を実行すると、次のことが起こります。
  - ボリュームのステータスが 120 秒以内で impaired に遷移します。詳細については、「Amazon EBS ボリュームステータスチェック」を参照してください。
  - ボリューム I/O が 60 秒以上一時停止している場合、VolumeStalledIOCheck の CloudWatch メトリクスは 1 になります。詳細については、「Amazon EBS ボリュームのメトリクス」を参照してください。
  - キューの長さ (VolumeQueueLength) の CloudWatch メトリクスはゼロ以外になります。アラームやモニタリングでは、キューの深さがゼロでないかどうかを監視する必要があります。
  - VolumeReadOps または VolumeWriteOps の CloudWatch メトリクスは 0 になります。これは、ボリュームが I/O を処理していないことを示します。

Amazon EC2 コンソールから基本的な実験を実行することも、 AWS FIS コンソールを使用してより高度な実験を実行することもできます。 AWS FIS コンソールを使用して高度な実験を実行する方法の詳細については、「 AWS Fault Injection Service ユーザーガイド」の「 のチュートリアル AWS FIS」を参照してください。

Amazon EC2 コンソールを使用して基本的な実験を行うには

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. ナビゲーションペインの [ボリューム] を選択します。
3. I/O を一時停止するボリュームを選択し、[アクション]、[フォールトインジェクション]、[ボリューム I/O の一時停止] の順に選択します。
4. [所要時間] には、ボリュームとインスタンス間の I/O を一時停止する期間を入力します。[所要時間] ドロップダウンリストの横のフィールドには、ISO 8601 形式の期間が表示されます。
5. サービスアクセスセクションで、実験を実行する AWS FIS ために が引き受ける IAM サービスロールを選択します。デフォルトのロールか、作成した既存のロールを使用できます。詳細については、「AWS FIS 実験の IAM ロールを作成する」を参照してください。
6. [ボリュームの I/O を一時停止] を選択します。プロンプトが表示されたら、確認フィールドに start と入力し、[実験を開始] を選択します。

7. 実験の進行状況と影響をモニタリングします。詳細については、「AWS FIS ユーザーガイド」の「Monitoring AWS FIS」を参照してください。

レイテンシーインジェクション

のレイテンシーインジェクションアクション (aws:ebs:volume-io-latency) を使用して AWS FIS 、Amazon EBS ボリュームの I/O レイテンシーの増加をシミュレートし、アプリケーションがストレージパフォーマンスの低下にどのように応答するかをテストします。このアクションでは、挿入するレイテンシーの値と、ターゲットボリュームに影響を与える I/O の割合を指定できます。を使用すると AWS FIS、事前設定されたレイテンシー実験テンプレートを使用して、ストレージの障害中に観察される可能性のあるさまざまな I/O レイテンシーパターンのテストを開始できます。これらのテンプレートは、アプリケーションに中断を発生させて障害耐性をテストするために使用できるシナリオの初期セットとして設計されています。これらは、アプリケーションが実際に経験する可能性があるすべてのタイプの影響を網羅するように設計されているわけではありません。アプリケーションのパフォーマンスニーズに基づいて、複数の異なるテストを実行するように調整することをお勧めします。利用可能なテンプレートをカスタマイズするか、新しい実験テンプレートを作成して、アプリケーション固有の要件をテストすることができます。

事前設定済みのレイテンシー実験テンプレート

Amazon EBS は、EBS コンソールと AWS FIS シナリオライブラリを通じて次のレイテンシー実験テンプレートを提供します。ターゲットボリュームでこれらのテンプレートを直接使用して、レイテンシーインジェクション実験を実行することができます。

- 持続レイテンシー: 一定のレイテンシーをシミュレートします。この実験は、1 つのレイテンシーインジェクションアクションを使用し、合計所要時間は 15 分です。この実験では、読み取り I/O の 50% と書き込み I/O の 100% に対し、永続的なレイテンシーをシミュレートします (15 分間で 500 ミリ秒)。
- レイテンシーの増加: 徐々に増加するレイテンシーをシミュレートします。この実験は、5 つのレイテンシーインジェクションアクションを使用し、合計所要時間は 15 分です。この実験では、読み取り I/O の 10% と書き込み I/O の 25% に対し、レイテンシーの段階的な増加をシミュレートします (3 分間で 50 ミリ秒、3 分間で 200 ミリ秒、3 分間で 700 ミリ秒、3 分間で 1 秒、3 分間で 15 秒)。
- 断続的なレイテンシー: 断続的なレイテンシーの急激なスパイクと、その間の復旧期間をシミュレートします。この実験は、3 つのレイテンシーインジェクションアクションを使用し、合計所要時間は 15 分です。この実験では、読み取りおよび書き込み I/O の 0.1% に対し、3 つのレイテンシーのスパイクをシミュレートします (1 分間継続する 30 秒のスパイク、2 分間継続する 10 秒の

スパイク、2 分間継続する 20 秒のスパイク)。レイテンシーの各スパイク間に 5 分間の復旧期間があります。

- レイテンシーの削減: 徐々に減るレイテンシーをシミュレートします。この実験は、5 つのレイテンシーインジェクションアクションを使用し、合計所要時間は 15 分です。この実験では、読み取り I/O と書き込み I/O の 10% に対してレイテンシーの段階的な減少をシミュレートします (3 分間で 20 秒、3 分間で 5 秒、3 分間で 900 ミリ秒、3 分間で 300 ミリ秒、3 分間で 40 ミリ秒)。

事前設定済みのシナリオをカスタマイズする

上記の事前設定済みテンプレートをカスタマイズするか、次のカスタマイズ可能なパラメータを使用して独自の新しい実験テンプレートを作成します。

- `readIOPercentage` — レイテンシーを挿入する読み取り I/O オペレーションの割合。これは、アクションの影響を受けるボリュームに対するすべての読み取り I/O オペレーションの割合です。

  範囲: 最小 0.1%、最大 100%

- `readIOLatencyMilliseconds` — 読み取り I/O オペレーションに挿入されるレイテンシーの量。これは、実験中に読み取り I/O の指定された割合で観測されるレイテンシーの値です。

  範囲: 最小 1 ミリ秒 (io2) / 10 ミリ秒 (io2 以外)、最大 60 秒

- `writeIOPercentage` — レイテンシーを挿入する書き込み I/O オペレーションの割合。これは、アクションの影響を受けるボリュームに対するすべての書き込み I/O オペレーションの割合です。

  範囲: 最小 0.1%、最大 100%

- `writeIOLatencyMilliseconds` — 書き込み I/O オペレーションに挿入されるレイテンシーの量。これは、実験中に書き込み I/O の指定された割合で観測されるレイテンシーの値です。

  範囲: 最小 1 ミリ秒 (io2) / 10 ミリ秒 (io2 以外)、最大 60 秒

- `duration` — 選択した I/O の割合にレイテンシーが挿入される期間。

  範囲: 最小 1 秒、最大 12 時間

レイテンシーインジェクションをモニタリングする

ボリュームのパフォーマンスへの影響は、次の方法でモニタリングできます。

- CloudWatch で平均レイテンシーメトリクスを使用して、1 分あたりの平均 I/O レイテンシーを取得します。詳細については、「Monitor your EBS volumes using CloudWatch」を参照してください。
- NVMe-CLI、CloudWatch エージェント、および Prometheus で利用できる EBS の詳細なパフォーマンス統計を使用して、1 秒あたりの平均 I/O レイテンシーを取得します。詳細なメトリクスでは、ボリュームのレイテンシー分散を分析するために使用できる I/O レイテンシーヒストグラムも提供されます。詳細については、「NVMe detailed performance statistics」を参照してください。
- Amazon EBS ボリュームステータスチェック を使用します。I/O レイテンシーを挿入すると、ボリュームのステータスは `warning` 状態に移行します。

考慮事項

EBS レイテンシーインジェクションを使用する場合は、次の点を考慮します。

- レイテンシーインジェクションは、P4d、P5、P5e、Trn2u、G6、G6f、Gr6、Gr6f、M8i、M8i-flex、C8i-flex、R8i、R8i-flex、I8ge、Mac-m4pro、Mac-m4 を除くすべての Nitro ベースのインスタンスタイプでサポートされています。
- 実験で指定されたレイテンシー値と観測結果のレイテンシーには最大 5% の差異が生じることがあります。
- ごく少数の I/O オペレーションしか行わない場合、アクションパラメータで指定された I/O の割合が、アクションの影響を受ける I/O の実際の割合と一致しない可能性があります。

Amazon EBS ボリュームでレイテンシーインジェクション実験を実行するには

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. ナビゲーションペインの [ボリューム] を選択します。
3. 実験を実行するボリュームを選択し、[アクション]、[レジリエンステスト]、[挿入ボリューム I/O レイテンシー] を選択します。

   AWS Fault Injection Service コンソールが開きます。
4. [実験を作成] ウィンドウで、実行する実験のタイプを、[断続的]、[増加中]、[持続]、または [減少中] から選択します。
5. IAM ロールを選択するには、新しいロールの作成を選択して、ユーザーに代わって実験を実行するために AWS FIS が使用する新しいロールを作成します。または、必要なアクセス許可を持つ IAM ロールを以前に作成している場合は、[既存の IAM ロールを使用] を選択します。

6. [料金の見積り] セクションには、実験の実行コストの見積りが表示されます。では AWS FIS、実験のターゲットアカウントの数に基づいて、アクションが最初から最後まで実行される 1 分ごとに課金されます。
7. [実験を開始する] を選択します。

Amazon EBS スナップショット

Amazon EBS スナップショットと呼ばれるポイントインタイムコピーを作成することで、Amazon EBS ボリューム上のデータをバックアップできます。スナップショットは増分バックアップです。つまり、最新のスナップショットを作成した時点から変更があったボリューム上のブロックのみが保存されます。これにより、スナップショットを作成するのに要する時間が最小限に抑えられ、データを複製しないことで、ストレージコストが節約されます。

> **Important**
>
> AWS は、EBS ボリュームに保存されているデータを自動的にバックアップしません。データの回復力とディザスタリカバリのために、EBS スナップショットを定期的に作成するか、Amazon Data Lifecycle Manager でバックアップを自動化 または AWS Backup を使用して自動スナップショット作成を設定したりするのは、ユーザーの責任になります。

スナップショットは Amazon S3 の S3 バケット (直接アクセスできない) に保存されます。Amazon EC2 コンソールまたは Amazon EC2 API を使用して、スナップショットの作成と管理を行うことができます。Amazon S3 コンソールまたは Amazon S3 API を使用してスナップショットにアクセスすることはできません。

スナップショットデータは、リージョン内のすべてのアベイラビリティーゾーンで自動的にレプリケートされます。これにより、スナップショットデータの可用性と耐久性が向上し、そのリージョンの任意のアベイラビリティーゾーンでボリュームを復元することが可能になります。

各スナップショットには、(スナップショットを作成した瞬間から) データを新しい EBS ボリュームに復元するために必要な情報がすべて含まれます。スナップショットから EBS ボリュームを作成すると、新しいボリュームは、スナップショットの作成に使用されたボリュームの完全なレプリカとして開始されます。

詳細については、Amazon EBS スナップショットの製品ページを参照してください。

スナップショットイベント

EBS スナップショットの状態は、CloudWatch Events を通じて追跡できます。詳細については、「EBS スナップショットイベント」を参照してください。

スナップショットの料金

スナップショットの料金は、保存されているデータの量に基づきます。スナップショットは増分であるため、スナップショットを削除すると、データストレージのコストが削減されない場合があります。スナップショットによって排他的に参照されるデータは、そのスナップショットが削除されると削除されますが、他のスナップショットによって参照されるデータは保持されます。詳細については、「AWS Billing ユーザーガイド」の「Amazon Elastic Block Store のボリュームおよびスナップショット」を参照してください。

内容

- Amazon EBS スナップショットの仕組み
- Amazon EBS スナップショットのライフサイクル
- Amazon EBS 高速スナップショット復元
- Amazon EBS スナップショットのロック
- Amazon EBS スナップショットのブロックパブリックアクセス
- Amazon EBS local snapshots on Outposts
- ローカルゾーン内のローカルスナップショット

Amazon EBS スナップショットの仕組み

ボリュームから作成する最初のスナップショットは、常に完全なスナップショットです。これには、スナップショットの作成時にボリュームに書き込まれたすべてのデータブロックが含まれます。同じボリュームの後続スナップショットは、増分スナップショットです。それらには、最後のスナップショットが作成されてからボリュームに書き込まれた、変更されたデータブロックと新規のデータブロックのみが含まれます。

フルスナップショットのサイズは、ソースボリュームのサイズではなく、バックアップするデータのサイズによって決まります。同様に、フルスナップショットに関連するストレージコストは、ソースボリュームのサイズではなく、スナップショショットのサイズによって決まります。例えば、50 GiB データのみを含む 200 GiB Amazon EBS ボリュームの最初のスナップショットを作成します。これにより、フルスナップショットのサイズは 50 GiB となり、50 GiB スナップショットストレージの料金が請求されます。

同様に、増分スナップショットのサイズとストレージコストは、前回のスナップショットが作成された後にボリュームに書き込まれたデータのサイズによって決まります。前述の例で考えた場合、20 GiB のデータを変更して 10 GiB のデータを追加した後に同じ 200 GiB ボリュームの 2 つ目のスナップショットを作成すると、増分スナップショットのサイズは 30 GiB になります。その後、その追加の 30 GiB スナップショットストレージの料金が請求されます。

スナップショットの料金の詳細については、「Amazon EBS 料金表」を参照してください。

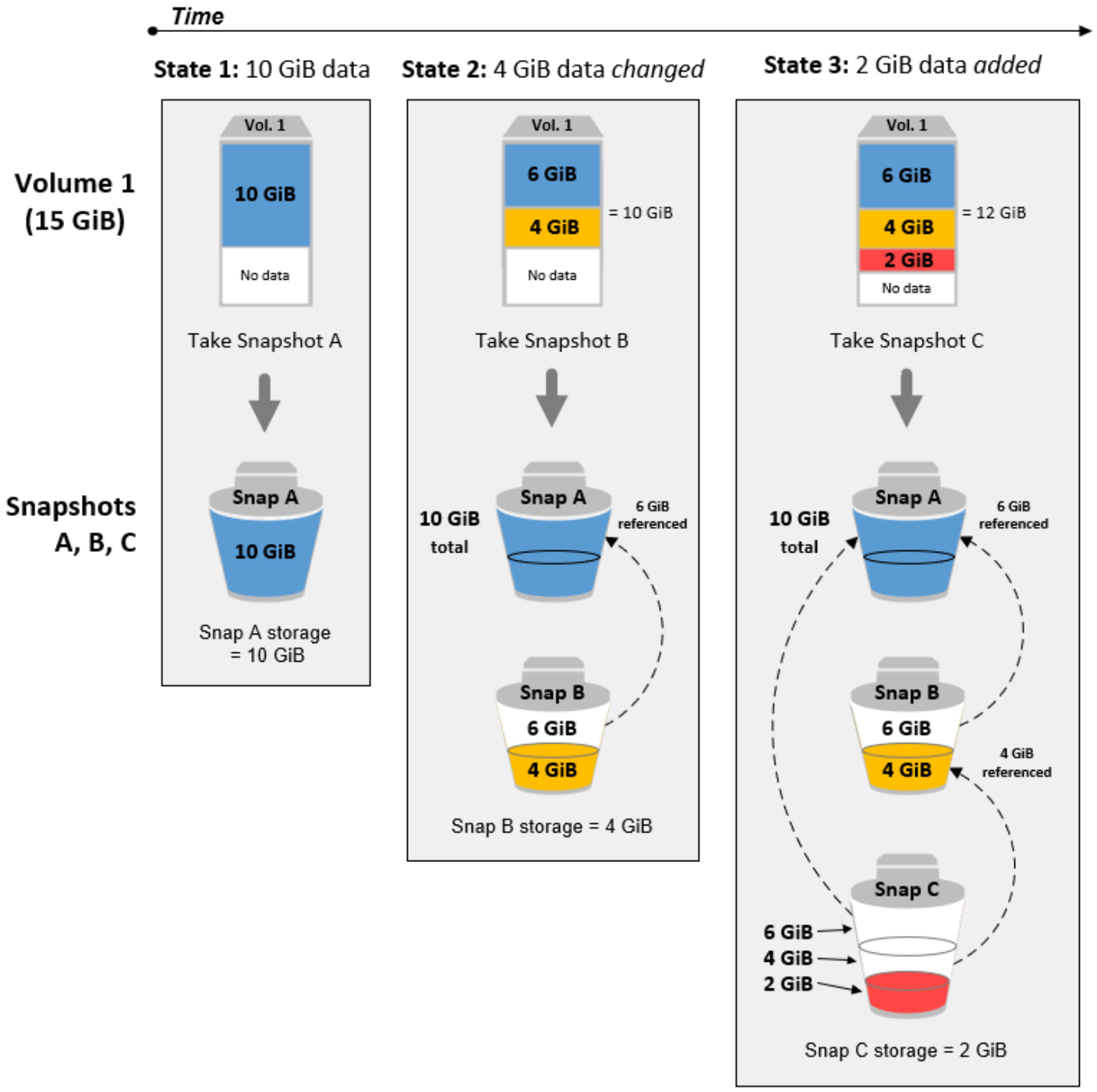
> **Important**
>
> 増分スナップショットをアーカイブすると、スナップショットの作成時にボリュームに書き込まれたすべてのブロックを含むフルスナップショットに変換されます。その後、Amazon EBS Snapshots Archive 階層に移動されます。アーカイブ階層のスナップショットは、標準階層のスナップショットとは異なるレートで請求されます。詳細については、「Amazon EBS スナップショットをアーカイブするための料金と請求」を参照してください。

次のセクションでは、EBS スナップショットがある時点でのボリュームの状態をキャプチャする方法、および変化するボリュームの後続のスナップショットが変更の履歴を作成する方法を説明します。

同じボリュームの複数のスナップショット

このセクションの図は、15 GiB サイズのボリューム 1 で 3 つの時点を示しています。これら 3 つのボリューム状態それぞれのスナップショットが作成されます。この図が特に示しているのは、次の点です。

- 状態 1 では、ボリュームに 10 GiB のデータがあります。スナップ A は、ボリュームの最初のスナップショットを取得したものです。スナップ A はフルスナップショットで、10 GiB データ全体がバックアップされます。
- 状態 2 では、ボリュームにはまだ 10 GiB のデータが含まれていますが、スナップ A の撮影後に変化したのは 4 GiB だけです。スナップ B は増分スナップショットです。変更された 4 GiB 部分だけをバックアップする必要があります。既にスナップ A にバックアップされている、変更がなかった残りの 6 GiB 分のデータは、スナップ B によって再度バックアップされるのではなく、参照されます。これは点線の矢印によって示されます。
- 状態 3 では、スナップ B が取得された後、2 GiB 分のデータがボリュームに追加され、合計 12 GiB となっています。スナップ C は増分スナップショットです。スナップ C は、スナップ B が作成されたあとに追加された 2 GiB 分のみコピーする必要があります。点線の矢印で示されているように、スナップ C はスナップ B に格納された 4 GiB 分のデータと、スナップ A に格納された 6 GiB 分のデータの両方を参照します。
- 3 つのスナップショットに必要な合計ストレージ量は 16 GiB です。これにより、スナップ A が 10 GiB、スナップ B が 4 GiB、スナップ C が 2 GiB になります。



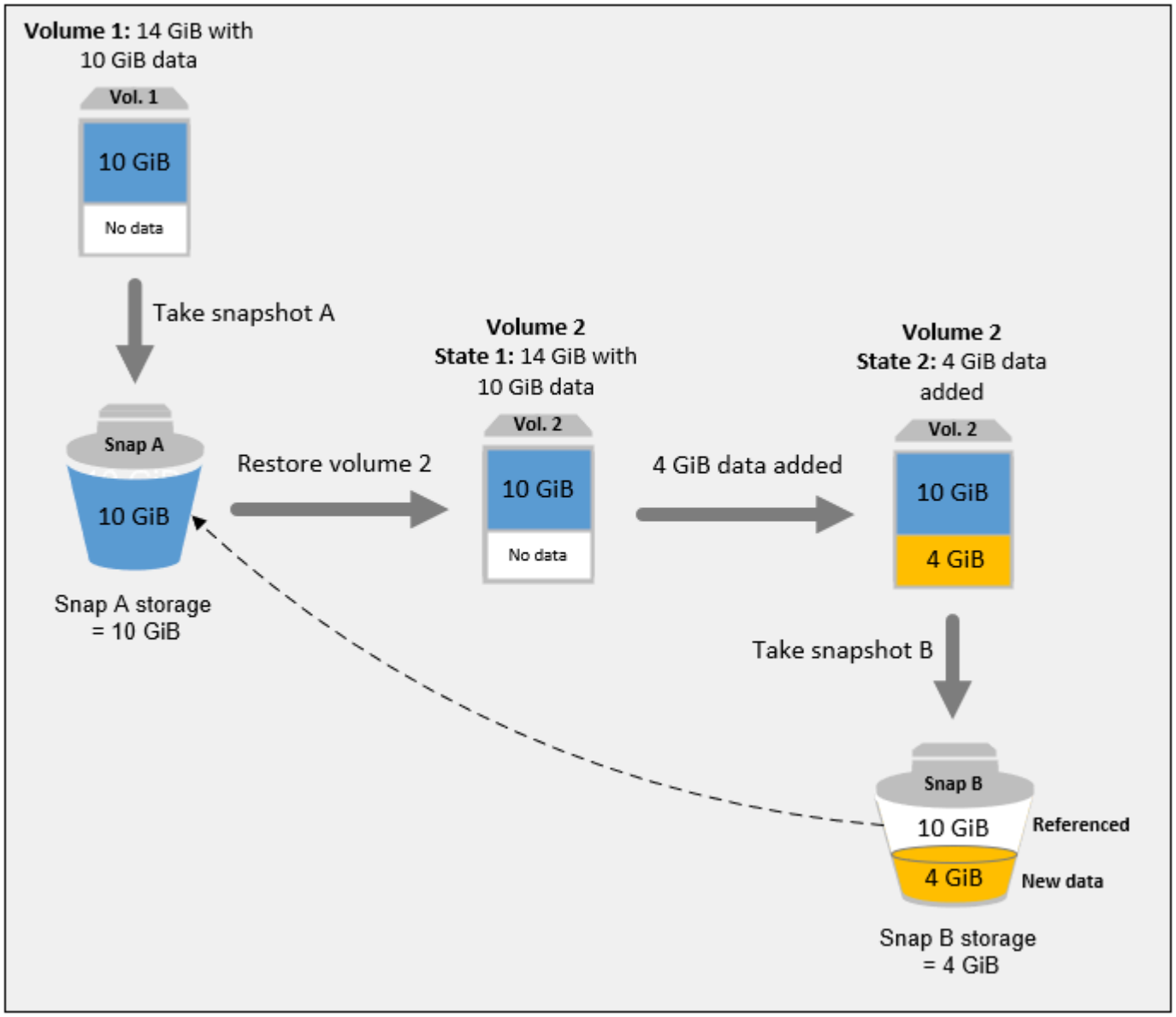
異なるボリュームの増分スナップショット

このセクションの図は、異なるボリュームから増分スナップショットを取得する方法を示しています。

1. 14 GiB 分のサイズのボリューム 1 は、10 GiB のデータがあります。スナップ A はボリュームで撮影された最初のスナップショットであるため、フルスナップショットであり、10 GiB のデータ全体がバックアップされます。
2. ボリューム 2 はスナップ A から作成されるので、スナップショットを取得した時点では ボリューム 1 の完全なレプリカとなります。
3. 時間が経つと、ボリューム 2 に 4 GiB のデータが追加され、合計サイズは 14 GiB になります。
4. ボリューム 2 は、スナップ B から取得されます。スナップ B では、スナップ A からボリュームが作成された後に追加された 4 GiB のデータのみがバックアップされます。もう一方の変更されていない 10 GiB のデータは、既にスナップ A に格納されており、再びバックアップされる代わりに、スナップ B から参照されます。

   別のボリュームから作成されてはいますが、スナップ B はスナップ A の増分スナップショットです。

> **Important**
>
> この図では、Vol 1 とスナップ A を所有していて、Vol 2 が Vol 1 と同じ KMS キーで暗号化されていることを前提としています。Vol 1 が別の AWS アカウントによって所有されており、そのアカウントがスナップ A を使用して共有した場合、スナップ B は完全なスナップショットになります。または、Vol 2 が Vol 1 とは異なる KMS キーで暗号化されている場合、スナップ B はフルスナップショットになります。



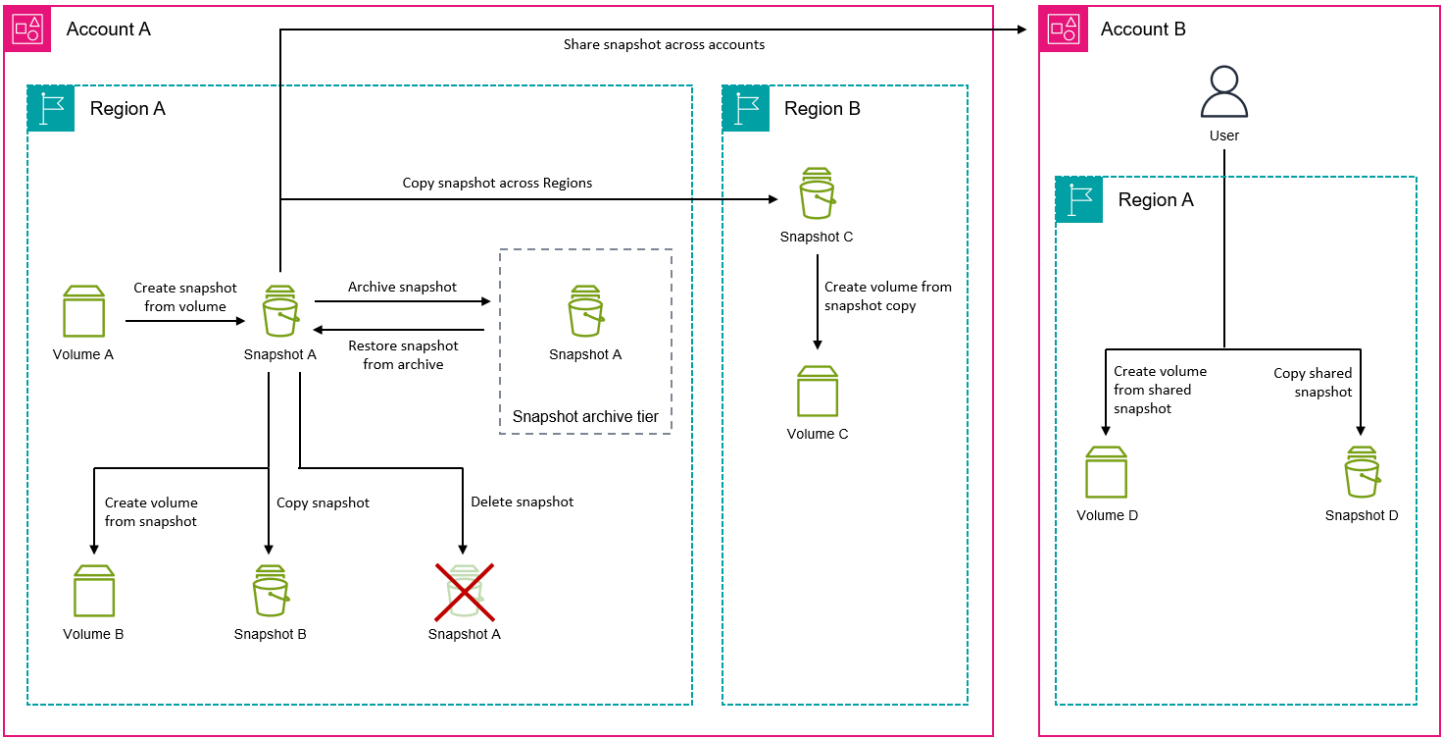
データが管理されている方法とスナップショットを削除するタイミングについては、Amazon EBS スナップショットの削除を参照してください。

Amazon EBS スナップショットのライフサイクル

Amazon EBS スナップショットのライフサイクルは、作成プロセスから始まります。Amazon EBS ボリュームからスナップショットを作成します。スナップショットを使用して新しい Amazon EBS ボリュームを復元できます。スナップショットのコピーは、同じリージョンまたは異なるリージョンで作成できます。スナップショットは AWS アカウント、パブリックまたはプライベートで他の と共有できます。これらのアカウントは、共有スナップショットからボリュームを復元するか、独自のアカウントで共有スナップショットのコピーを作成することができます。スナップショットにすぐ

にアクセスする必要がない場合、スナップショットをアーカイブしてストレージコストを節約できます。

次の図は、スナップショットライフサイクルの一部としてスナップショットに実行できるアクションを示しています。



タスク

- Amazon EBS スナップショットの作成
- Amazon EBS スナップショットに関する情報の表示
- Amazon EBS スナップショットのコピー
- Amazon EBS スナップショットを他の AWS アカウントと共有する
- Amazon EBS スナップショットのアーカイブ
- Amazon EBS スナップショットの削除

Amazon EBS スナップショットの作成

Amazon EBS ボリュームの Amazon EBS スナップショットを作成して、そのボリュームのポイントインタイムバックアップを作成できます。個々の Amazon EBS ボリュームのスナップショットを作成するか、Amazon EC2 インスタンスにアタッチされたボリュームのすべてまたはサブセットのマルチボリュームスナップショットを作成できます。

スナップショットの作成は非同期です。スナップショットはすぐに作成されますが、すべてのデータが Amazon S3 に転送されるまで pending 状態のままになります。これには、ボリュームで変更されたブロックの数に応じて、完了までに数時間かかることがあります。この間、スナップショットに影響を与えずにボリュームを引き続き使用できます。スナップショットには、スナップショットのリクエスト時にボリュームに書き込まれたデータのみが含まれます。アプリケーションやオペレーティングシステムによってキャッシュされたデータは含まれていません。

> **Tip**
>
> スナップショットの一貫性と完全性を確保するために、スナップショットを作成する前にボリュームへの書き込みを一時停止することをお勧めします。ボリュームへの書き込みを一時停止できない場合は、スナップショットを作成する前に、インスタンス内からボリュームのマウントを解除することをお勧めします。スナップショットが pending 状態になれば、再マウントして書き込みを再開できます。
>
> Amazon EC2 のルートデバイスとして機能するボリュームのスナップショットを作成する場合、スナップショットを取る前にインスタンスを停止することをお勧めします。

内容

- スナップショットの暗号化
- スナップショットの送信先
- スナップショットの自動化
- スナップショットを作成するための考慮事項
- EBS ボリュームのスナップショットを作成する
- EC2 インスタンスからマルチボリュームの EBS スナップショットを作成する

スナップショットの暗号化

スナップショットは、作成されたボリュームと同じ暗号化ステータスを自動的に取得します。暗号化されていないボリュームから作成されたスナップショットは暗号化されません。暗号化されたボリュームから作成されたスナップショットは、ボリュームと同じ KMS キーを使用して自動的に暗号化されます。

> **Tip**
>
> 暗号化されていないボリュームから暗号化されたスナップショットを作成する必要がある場合は、まずボリュームの暗号化されていないスナップショットを作成し、次にそのスナップショットの暗号化されたコピーを作成します。

スナップショットの送信先

ソースリソース (ボリュームまたはインスタンス) の場所によって、スナップショットを作成できる場所が決まります。

- ソースリソースがリージョンにある場合は、ソースリソースと同じリージョンにスナップショットを作成する必要があります。
- ソースリソースがローカルゾーンにある場合は、同じローカルゾーンまたはその親リージョンにスナップショットを作成できます。詳細については、「ローカルゾーン内のローカルスナップショット」を参照してください。
- ソースボリュームが Outpost にある場合は、同じ Outpost またはその親リージョンにスナップショットを作成できます。詳細については、「Amazon EBS local snapshots on Outposts」を参照してください。

スナップショットの自動化

Amazon Data Lifecycle Manager と AWS Backup を使用して、スナップショットの作成を自動化できます。

スナップショットを作成するための考慮事項

- Amazon EC2 インスタンスにアタッチされているボリュームで、休止状態になっている、または休止状態が有効になっているスナップショットは作成しないことをお勧めします。詳細については、「Amazon EC2 インスタンスの休止の仕組み」を参照してください。
- ボリュームの前のスナップショットが pending 状態の間でもボリュームのスナップショットを作成できますが、1 つのボリュームに複数の pending 状態のスナップショットがあると、スナップショットが完了するまでボリュームのパフォーマンスが低下する場合があります。
- pending 状態で実行できるスナップショットの数と、ボリュームタイプごとにリクエストできる同時スナップショットの数には制限があります。詳細については、「Quotas for Amazon EBS」を

参照してください。これらのクォータのいずれかを超える場合は、現在のスナップショットが完了するまで待ってから、もう一度試してください。

EBS ボリュームのスナップショットを作成する

1 つのボリュームの 1 つのスナップショットを作成できます。または、スナップショットの作成を自動化するには、Amazon Data Lifecycle Manager または AWS Backup を使用します。EC2 インスタンスのすべてのボリュームのスナップショットを作成するには、マルチボリュームスナップショットを使用します。

Console

スナップショットを作成するには

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. ナビゲーションペインで、[Snapshots] (スナップショット)、[Create snapshot] (スナップショットの作成) の順にクリックします。
3. [リソースタイプ] で、[ボリューム] を選択します。
4. [Volume ID] (ボリューム ID) で、スナップショットを作成するボリュームを選択します。[暗号化] フィールドは、ボリュームと、結果として生じるスナップショットの暗号化ステータスを示します。変更することはできません。
5. (オプション) [説明] には、スナップショットの簡潔な説明を入力します。
6. ボリュームが Outpost またはローカルゾーンにある場合、[スナップショットの送信先] フィールドが表示されます。次のいずれかを行います。

   - ボリュームがローカルゾーンにある場合は、[ローカルゾーン] を選択して同じローカルゾーンにスナップショットを作成するか、[AWS リージョン] を選択してローカルゾーンの親リージョンにスナップショットを作成します。
   - ボリュームが Outpost にある場合は、[AWS Outpost] を選択して同じ Outpost にスナップショットを作成するか、[AWS リージョン] を選択して Outpost の親リージョンにスナップショットを作成します。

> **Note**
>
> ボリュームがリージョンにある場合、[スナップショットの送信先] は表示されません。スナップショットは、ボリュームと同じリージョンに自動的に作成されます。

7. (オプション) スナップショットにカスタムタグを割り当てるには、[タグ] セクションで [タグの追加] を選択し、キーと値のペアを入力します。最大 50 個のタグを追加できます。
8. [スナップショットを作成] を選択します。

AWS CLI

スナップショットを作成するには

create-snapshot コマンドを使用します。

```
aws ec2 create-snapshot  \
    --volume-id vol-01234567890abcdef \
    --description "Snapshot of the root volume for i-1234567890abcdef0"
```

PowerShell

スナップショットを作成するには

New-EC2Snapshot コマンドレットを使用します。

```
New-EC2Snapshot `
    -VolumeId vol-01234567890abcdef `
    -Description "Snapshot of the root volume for i-1234567890abcdef0"
```

EC2 インスタンスからマルチボリュームの EBS スナップショットを作成する

デフォルトでは、Amazon EC2 インスタンスからマルチボリュームスナップショットを作成すると、Amazon EBS は、インスタンスにアタッチされているすべての Amazon EBS ボリュームのスナップショットを作成します。ただし、ルートボリュームを除外することも、必要に応じて特定のデータボリュームを除外することもできます。

> **Tip**
>
> マルチボリュームスナップショットは、まとめて識別および管理しやすいようにタグ付けすることをお勧めします。また、ソースボリュームから対応するスナップショットにタグをコピーして、アクセスポリシー、アタッチメント情報、コスト配分などのスナップショットメタデータを設定して、ソースボリュームと一致させることができます。

マルチボリュームスナップショットに関する考慮事項

- すべてのスナップショットが正常に完了すると、 の結果を含む createSnapshots CloudWatch イベントsucceededが AWS アカウントに送信されます。マルチボリュームスナップショットセットのいずれかのスナップショットが失敗すると、他のすべてのスナップショットは error 状態に陥り、createSnapshots CloudWatch イベントの結果として生じる failed がアカウントに送信されます。詳細については、「スナップショットの作成 (createSnapshots)」を参照してください。
- マルチボリュームスナップショットは、インスタンスにアタッチされる最大 128 個の Amazon EBS ボリュームをサポートしています。これには、ルートボリュームと最大 127 個のデータボリュームが含まれます。
- マルチボリュームスナップショットセットの各スナップショットは個別スナップショットで、個別スナップショットと同じ方法で使用でき、同じ機能をサポートしています。
- Amazon EC2 Windows インスタンスにアタッチされたすべての Amazon EBS ボリュームのアプリケーション整合性のあるスナップショットを取得するには、AWS Systems Manager コマンドドキュメントを使用します。

Console

マルチボリュームスナップショットを作成するには

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. ナビゲーションペインで、[Snapshots] (スナップショット)、[Create snapshot] (スナップショットの作成) の順にクリックします。
3. [リソースタイプ] で、[Instance (インスタンス)] を選択してください。
4. [Description] (説明) に、スナップショットの簡潔な説明を入力します。この説明は、すべてのスナップショットに適用されます。
5. インスタンスが Outpost またはローカルゾーンにある場合、[スナップショットの送信先] フィールドが表示されます。次のいずれかを行います。

   - インスタンスがローカルゾーンにある場合は、[ローカルゾーン] を選択して同じローカルゾーンにスナップショットを作成するか、[AWS リージョン] を選択してローカルゾーンの親リージョンにスナップショットを作成します。
   - インスタンスが Outpost にある場合は、[AWS Outpost] を選択して同じ Outpost にスナップショットを作成するか、[AWS リージョン] を選択して Outpost の親リージョンにスナップショットを作成します。

> **Note**
>
> インスタンスがリージョンにある場合、[スナップショットの送信先] は表示されません。スナップショットは、インスタンスと同じリージョンに自動的に作成されます。

6. (オプション) インスタンスのルートボリュームを除外するには、[ルートボリュームを除外] を選択します。
7. (オプション) データボリュームを除外するには、[特定のデータボリュームを除外] を選択します。[Attached data volumes] (アタッチされたデータボリューム) セクションには、選択したインスタンスに現在アタッチされているすべてのデータボリュームが一覧表示されます。

   除外するデータボリュームを選択します。マルチボリュームスナップショットセットには、未選択のままになっているボリュームのみが含まれます。
8. (オプション) ソースボリュームから対応するスナップショットにタグを自動的にコピーするには、[ボリュームからタグをコピー] で [タグをコピー] を選択します。
9. (オプション) スナップショットに追加のカスタムタグを割り当てるには、[タグ] セクションで [タグの追加] を選択し、キーと値のペアを入力します。最大 50 個のタグを追加できます。
10. [スナップショットを作成] を選択します。

AWS CLI

マルチボリュームスナップショットを作成するには

create-snapshots コマンドを使用します。

```
aws ec2 create-snapshots \
    --instance-specification InstanceId=i-1234567890abcdef0 \
    --description "from a multi-volume snapshot of i-1234567890abcdef0"
```

ルートボリュームを除外するには、`--instance-specification` オプションに以下を追加します。

```
ExcludeBootVolume=true
```

データボリュームを除外するには、--instance-specification オプションに以下を追加します。

```
ExcludeDataVolumeIds=vol-01234567890abcdef
```

PowerShell

マルチボリュームスナップショットを作成するには

New-EC2SnapshotBatch コマンドレットを使用します。

```
New-EC2SnapshotBatch `
    -InstanceSpecification_InstanceId i-1234567890abcdef0 `
    -Description "from a multi-volume snapshot of i-1234567890abcdef0"
```

ルートボリュームを除外するには、次のパラメータを追加します。

```
-InstanceSpecification_ExcludeBootVolume $true
```

データボリュームを除外するには、次のパラメータを追加します。

```
-InstanceSpecification_ExcludeDataVolumes "vol-01234567890abcdef"
```

Amazon EBS スナップショットに関する情報の表示

スナップショットに関する詳細情報を表示できます。

Console

スナップショットに関する情報を表示するには

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. ナビゲーションペインで、[Snapshots] を選択します。
3. 所有しているスナップショットだけを表示するには、画面の左上隅で、[Owned by me] (自分が所有) を選択します。タグとスナップショット属性を使用してスナップショットをフィルターすることもできます。[Filter] (フィルター) フィールドで、属性フィールドを選択し、属性値を選択または入力します。たとえば、暗号化されたスナップショットだけを表示するには、[Encryption] (暗号化) を選択し、[true] を入力します。

4. 特定のスナップショットの詳細情報を表示するには、リストで ID を選択します。

> **Note**
>
> [フルスナップショットサイズ] フィールドには、スナップショットのフルサイズがバイト単位で表示されます。これはスナップショットの増分サイズではなく、スナップショットの作成時にソースボリュームに書き込まれたすべてのブロックのサイズを表します。
> [ボリュームサイズ] フィールドには、他のサイズが指定されていない場合にスナップショットから作成される EBS ボリュームのサイズが表示されます。

AWS CLI

スナップショットに関する情報を表示するには

describe-snapshots コマンドを使用します。

Example例 1: タグに基づくフィルタリング

次の例は、Stack=production タグでスナップショットの詳細を示します。

```
aws ec2 describe-snapshots --filters Name=tag:Stack,Values=production
```

Example例 2: ボリュームに基づくフィルタリング

次の例は、指定されたボリュームから作成されたスナップショットの詳細を示します。

```
aws ec2 describe-snapshots --filters Name=volume-id,Values=vol-049df61146c4d7901
```

Example例3: スナップショットの経過日に基づくフィルタリング

JMESPath の式を使用して結果をフィルタリングできます。例えば、次のコマンドを実行すると、指定した日より前にアカウントで作成したすべてのスナップショットの ID が表示されます。所有者を指定しない場合、結果にはすべてのパブリックスナップショットが含まれます。

```
aws ec2 describe-snapshots \
    --filters Name=owner-id,Values=123456789012 \
    --query "Snapshots[?(StartTime<='2024-03-31')].[SnapshotId]" \
```

```
    --output text
```

次のコマンドは指定した日付範囲で作成されたすべてのスナップショットの ID を表示します。

```
aws ec2 describe-snapshots \
    --filters Name=owner-id,Values=123456789012 \
    --query "Snapshots[?(StartTime>='2024-01-01') && (StartTime<='2024-12-31')].
[SnapshotId]" \
    --output text
```

PowerShell

スナップショットに関する情報を表示するには

Get-EC2Snapshot コマンドレットを使用します。

例 1: スナップショットを説明する

次の の例では、指定したスナップショットを示しています。

```
Get-EC2Snapshot -SnapshotId snap-0abcdef1234567890
```

例 2: ボリュームに基づくフィルタリング

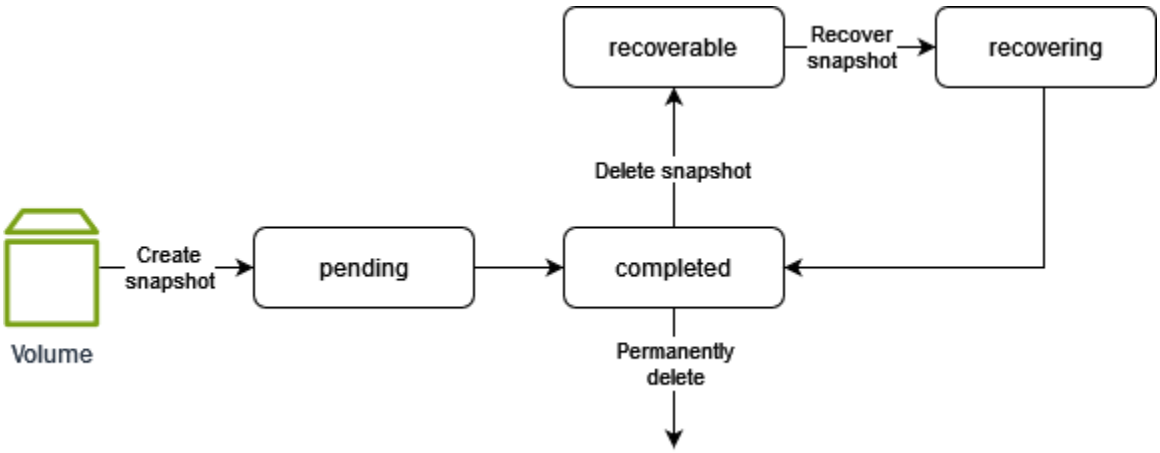
次の例は、指定されたボリュームから作成されたスナップショットの詳細を示します。

```
Get-EC2Snapshot`
    -Filter @{Name="volume-id"; Values="vol-01234567890abcdef"}
```

スナップショットの状態

Amazon EBS スナップショットは、作成されてから完全に削除されるまで、さまざまな状態に移行します。

次の図は、スナップショットの状態間の移行を示しています。スナップショットを作成すると、pending 状態になります。スナップショットが使用可能になると、completed 状態になります。スナップショットが不要になったら、削除できます。ごみ箱の保持ルールに一致するスナップショットを削除した場合、ごみ箱に保持されてから recoverable 状態になります。ごみ箱からスナップショットを復元した場合、recovering 状態になった後に completed 状態になります。それ以外の場合、スナップショットは完全に削除されます。



次の表はスナップショットの状態をまとめたものです。

ステータス	説明
pending	スナップショットの作成プロセスはまだ進行中です。スナップショットは、pending 状態にある間は使用できません。
completed	スナップショットの作成プロセスが完了し、スナップショットが使用可能になりました。
recoverable	スナップショットは現在、ごみ箱にあります。スナップショットを使用するには、まずごみ箱から復元する必要があります。
recovering	スナップショットはごみ箱から復元中です。スナップショットが復元されると、completed 状態に移行して使用可能になります。
error	スナップショットの作成プロセスが失敗しました。スナップショットが error 状態にある場合は使用できません。

Amazon EBS スナップショットのコピー

スナップショットを作成し、completed 状態になったら、スナップショットコピーを作成できます。スナップショットコピーは元のスナップショットの正確なコピーですが、一意のリソース ID を

持っています。所有しているスナップショットと、共有されているスナップショットをプライベートまたはパブリックにコピーできます。次のユースケースでは、スナップショットをコピーする必要がある場合があります。

- 地理的拡張 — アプリケーションを新しいリージョンで起動します。
- 移行 — アプリケーションを新しい送信先に移動して、可用性を向上させるか、またはコストを最小化します。
- ディザスタリカバリ — データの冗長性のために、データとログをセカンダリリージョンにバックアップする必要があります。
- 暗号化 — 以前に暗号化されていないスナップショットを暗号化するか、別の KMS キーを使用して暗号化されたスナップショットを再暗号化する必要があります。
- 共有スナップショットのコピー — 共有されているスナップショットをコピーする必要があります。
- データ保持と監査の要件 — 監査またはデータ保持のためにデータを保持するには、暗号化されたスナップショットをあるアカウントから別の AWS アカウントにコピーする必要があります。別のアカウントを使用すると、メイン AWS アカウントが侵害された場合に保護されます。

マルチボリュームスナップショットをコピーするには、作成時に割り当てたタグを使用して、そのセットの一部であるスナップショットをすべて識別し、個別にコピーします。

Amazon RDS スナップショットのコピーについては、『Amazon RDS ユーザーガイド』のDB スナップショットのコピーを参照してください。

料金

スナップショットのコピーの料金の詳細については、「Amazon EBS の料金」を参照してください。

内容

- スナップショットのコピーに関する考慮事項
- スナップショットコピーの送信先
- 増分スナップショットコピー
- Amazon EBS スナップショットと EBS-backed AMI の時間ベースのコピー
- 暗号化とスナップショットのコピー
- スナップショットをコピーする

スナップショットのコピーに関する考慮事項

- VM AWS Marketplace Import/Export、Storage Gateway スナップショットをコピーできますが、スナップショットがコピー先リージョンでサポートされていることを確認する必要があります。
- スナップショットの同時コピーリクエストは、送信先リージョンあたり 20 に制限されています。このクォータを超えると、ResourceLimitExceeded エラーが発生します。このエラーが発生した場合は、1 つ以上のコピー要求が完了するのを待ってから、新しいスナップショットコピー要求を作成してください。
- ユーザー定義タグは元のスナップショットからスナップショットコピーへコピーされません。コピー操作中または操作後に、ユーザー定義タグを追加することができます。
- スナップショットコピーオペレーションによって作成されたスナップショットには、vol-ffff や vol-ffffffff などの任意のボリューム ID が付されます。これらの任意のボリューム ID は、いかなる目的にも使用しないでください。
- スナップショットコピー操作のために指定されたリソースレベルのアクセス許可は、スナップショットコピーとソーススナップショットに適用できます。例については、「例: スナップショットのコピー」を参照してください。
- 高速スナップショット復元が有効化されているスナップショットをコピーすると、そのスナップショットコピーの高速スナップショット復元は自動で有効化されまん。スナップショットコピーの高速スナップショット復元を明示的に有効にする必要があります。
- スナップショットをコピーして、新しい KMS キーで暗号化すると、完全な (増分ではない) コピーが作成されます。その結果、追加のストレージコストが発生します。
- スナップショットを新しいリージョンにコピーすると、完全な (増分ではない) コピーが作成されます。その結果、追加のストレージコストが発生します。
- 外部またはクロスリージョンのデータ転送を使用する場合は、EC2 データ転送の追加料金が適用されます。開始後にスナップショットを削除しても、転送済みのデータに対して料金が課されます。

スナップショットコピーの送信先

ソーススナップショットの場所によって、コピーできるかどうかが決まります。

- 1 つのリージョンにあるソーススナップショットは、そのリージョン内、別のリージョン、そのリージョンに関連付けられた Outpost、またはそのリージョン内のローカルゾーンにコピーできます。

- 1 つのローカルゾーンにあるソーススナップショットは、同じローカルゾーン内、同じゾーングループ内の別のローカルゾーン、またはそのローカルゾーンの親リージョンにコピーできます。
- ソーススナップショットが Outpost にある場合、コピーすることはできません。

増分スナップショットコピー

同じ KMS キーを使用する同じアカウントとリージョン内のスナップショットコピー操作は常に増分コピーになります。ただし、別の KMS キーを使用してスナップショットコピーを暗号化すると、そのコピーは完全なコピーになります。

リージョンまたはアカウントにわたってスナップショットをコピーする場合、次の条件に合致すればコピーは増分となります。

- スナップショットが送信先のリージョンまたはアカウントにコピーされたことがある。
- 最新のスナップショットコピーが送信先のリージョンまたはアカウントにまだ存在する。
- 最新のスナップショットコピーがまだアーカイブされていない。
- 送信先のリージョンまたはアカウントのすべてのスナップショットのコピーが、暗号化されていないか、あるいは同じ KMS キーを使って暗号化されていた。

> **Tip**
>
> 送信先リージョンまたはアカウントで最近のボリュームのスナップショットコピーを追跡できるよう、スナップショットコピーにボリューム ID と作成時刻をタグ付けすること推奨します。

スナップショットコピーが増分かどうか確認するには、copySnapshot CloudWatch イベントをチェックします。

Amazon EBS スナップショットと EBS-backed AMI の時間ベースのコピー

時間ベースのコピーは、EBS スナップショットと EBS-backed AMIs がリージョン内および AWS リージョン間で指定された期間内にコピーされるようにすることで、データレプリケーションのコンプライアンス要件またはビジネス要件を満たすのに役立ちます。時間ベースのコピーは、バックアップ管理者が厳格なディザスタリカバリ要件 (目標復旧時点と目標復旧時間) を満たすのにも役立ちます。また、スナップショットと EBS-backed AMI のコピー時間が予測可能になるため、開発の俊敏性が向上します。

時間ベースのスナップショットおよび EBS-backed AMI コピー操作では、コピーを完了する完了期間を 15 分から 48 時間の間で指定します。完了期間は 15 分単位で指定する必要があります。

トピック

- クォータ
- 完了期間の決定
- 考慮事項
- モニタリング
- 料金と請求

クォータ

時間ベースのスナップショットおよび EBS-backed AMI コピー操作には、次のクォータが適用されます。

クォータ	説明	クォータ値	引き上げ可能
スナップショットコピー操作スループットのクォータ	1 回の時間ベースのスナップショットコピー操作で達成できる最大スループット。 **Note** AMI コピー操作の場合、クォータは AMI に関連付けられた個々のスナップショットに適用されます。	500 MiB/秒	いいえ
累積スナップショットコピースループットのクォータ	送信元リージョンと送信先リージョン間の時間ベースのスナ	2,000 MiB/秒	あり

クォータ	説明	クォータ値	引き上げ可能
	ップショットコピー操作を同時に実行することで達成できる最大累積スループット。 Note AMI コピー操作の場合、AMI に関連付けられた個々のスナップショットはクォータにカウントされます。		

時間ベースのスナップショットコピー操作を開始するときは、完了期間を指定します。リクエストで使用されるスループットは、スナップショットデータのサイズとリクエストされた完了期間によって決まります。例えば、225,000 MiB (0.214 TiB) のデータを含むスナップショットをコピーし、15 分間の完了期間をリクエストした場合、スループットは 250 MiB/秒 (225,000 MiB ÷ 15 分 = 250 MiB/秒) です。

時間ベースの AMI コピー操作を開始すると、指定した完了期間が AMI に関連付けられた各スナップショットに適用されます。各スナップショットのサイズは異なる可能性があるため、各スナップショットは異なるスループットでコピーされ、すべてのスナップショットが完了期間内にコピーされます。例えば、次のスナップショットが関連付けられている AMI があるとします。

- スナップショット 1: 200,000 MiB
- スナップショット 2: 500,000 MiB
- スナップショット 3: 450,000 MiB

この AMI の時間ベースのコピーを開始し、完了期間を 60 分と指定した場合、リクエストでは次のスループットが使用されます。

- スナップショット 1: 55.56 MiB/秒 (200,000 MiB ÷ 60 分 = 55.56 MiB/秒)
- スナップショット 2: 138.89 MiB/秒 (500,000 MiB ÷ 60 分 = 138.89 MiB/秒)
- スナップショット 3: 125 MiB/秒 (450,000 MiB ÷ 60 分 = 125 MiB/秒)

つまり、リクエストは 319.45 MiB/秒の累積スナップショットコピースループットクォータを使用して、コピーが 60 分で完了することを確認します。

時間ベースのスナップショットまたは EBS-backed AMI コピーリクエストを開始した場合、使用可能な累積スナップショットコピースループットのクォータは以下のようになります。

- 必要なスループットレート以上の場合、コピーはリクエストされた完了期間内に完了します。
- 必要なスループットレートより小さいがゼロより大きい場合、リクエストは成功しますが、リクエストされた完了期間よりも時間がかかります。コピーは使用可能なスループットクォータを使用して完了します。
- ゼロ (クォータに達した) の場合、リクエストは失敗します。

完了期間の決定

時間ベースのスナップショットまたは EBS-backed AMI コピー操作でリクエストできる最小完了期間は 15 分で、リクエストできる最大完了期間は 48 時間です。完了期間は 15 分単位で指定する必要があります。

時間ベースのスナップショットコピー操作の同時実行

すべての同時操作の合計スループットが累積スナップショットコピースループットのクォータ (デフォルトでは 2,000 MiB/秒) 内である限り、同じ送信元リージョンと送信先リージョン間で時間ベースのスナップショットコピー操作を同時に実行できます。

既存のスナップショットに必要な完了期間を達成できるかどうかを確認するには、すべてのスナップショットの合計サイズを必要な完了期間で割り、必要なスループットレートを決定します。

> **Tip**
>
> スナップショット内のデータの正確なサイズがわからない場合は、代わりにフルスナップショットサイズをプロキシとして使用できます。スナップショットの全サイズを取得するには、describe-snapshots AWS CLI コマンドを使用します。

```
required throughput rate = combined snapshot size ÷ required completion duration
```

必要なスループットレートが累積スナップショットコピースループットのクォータより低い場合は、必要な完了期間を達成できます。必要なスループットレートが累積スナップショットコピースループットのクォータより高い場合は、必要なスループットレートよりも 10% 以上高いクォータの引き上げをリクエストすることをお勧めします。

> **Tip**
>
> Amazon EC2 コンソールに用意されている計算ツールを使用することで、特定の累積スナップショットコピースループットクォータに基づいて、特定の期間に 2 つのリージョン間でコピーしたスナップショットデータの量と、そのデータ量で達成可能な最小完了期間を確認できます。計算ツールは `SnapshotCopyBytesTransferred` CloudWatch メトリクスを使用して、ある期間に 2 つのリージョン間でコピーされたデータを計算します。計算ツールを開くには、Amazon EC2 コンソールナビゲーションパネルで [スナップショット] を選択し、[アクション]、[コピー時間計算ツールを起動] を選択します。
> スナップショットコピー期間計算ツールは AWS Outposts、、Local Zones、および Wavelength Zones ではサポートされていません。

個々の時間ベースのスナップショットコピー操作

スナップショットデータのサイズをスナップショットコピー操作のスループットクォータ (500 MiB/秒) で割ることで、個々の時間ベースのスナップショットコピー操作の最小完了期間を計算できます。

> **Tip**
>
> スナップショット内のデータの正確なサイズがわからない場合は、代わりにフルスナップショットサイズをプロキシとして使用できます。スナップショットの全サイズを取得するには、describe-snapshots AWS CLI コマンドを使用します。

```
minimum completion duration = Max(15 minutes, (snapshot data size ÷ 500 MiB/s)
```

例えば、900,000 MiB のデータを含むスナップショットの最小完了期間は 30 分です。

```
minimum completion duration = Max(15 minutes, (900,000 MiB ÷ 500 MiB/s)
= Max(15 minutes, 30 minutes)
= 30 minutes
```

時間ベースの AMI コピー操作

単一の関連付けられたスナップショットを持つ EBS-backed AMI に対して時間ベースの AMI コピー操作を開始すると、個々の時間ベースのスナップショットコピー操作と同じ動作をし、同じスループット制限が適用されます。

複数の関連スナップショットを持つ EBS-backed AMI に対して時間ベースの AMI コピー操作を開始すると、同時に実行される時間ベースのスナップショットコピー操作と同じように動作し、同じスループット制限が適用されます。関連付けられたスナップショットごとに個別のスナップショットコピーリクエストが発生し、それぞれが累積スナップショットコピースループットクォータに影響します。指定した完了期間は、関連付けられている各スナップショットに適用されます。

考慮事項

- 同じリージョン内でスナップショットをコピーするとき、またはリージョン間でスナップショットをコピーするときに、時間ベースのスナップショットおよび EBS-backed AMI コピー操作を開始できます。
- 同じスナップショットまたは AMI に対して 2 つの時間ベースのコピー操作を開始した場合、2 番目のコピー操作の完了期間は、最初のコピー操作が完了した後にのみ開始されます。
- 時間ベースのコピーオペレーションとスナップショットコピー期間計算ツールは AWS Outposts、、ローカルゾーン、および Wavelength Zones ではサポートされていません。

モニタリング

Amazon EC2 コンソールと AWS CLIを使用して、時間ベースのスナップショットおよび EBS-backed AMI コピー操作の進行状況をモニタリングできます。コンソールでスナップショットを選択し、[詳細] タブで [進行状況] フィールドを調べます。を使用して AWS CLI、describe-snapshots コマンドレスポンスのProgress出力要素を検査します。

時間ベースのスナップショットまたは EBS-backed AMI コピー操作がリクエストされた完了期間内に完了したかどうかを確認するには、コンソールで [開始] と [完了] の時刻の差を確認するか、describe-snapshots のレスポンスで StartTime および CompletionTime を確認します。

copySnapshot Amazon EventBridge イベントを使用して、時間ベースのコピー操作の結果をモニタリングすることもできます。このイベントは、操作が完了したかどうか、およびリクエストされた完了期間が達成されたかどうかを示します。完了期間が達成されなかった場合、イベントには原因に関する詳細情報が記載されます。詳細については、「EBS スナップショットイベント」を参照してください。

料金と請求

> **Note**
>
> 標準のスナップショットコピー操作と同様に、スナップショットを新しいリージョンにコピーすると、完全な (増分ではない) コピーが作成され、追加のストレージコストが発生します。同じスナップショットの後続のコピーは増分です。また、外部またはクロスリージョンのデータ転送を使用する場合は、Amazon EC2 データ転送の追加料金が適用されます。

時間ベースのスナップショットおよび EBS-backed AMI コピー操作には、追加料金が適用されます。時間ベースのコピー操作は、コピーされたスナップショットデータの GiB ごとに、リクエストされた完了期間に基づくレートで課金されます。固定レートは次のとおりです。

> **Note**
>
> 完了期間は 15 分単位で指定する必要があります。最小完了期間は 15 分、最大完了期間は 48 時間です。

- 15 分 — データ 1 GiB あたり 0.020 USD

- 30 分と 45 分 — データ 1 GiB あたり 0.018 USD
- 1 時間 ~ 1 時間 45 分 — データ 1 GiB あたり 0.016 USD
- 2 時間 ~ 3 時間 45 分 — データ 1 GiB あたり 0.014 USD
- 4 時間 ~ 7 時間 45 分 — データ 1 GiB あたり 0.012 USD
- 8 時間 ~ 15 時間 45 分 — データ 1 GiB あたり 0.010 USD
- 16 時間以上 — データ 1 GiB あたり 0.005 USD

例えば、完了時間が 8 時間で 3,000 GiB のデータを含むスナップショットをコピーすると、30 USD (0.010 USD x 3,000 GiB) が請求されます。

時間ベースのコピー操作を開始したものの、クォータを超えたためにリクエストした完了期間が達成されなかった場合は、リクエストした完了期間ではなく、実際の完了期間に基づいて請求されます。例えば、1 時間の完了期間をリクエストしたものの、操作が 2 時間で完了した場合、2 時間の完了期間のレートに基づいて請求されます。

Amazon EBS がリクエストされた完了期間を達成できない場合、またはサービス側の問題によりリクエストがキャンセルされた場合、時間ベースのスナップショットコピー操作の追加料金は請求されません。

時間ベースのスナップショットコピー操作の進行中にスナップショットコピーを削除すると、その時点までにコピーされたデータに関する請求は、指定した完了期間に対応するレートで行われます。

暗号化とスナップショットのコピー

> **Note**
>
> Amazon S3 のサーバー側の暗号化 (256 ビット AES) は、コピー操作中に転送中のスナップショットのデータを保護します。

暗号化されていないソーススナップショットの暗号化されたスナップショットコピーを作成できます。また、ソーススナップショットとは異なる KMS キーを使用してスナップショットコピーを暗号化できます。ただし、コピー操作中にスナップショットコピーの暗号化状態を変更すると、完全コピー (増分ではない) が返され、より大規模なデータ転送およびストレージ料金が発生する可能性があります。

> **Tip**
>
> 共有された暗号化されたスナップショットを使用する場合は、自分が所有する KMS キーを使用してスナップショットを再暗号化することをお勧めします。これにより、KMS キーが侵害された、または所有者がアクセス許可を取り消したため、スナップショットを使用して作成した暗号化されたボリュームへのアクセスが失われる可能性がある場合でも保護されます。

暗号化されたスナップショットのコピーのアクセス許可

暗号されたスナップショットをコピーするには、ユーザーに Amazon EBS 暗号化を使用するための次のアクセス許可が必要です。

- • `kms:DescribeKey`
  - `kms:CreateGrant`
  - `kms:GenerateDataKey`
  - `kms:GenerateDataKeyWithoutPlaintext`
  - `kms:ReEncrypt`
  - `kms:Decrypt`
- 別の AWS アカウントから共有されている暗号化されたスナップショットをコピーするには、そのスナップショットの暗号化に使用されたカスタマーマネージドキーを使用するアクセス許可が必要です。詳細については、「共有 Amazon EBS スナップショットの暗号化に使用される KMS キーを共有」を参照してください。

スナップショットコピーの暗号化の結果

ユーザーが所有するスナップショットおよび共有されたスナップショットをコピーする際の設定に関する表は次のとおりです。

送信先リージョンのデフォルトでの暗号化	ソーススナップショット	スナップショットコピーの暗号化の結果	メモ
Disabled	暗号化されていない	オプションの暗号化	コピーを暗号化する場合は、使用する KMS キーを指定できます。コピーを暗号化しても KMS キーを指定しない場合、暗号化用に指定されたキーがデフォルトで使用されます。
Disabled	暗号化された	自動暗号化	使用する KMS キーを指定できます。KMS キーを指定しない場合、 AWS マネージドキー (aws/ebs) が使用されます。
有効	暗号化されていない	自動暗号化	使用する KMS キーを指定できます。KMS キーを指定しない場合、暗号化用に指定されたキーがデフォルトで使用されます。
有効	暗号化された	自動暗号化	使用する KMS キーを指定できます。KMS キーを指定しない場合、暗号化用に指定されたキーがデフォルトで使用されます。

スナップショットをコピーする

スナップショットは、別のリージョンにコピーできます。また、暗号化されていないスナップショットを暗号化されたスナップショットにコピーすることも可能です。ただし、暗号化キーを使用するアクセス許可なしで、暗号化されたスナップショットをコピーしようとすると、操作は警告なしに失敗し、スナップショットコピーには [指定されたキー ID にアクセスできません] というステータスメッセージが表示されます。

Console

スナップショットをコピーするには

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. ナビゲーションペインで、[Snapshots] を選択します。

3. コピーするスナップショットを選択し、[Actions] (アクション)、[Copy snapshot] (スナップショットのコピー) の順にクリックします。
4. [Description] (説明) に、スナップショットのコピーの簡潔な説明を入力します。

   デフォルトでは、スナップショットとコピーを見分けられるよう、元のスナップショットに関する情報が説明に含まれています。
5. スナップショットコピーの送信先を指定します。

   - スナップショットを同じリージョンまたは別のリージョンにコピーするには、[AWS リージョン] を選択し、送信先リージョンを選択します。
   - スナップショットをローカルゾーンにコピーするには、[AWS ローカルゾーン] を選択し、送信先のローカルゾーンを選択します。
   - (Outpost のお客様のみ) スナップショットを Outpost にコピーするには、[AWS Outpost] を選択し、送信先 Outpost の ARN を入力します。
6. 特定の期間内にスナップショットコピーを完了する必要がある場合は、[時間ベースのコピーを有効にする] を選択します。[完了期間] には、必要な完了期間を 15 分単位で入力します。詳細については、Amazon EBS スナップショットと EBS-backed AMI の時間ベースのコピーを参照してください。

   スナップショットコピーを特定の期間内に完了する必要がない場合は、時間ベースのコピーを有効にしないでください。この場合、スナップショットのコピーはベストエフォートベースで完了します。
7. (Outpost のお客様のみ) 選択したリージョンの Outpost にスナップショットコピーを作成するには、[スナップショットの送信先] で [AWS Outpost] を選択し、[送信先の Outpost の ARN] で、スナップショットをコピーする Outpost の ARN を入力します。[スナップショットの送信先] フィールドは、選択したリージョンに Outpost がある場合にのみ表示されます。
8. スナップショットコピーの暗号化ステータスを指定します。

   ソースのスナップショットが暗号化されている場合、またはアカウントで [デフォルトで暗号化] を有効にしている場合、スナップショットコピーは自動的に暗号化されます。ソーススナップショットが暗号化されておらず、アカウントがデフォルトで暗号化を有効にしていない場合、暗号化はオプションです。
9. [スナップショットをコピー] を選択します。

AWS CLI

スナップショットを別のリージョンにコピーするには

copy-snapshot コマンドを使用します。次の例では、指定したスナップショットをソースリージョンから現在のリージョン (--region オプションで指定) にコピーしています。

```
aws ec2 copy-snapshot \
    --source-snapshot-id snap-0abcdef1234567890 \
    --source-region us-east-1 \
    --region us-west-2
```

暗号化されていないスナップショットから暗号化されたスナップショットを作成するには

copy-snapshot コマンドを使用します。次の例では、指定した暗号化されていないスナップショットをソースリージョンから現在のリージョンにコピーし、指定した KMS キーを使用して新しいスナップショットを暗号化しています。

```
aws ec2 copy-snapshot \
    --source-snapshot-id snap-0abcdef1234567890 \
    --source-region us-east-1 \
    --encrypted \
    --kms-key-id alias/my-kms-key
```

PowerShell

スナップショットを別のリージョンにコピーするには

Copy-EC2Snapshot コマンドレットを使用します。次の例では、指定したスナップショットをソースリージョンから現在のリージョン (--region オプションで指定) にコピーしています。

```
Copy-EC2Snapshot `
    -SourceSnapshotId snap-0abcdef1234567890 `
    -SourceRegion us-east-1 `
    -Region us-west-2
```

暗号化されていないスナップショットから暗号化されたスナップショットを作成するには

Copy-EC2Snapshot コマンドレットを使用します。次の例では、指定した暗号化されていないスナップショットをソースリージョンから現在のリージョンにコピーし、指定した KMS キーを使用して新しいスナップショットを暗号化しています。

```
Copy-EC2Snapshot `
    -SourceSnapshotId snap-0abcdef1234567890 `
    -SourceRegion us-east-1 `
    -Encrypted $true `
    -KmsKeyId alias/my-kms-key
```

Amazon EBS スナップショットを他の AWS アカウントと共有する

スナップショットの許可を変更することで、他の AWS アカウントとスナップショットを共有できます。スナップショットは、他のすべての AWS アカウントとパブリックに共有することも、指定した個々の AWS アカウントとプライベートに共有することもできます。許可を受けたユーザーは、共有するスナップショットを使用して自分の EBS ボリュームを作成できますが、元のスナップショットは影響を受けません。

> **Important**
>
> スナップショットを共有すると、スナップショットのすべてのデータに他人がアクセスできるようになります。スナップショットの共有は、自分のスナップショットデータすべてを委託できる人とだけ行ってください。

スナップショットがパブリックに共有されないようにするために、Amazon EBS スナップショットのブロックパブリックアクセス を有効にします。

トピック

- スナップショットを共有する前に
- スナップショットの共有
- 共有 Amazon EBS スナップショットの暗号化に使用される KMS キーを共有
- 共有されている Amazon EBS スナップショットを使用
- 共有するスナップショットの使用方法を決定する

スナップショットを共有する前に

スナップショットの共有には、次の考慮事項が適用されます。

- そのリージョンでスナップショットのブロックパブリックアクセスが有効になっている場合、スナップショットをパブリックに共有しようとする試みはブロックされます。スナップショットは引き続きプライベートに共有できます。
- スナップショットは、スナップショットが作成されたリージョンに制限されます。別のリージョンとスナップショットを共有するには、そのリージョンにスナップショットをコピーして、そのコピーを共有します。詳細については、「Amazon EBS スナップショットのコピー」を参照してください。
- デフォルトの AWS マネージドキーで暗号化されたスナップショットを共有することはできません。共有できるのは、カスタマーマネージド型キーを使用して暗号化されたスナップショットだけです。詳細については、AWS Key Management Service デベロッパーガイドのキーの作成を参照してください。
- 暗号化されていないスナップショットのみをパブリックに共有できます。
- 暗号化されたスナップショットを共有する場合は、スナップショットの暗号化に使用するカスタマーマネージド型キーも共有する必要があります。詳細については、共有 Amazon EBS スナップショットの暗号化に使用される KMS キーを共有を参照してください。

スナップショットの共有

スナップショットはパブリックに共有することも、特定の AWS アカウントと共有することもできます。

Console

スナップショットを共有するには

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. ナビゲーションペインで、[Snapshots] を選択します。
3. スナップショットを選択し、[Actions] (アクション) 、[Modify Permissions] (権限の変更) の順にクリックします。
4. スナップショットの権限を指定します。[Current setting] (現在の設定) は、スナップショットの現在の共有権限を示します。
   - スナップショットをすべての AWS アカウントとパブリックに共有するには、「パブリック」を選択します。
   - スナップショットを特定の AWS アカウントとプライベートに共有するには、プライベート を選択します。次に、[Sharing accounts] (アカウントの共有) セクション

で、[Add account] (アカウントの追加) を選択し、共有するアカウントの 12 桁のアカウント ID (ハイフンなし) を入力します。

5. [Save changes] (変更の保存) をクリックします。

AWS CLI

スナップショットのアクセス許可は、スナップショットの `createVolumePermission` 属性を使用して指定します。スナップショットを公開するには、グループを `all` に設定します。スナップショットを特定の AWS アカウントと共有するには、ユーザーを AWS アカウントの ID に設定します。

スナップショットをパブリックに共有するには

modify-snapshot-attribute コマンドを使用します。

`--attribute` で、`createVolumePermission` を指定します。`--operation-type` で、`add` を指定します。`--group-names` で、`all` を指定します。

```
aws ec2 modify-snapshot-attribute \
    --snapshot-id snap-0abcdef1234567890 \
    --attribute createVolumePermission \
    --operation-type add \
    --group-names all
```

スナップショットをプライベートに共有するには

modify-snapshot-attribute コマンドを使用します。

`--attribute` で、`createVolumePermission` を指定します。`--operation-type` で、`add` を指定します。では`--user-ids`、スナップショットを共有する AWS アカウントの 12 桁の IDs を指定します。

```
aws ec2 modify-snapshot-attribute \
    --snapshot-id snap-0abcdef1234567890 \
    --attribute createVolumePermission \
    --operation-type add \
    --user-ids 123456789012 111122223333
```

PowerShell

スナップショットのアクセス許可は、スナップショットの createVolumePermission 属性を使用して指定します。スナップショットを公開するには、グループを all に設定します。スナップショットを特定の AWS アカウントと共有するには、ユーザーを AWS アカウントの ID に設定します。

スナップショットをパブリックに共有するには

Edit-EC2SnapshotAttribute コマンドレットを使用します。

-Attribute で、CreateVolumePermission を指定します。-OperationType で、Add を指定します。-GroupName で、all を指定します。

```
Edit-EC2SnapshotAttribute `
    -SnapshotId snap-0abcdef1234567890 `
    -Attribute CreateVolumePermission `
    -OperationType Add `
    -GroupName all
```

スナップショットをプライベートに共有するには

Edit-EC2SnapshotAttribute コマンドレットを使用します。

-Attribute で、CreateVolumePermission を指定します。-OperationType で、Add を指定します。ではUserId、スナップショットを共有する AWS アカウントの 12 桁の IDs を指定します。

```
Edit-EC2SnapshotAttribute `
    -SnapshotId snap-0abcdef1234567890 `
    -Attribute CreateVolumePermission `
    -OperationType Add `
    -UserId 123456789012 111122223333
```

共有 Amazon EBS スナップショットの暗号化に使用される KMS キーを共有

暗号化されたスナップショットを共有する場合は、スナップショットの暗号化に使用するカスタマーマネージド型キーも共有する必要があります。カスタマーマネージド型キーを作成したとき、または後でカスタマーマネージド型キーにクロスアカウント権限を適用することができます。

暗号化されたスナップショットにアクセスしている共有のカスタマーマネージド型キーのユーザーには、そのキーに対して、次の操作を実行するためのアクセス許可が与えられている必要があります。

- `kms:DescribeKey`
- `kms:CreateGrant`
- `kms:GenerateDataKey`
- `kms:GenerateDataKeyWithoutPlaintext`
- `kms:ReEncrypt`
- `kms:Decrypt`

> **Tip**
>
> 最小権限のプリンシパルに従うには、`kms:CreateGrant` へのフルアクセスを許可しないでください。代わりに、 `kms:GrantIsForAWSResource` 条件キーを使用して、 AWS サービスによってユーザーに代わって権限が作成された場合にのみ、ユーザーが KMS キーに権限を作成できるようにします。

カスタマーマネージド型キーへのアクセスの制御方法については、AWS Key Management Service デベロッパーガイドのAWS KMSでのキーポリシーの使用を参照してください。

AWS KMS コンソールを使用してカスタマーマネージドキーを共有するには

1. https://console.aws.amazon.com/kms で AWS KMS コンソールを開きます。
2. を変更するには AWS リージョン、ページの右上隅にあるリージョンセレクターを使用します。
3. ナビゲーションペインで、[カスタマー管理型のキー] を選択します。
4. [エイリアス] 列で、スナップショットの暗号化に使用したカスタマーマネージド型キーのエイリアス (テキストリンク) を選択します。キーの詳細が新しいページで開きます。
5. [キーポリシー] セクションに、ポリシービューまたはデフォルトビューのいずれかが表示されます。ポリシービューは、キーポリシードキュメントを表示します。デフォルトビューは、[キー管理者]、[キーの削除]、[キーの使用]、[その他の AWS アカウント] の各セクションを表示します。デフォルトビューは、コンソールでポリシーを作成し、それをカスタマイズしていない場合に表示されます。デフォルトビューが使用できない場合は、ポリシービューでポリシーを手動で編集する必要があります。詳細については、AWS Key Management Service デベロッパーガイドのキーポリシーの表示 (コンソール)を参照してください。

アクセス可能なビューに応じて、ポリシービューまたはデフォルトビューのいずれかを使用して、次のように 1 つ以上の AWS アカウント IDsをポリシーに追加します。

- (ポリシービュー) [編集] を選択します。次のステートメントに 1 つ以上の AWS アカウント IDs を追加します: `"Allow use of the key"`および `"Allow attachment of persistent resources"`。[Save changes] (変更の保存) をクリックします。次の例では、 AWS アカウント ID 444455556666がポリシーに追加されます。

```
{
  "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}}
}
```

- (デフォルトビュー) 他の AWS アカウントまでスクロールダウンします。他の AWS アカウントを追加を選択し、プロンプトに従って AWS アカウント ID を入力します。別のアカウ

ントを追加するには、別の AWS アカウントを追加を選択し、 AWS アカウント ID を入力します。すべての AWS アカウントを追加したら、[Save changes] (変更の保存) を選択します。

共有されている Amazon EBS スナップショットを使用

暗号化されていない共有スナップショットを使用するには

ID または説明で共有スナップショットを見つけます。このスナップショットは、アカウント内で所有している、他のスナップショットと同じように使用できます。例えば、スナップショットからボリュームを作成したり、別のリージョンにコピーしたりすることができます。

暗号化された共有スナップショットを使用するには

ID または説明で共有スナップショットを見つけます。アカウントに共有スナップショットのコピーを作成し、所有している KMS キーを使用して、そのコピーを暗号化します。その後、コピーを使用してボリュームを作成したり、別のリージョンにコピーしたりできます。

Console

スナップショットのアクセス許可を表示するには

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. ナビゲーションペインで、[スナップショット] を選択します。
3. スナップショットを選択します。
4. フィルターが [自分が共有] である場合、スナップショットはこのアカウントによって所有されています。フィルターが [プライベートスナップショット] の場合、スナップショットはこのアカウントによって所有されているか、このアカウントと明確に共有されています。スナップショットを選択し、[詳細] タブの [所有者] で、このアカウントが指定されているか、別のアカウントが指定されているかを確認します。

AWS CLI

スナップショットのアクセス許可を表示するには

describe-snapshot-attribute コマンドを使用して、指定したスナップショットのスナップショットアクセス許可を取得します。

```
aws ec2 describe-snapshot-attribute \
```

```
    --snapshot-id snap-0abcdef1234567890 \
    --attribute createVolumePermission
```

以下は出力の例です。

```
{
    "SnapshotId": "snap-0abcdef1234567890",
    "CreateVolumePermissions": [
        {
            "UserId": "111122223333"
        }
    ]
}
```

PowerShell

スナップショットのアクセス許可を表示するには

Get-EC2SnapshotAttribute コマンドレットを使用します。

```
(Get-EC2SnapshotAttribute `
    -SnapshotId snap-0abcdef1234567890 `
    -Attribute createVolumePermission).CreateVolumePermissions
```

以下は出力の例です。

```
Group UserId
----- ------
      111122223333
```

共有するスナップショットの使用方法を決定する

を使用して AWS CloudTrail 、他のユーザーと共有したスナップショットがコピーされているか、ボリュームの作成に使用されているかをモニタリングできます。共有したスナップショットに対してアクションを実行すると、次のイベントが CloudTrail に ログ記録されます。

- SharedSnapshotCopyInitiated — 共有スナップショットをコピーしています。
- SharedSnapshotVolumeCreated — ボリュームを作成するために共有スナップショットを使用しています。

CloudTrail の使用の詳細については、「Log Amazon EC2 and Amazon EBS API calls with AWS CloudTrail」を参照してください。

Amazon EBS スナップショットのアーカイブ

Amazon EBS Snapshots Archive は、頻繁な検索や高速な検索を必要としない、アクセス頻度の低いスナップショットを低コストで長期保存するために使用できるストレージ階層です。

デフォルトでは、スナップショットを作成すると、Amazon EBS スナップショットスタンダード階層に保存されます(標準階層)。標準階層に保存されるスナップショットは、増分です。これは、最新のスナップショットが作成された後に変更された、ボリューム内のブロックのみが保存されることを意味します。

スナップショットをアーカイブすると、増分スナップショットはフルスナップショットに変換され、標準階層から Amazon EBS Snapshots Archive 階層 (アーカイブ階層) に移動します。完全なスナップショットには、スナップショットの作成時にボリュームに書き込まれたすべてのブロックが含まれます。

アーカイブされたスナップショットにアクセスする必要がある場合は、アーカイブ階層から標準階層に復元し、アカウント内の他のスナップショットと同じ方法で使用できます。

Amazon EBS Snapshots Archive では、90 日以上保存する予定で、ほとんどアクセスする必要のないスナップショットに対して、スナップショットストレージコストを最大 75% 削減できます。

一般的なユースケースを以下に示します。

- プロジェクト終了時のスナップショットなど、ボリューム内の唯一のスナップショットをアーカイブする
- コンプライアンス上の理由から、ある時点の、完全な増分スナップショットをアーカイブします。
- 毎月、四半期ごと、または年ごとの増分スナップショットをアーカイブします。

トピック

- クォータ
- Amazon EBS スナップショットをアーカイブする際の考慮事項と制限
- Amazon EBS スナップショットをアーカイブするための料金と請求
- Amazon EBS スナップショットのアーカイブに関するガイドラインとベストプラクティス
- Amazon EBS スナップショットのアーカイブに必要な IAM アクセス許可
- Amazon EBS スナップショットをアーカイブ

- アーカイブされた Amazon EBS スナップショットを復元
- 一時的に復元された Amazon EBS スナップショットの復元期間を変更
- Amazon EBS スナップショットアーカイブを表示
- CloudWatch Events を使用して Amazon EBS スナップショットアーカイブをモニタリング

クォータ

このセクションでは、アーカイブされたスナップショットと進行中のスナップショットのデフォルトのクォータについて説明します。

クォータ	デフォルトのクォータ			
ボリュームあたりのアーカイブされたスナップショット	25			
アカウントごとの進行中のスナップショットアーカイブ	25			
アカウントごとの進	5			

クォータ	デフォルトのクォータ			
行中のスナップショットリストア				

デフォルトの制限を超える場合は、 サポート Center Create ケースフォームに入力して制限の引き上げをリクエストします。

Amazon EBS スナップショットをアーカイブする際の考慮事項と制限

Amazon EBS スナップショットをアーカイブするときは、次の点に注意してください。

考慮事項

- 最小アーカイブ期間は 90 日です。90 日間の最小アーカイブ期間より前にアーカイブされたスナップショットを削除または永続的に復元すると、アーカイブ階層の残りの日数に対して請求され、最も近い時間に四捨五入されます。詳細については、Amazon EBS スナップショットをアーカイブするための料金と請求を参照してください。
- スナップショットのサイズによっては、アーカイブ階層から標準階層にアーカイブスナップショットを復元するのに最大 72 時間かかる場合があります。
- アーカイブされるスナップショットは、常に完全なスナップショットです。完全なスナップショットには、スナップショットの作成時にボリュームに書き込まれたすべてのブロックが含まれます。完全なスナップショットは、作成元の増分スナップショットよりも大きくなります。ただし、標準階層にボリュームのスナップショットが 1 つしかない場合、アーカイブ階層の完全なスナップショットのサイズは、標準階層のスナップショットと同じサイズになります。これは、ボリュームの最初のスナップショットが常に完全なスナップショットであるためです。スナップショットの全サイズを取得するには、describe-snapshots AWS CLI コマンドを使用します。
- 毎月、毎四半期、または毎年のスナップショットのアーカイブをお勧めします。単一ボリュームの毎日の増分スナップショットをアーカイブすると、標準ティアで保持する場合と比較してコストが高くなる可能性があります。
- スナップショットがアーカイブされると、スナップショット系統内の他のスナップショットによって参照されるスナップショットのデータは標準階層で保持されます。標準階層で保持される参照データに関連するデータおよびストレージコストは、系統内の次のスナップショットに割り当てら

れます。これにより、系列内の後続のスナップショットがアーカイブの影響を受けないことが保証されます。

- ごみ箱の保持ルールに一致するアーカイブスナップショットを削除すると、アーカイブされたスナップショットは、保持ルールで定義されている保持期間中、ごみ箱に保持されます。スナップショットを使用するには、まずそのスナップショットをごみ箱から復元し、次にアーカイブ階層から復元する必要があります。詳細については、「ごみ箱」および「Amazon EBS スナップショットをアーカイブするための料金と請求」を参照してください。
- アーカイブされたスナップショットをブロックデバイスマッピングに使用したり、Amazon EBS ボリュームを作成したりすることはできません。
- 、APIs AWS Backup コンソール、またはコマンドラインツールを使用して作成されたスナップショット AWS Backup をアーカイブできます。詳細については、「AWS Backup デベロッパーガイド」の「Creating a backup plan」を参照してください。

制限

- `completed` 状態にあるスナップショットのみアーカイブできます。
- アーカイブできるのは、アカウント内で所有しているスナップショットだけです。共有されているスナップショットをアーカイブするには、まずスナップショットをアカウントにコピーしてから、スナップショットコピーをアーカイブします。
- アーカイブされたスナップショットを使用するには、まずそのスナップショットを標準階層に復元する必要があります。`CreateVolume` および `RunInstances` API オペレーションでスナップショットからボリュームを作成する場合や、スナップショットを共有またはコピーする場合には、スタンダード階層への復元が必要です。詳細については、「アーカイブされた Amazon EBS スナップショットを復元」を参照してください。
- 1 つ以上の AMI に関連付けられているスナップショットをアーカイブできるのは、関連付けられている AMI がすべて無効になっている場合に限ります。詳細については、「AMI の無効化」を参照してください。
- 関連付けられたスナップショットが一時的に復元された場合、無効になっている AMI を有効にすることはできません。AMI を有効にする前に、関連付けられたすべてのスナップショットを完全に復元する必要があります。
- スナップショットのアーカイブまたはスナップショットの復元プロセスは、開始後にキャンセルできません。
- アーカイブされたスナップショットは共有できません。他のアカウントと共有しているスナップショットをアーカイブすると、スナップショットが共有されているアカウントは、スナップショットがアーカイブされた後にアクセスできなくなります。

- アーカイブされたスナップショットはコピーできません。アーカイブされたスナップショットをコピーする必要がある場合は、まずそのスナップショットを復元する必要があります。
- ローカルスナップショットでは、高速スナップショット復元を有効化できません。高速スナップショット復元は、スナップショットがアーカイブされると自動的に無効になります。高速スナップショット復元を使用する必要がある場合は、スナップショットを復元した後で、手動で有効にする必要があります。

Amazon EBS スナップショットをアーカイブするための料金と請求

アーカイブされたスナップショットは、1 GB あたり 0.0125 ドル/月 の料金が請求されます。たとえば、100 GiB のスナップショットをアーカイブすると、1 か月あたり 1.25 ドル (100 GiB * 0.0125 ドル) が請求されます。

スナップショットの復元は、復元されたデータ 1 GB あたり 0.03 ドルの料金が請求されます。たとえば、アーカイブ階層から 100 GiB のスナップショットを復元すると、3 ドル (100 GiB * 0.03 ドル) が 1 回請求されます。

スナップショットが標準階層に復元された後、スナップショットは 1 GB あたり 0.05 ドル のスナップショット標準料金が請求されます。

詳細については、Amazon EBS の料金表を参照してください。

最小アーカイブ期間の請求

最小アーカイブ期間は 90 日です。90 日間の最小アーカイブ期間より前にアーカイブされたスナップショットを削除または永続的に復元すると、残りの日分のアーカイブ階層ストレージ料金に相当する日数の日割り計算で、最も近い時間に四捨五入された料金が請求されます。たとえば、40 日後にアーカイブされたスナップショットを削除または完全に復元すると、最小アーカイブ期間の残りの 50 日間分の料金が請求されます。

> **Note**
>
> 最小アーカイブ期間が 90 日前にアーカイブされたスナップショットを一時的に復元しても、この料金は発生しません。

一時的復元

スナップショットを一時的に復元すると、スナップショットはアーカイブ階層から標準階層に復元され、スナップショットのコピーはアーカイブ階層に残ります。一時復元期間の間、標準階層のスナッ

プショットとアーカイブ層のスナップショットコピーの両方に対して請求されます。一時的に復元されたスナップショットが標準階層から削除されると、そのスナップショットに対する請求はなくなり、アーカイブ階層内のスナップショットに対してのみ請求されます。

永続的復元

スナップショットを永続的に復元すると、スナップショットはアーカイブ層から標準階層にリストアされ、スナップショットはアーカイブ階層から削除されます。スナップショットに対して請求されるのは、標準階層のみです。

スナップショットの削除

アーカイブ中にスナップショットを削除すると、すでにアーカイブ階層に移動されたスナップショットデータに対して請求されます。このデータには 90 日の最小アーカイブ期間が適用され、削除時にそれに応じて請求されます。例えば、100 GiB のスナップショットをアーカイブし、40 GiB のみがアーカイブされた後にスナップショットを削除すると、アーカイブ済みの 40 GiB の最小アーカイブ期間 90 日分の料金 1.50 USD が請求されます (1 GB あたり 0.0125 USD/月 * 40 GB * (90 日 * 24 時間) / (24 時間/日 * 30 日/月))。

アーカイブ階層からの復元中にスナップショットを削除すると、スナップショットのフルサイズのスナップショット復元に対して請求されます (スナップショットサイズ* 0.03 USD)。たとえば、アーカイブ層から 100 GiB のスナップショットを復元し、スナップショットの復元が完了する前にスナップショットを削除すると、3 USD (100 GiB のスナップショットサイズ* 0.03 ドル) が請求されます。

ごみ箱

アーカイブされたスナップショットは、ごみ箱に入っている間、アーカイブされたスナップショットの料金で請求されます。ごみ箱にあるアーカイブされたスナップショットには、90 日間の最小アーカイブ期間が適用され、最小アーカイブ期間より前にごみ箱によって削除された場合は、それに応じて請求されます。つまり、保持ルールによってアーカイブされたスナップショットがごみ箱から最低期間 90 日前に削除された場合、残りの日分の料金が請求されます。

スナップショットのアーカイブ中に保持ルールに一致するスナップショットを削除すると、アーカイブされたスナップショットは、保持ルールで定義されている保持期間中、ごみ箱に保持されます。これは、アーカイブされたスナップショットの料金で請求されます。

スナップショットのリストア中に保持ルールに一致するスナップショットを削除すると、復元されたスナップショットは保持期間の残りの期間ごみ箱に保持され、標準のスナップショットレートで請求されます。復元されたスナップショットを使用するには、まずそのスナップショットをごみ箱から復元する必要があります。

詳細については、「ごみ箱」を参照してください。

コスト追跡

アーカイブされたスナップショットは、同じリソース ID と Amazon リソースネーム (ARN) AWS Cost and Usage Report を持つ に表示されます。詳細については、「AWS Cost and Usage Report ユーザーガイド」を参照してください。

次の使用タイプを使用して、関連するコストを識別できます。

- `SnapshotArchiveStorage`— 月間データストレージの手数料
- `SnapshotArchiveRetrieval` — スナップショット復元の 1 回払い料金
- `SnapshotArchiveEarlyDelete`— 最小アーカイブ期間 (90 日) 前にスナップショットを削除または永続的に復元する手数料

Amazon EBS スナップショットのアーカイブに関するガイドラインとベストプラクティス

このセクションでは、スナップショットのアーカイブに関するガイドラインとベストプラクティスをいくつか示します。

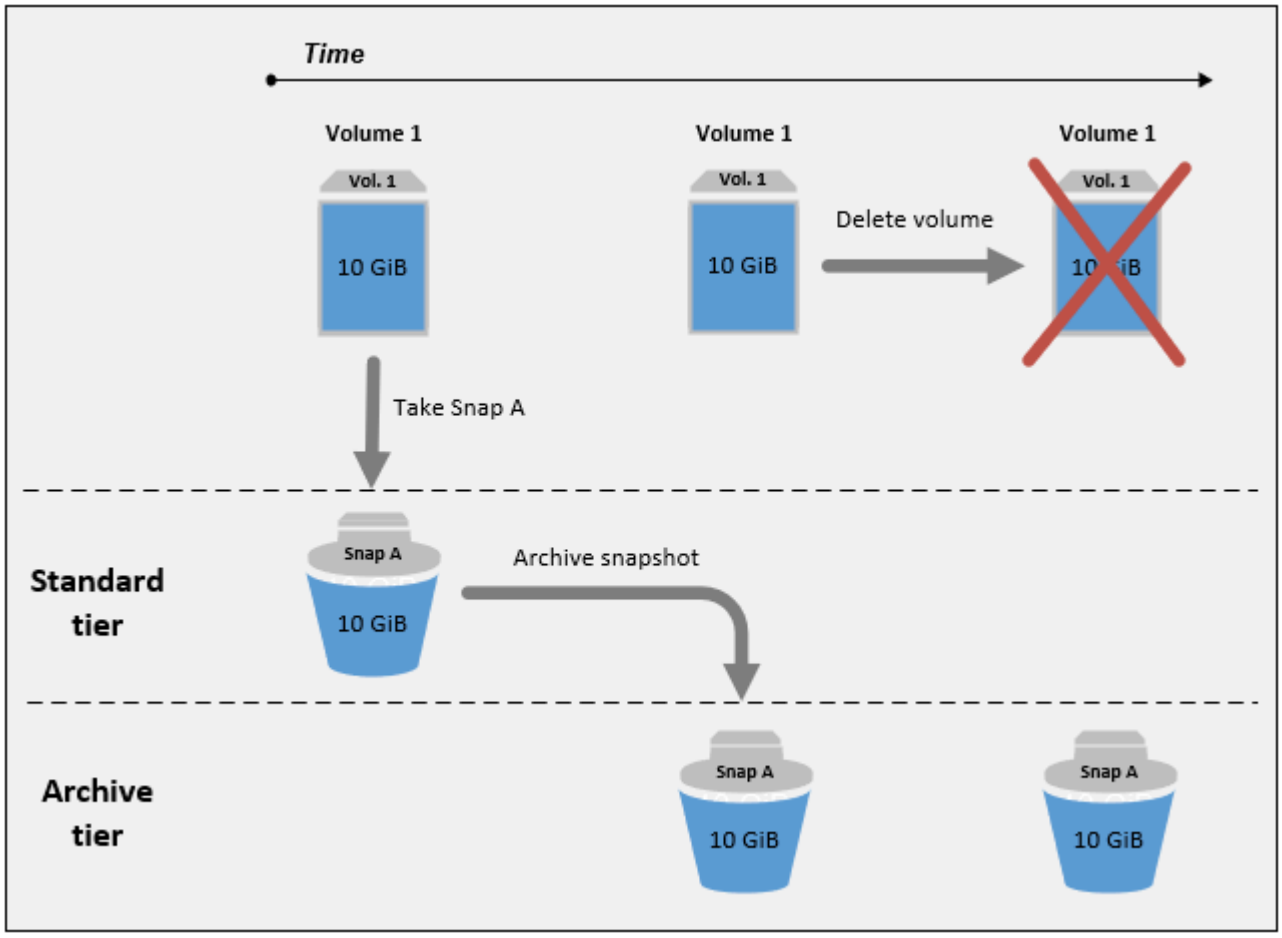
トピック

- ボリュームの 1 つのスナップショットをアーカイブする
- 単一ボリュームの増分スナップショットをアーカイブする
- コンプライアンス上の理由から完全なスナップショットをアーカイブする
- 標準階層のストレージコストの削減を決定する

ボリュームの 1 つのスナップショットをアーカイブする

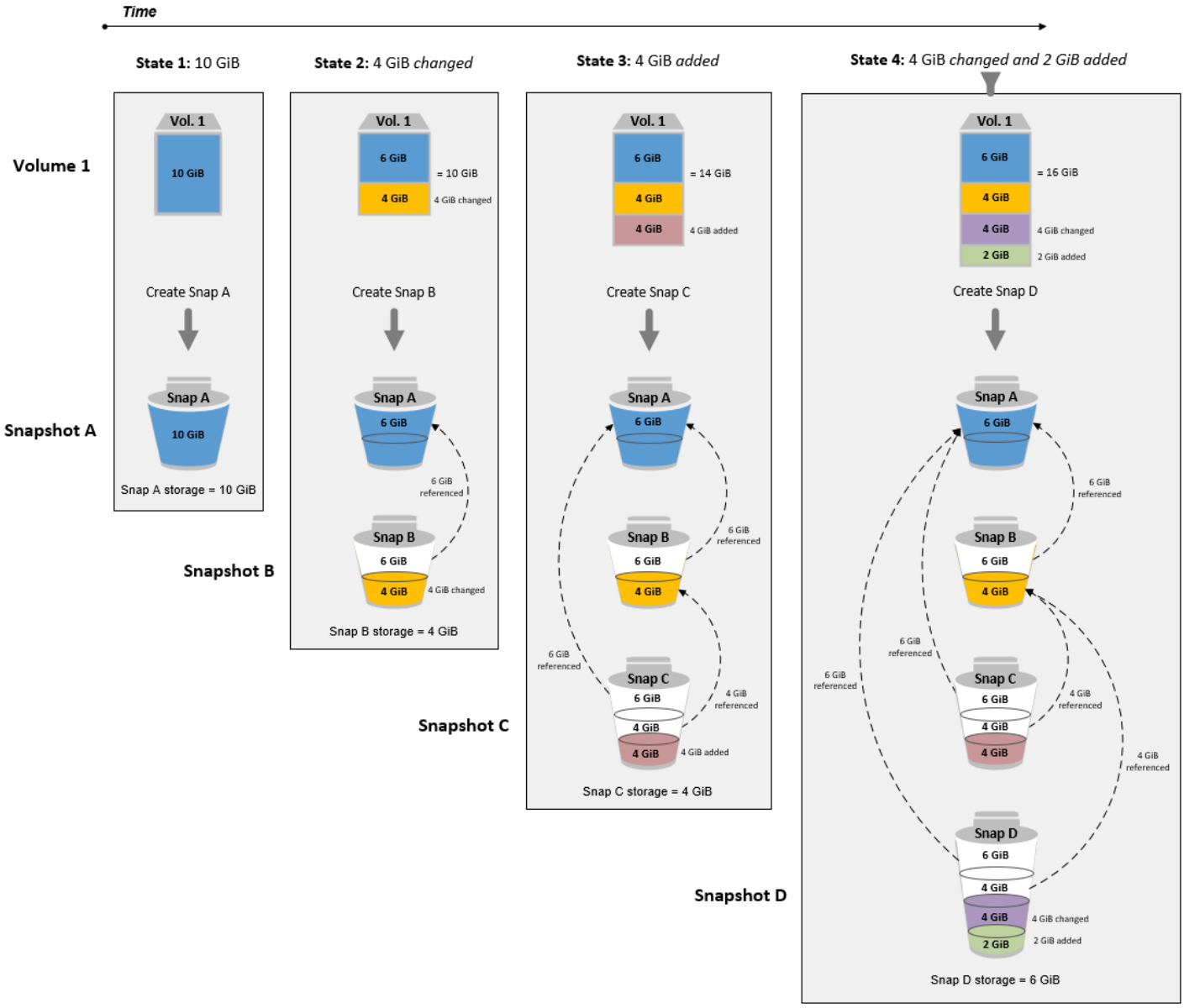
ボリュームのスナップショットが 1 つしかない場合、スナップショットは、スナップショットの作成時にボリュームに書き込まれたブロックと常に同じサイズになります。このようなスナップショットをアーカイブすると、標準階層のスナップショットは同等サイズのフルスナップショットに変換され、標準階層からアーカイブ階層に移動されます。

これらのスナップショットをアーカイブすることで、ストレージコストの削減に役立ちます。ソースボリュームが不要になった場合は、ストレージコストをさらに節約するためにボリュームを削除できます。



単一ボリュームの増分スナップショットをアーカイブする

増分スナップショットをアーカイブすると、スナップショットは完全なスナップショットに変換され、アーカイブ階層に移動されます。たとえば、以下の画像では、スナップ B をアーカイブする場合、スナップショットは 10 GiB の完全なスナップショットに変換され、アーカイブ階層に移動されます。同様に、スナップ C をアーカイブする場合の場合、アーカイブ階層内の完全なスナップショットのサイズは 14 GiB です。



標準階層のストレージコストを削減するためにスナップショットをアーカイブする場合は、増分スナップショットのセットに最初のスナップショットをアーカイブしないでください。これらのスナップショットは、スナップショット系統内の後続のスナップショットによって参照されます。ほとんどの場合、これらのスナップショットをアーカイブしても、ストレージコストは削減されません。

> **Note**
>
> 増分スナップショットのセットには、最後のスナップショットをアーカイブしないでください。最後のスナップショットは、ボリュームの最新のスナップショットです。ボリュームが

破損または紛失した場合に、標準階層からボリュームを作成する場合は、標準階層でこのスナップショットが必要になります。

後続のスナップショットで参照されるデータを含むスナップショットを系統でアーカイブすると、参照されるデータに関連付けられているデータストレージコストとストレージコストは、系統内の後続のスナップショットに割り当てられます。この場合、スナップショットをアーカイブしても、データストレージやストレージのコストは削減されません。例えば、上の図でスナップ B をアーカイブする場合、その 4 GiB のデータはスナップ C に帰属しています。この場合、アーカイブ階層のスナップ B のフルバージョンのストレージコストが発生し、標準階層のストレージコストは変わらないため、全体のストレージコストが増加します。

スナップ C をアーカイブする場合、データが系統の後半で他のスナップショットによって参照されないため、標準階層ストレージは 4 GiB 減少します。また、スナップショットが完全なスナップショットに変換されるため、アーカイブ階層のストレージは 14 GiB 増加します。

コンプライアンス上の理由から完全なスナップショットをアーカイブする

コンプライアンス上の理由から、毎月、四半期、または年単位でボリュームのフルバックアップを作成する必要がある場合があります。これらのバックアップでは、スナップショット系統内の他のスナップショットへの後方参照または前方参照なしで、スタンドアロンスナップショットが必要になる場合があります。EBS Snapshots Archive でアーカイブされたスナップショットは完全なスナップショットであり、系統内の他のスナップショットへの参照はありません。さらに、コンプライアンスのために、これらのスナップショットを数年間保持する必要があります。EBS スナップショットアーカイブを使用すると、これらの完全なスナップショットを長期保存するためにコストパフォーマンスに優れた方法でアーカイブできます。

標準階層のストレージコストの削減を決定する

増分スナップショットをアーカイブしてストレージコストを削減する場合は、アーカイブ階層の完全なスナップショットのサイズと、標準階層のストレージの削減を考慮する必要があります。ここでは、その方法について説明します。

Important

API レスポンスは、API が呼び出された時点での正確なデータです。API レスポンスは、スナップショットの系統が変更された結果、スナップショットに関連付けられたデータが変わるため、異なる可能性があります。

標準階層でのストレージおよびストレージコストの削減を判断するには、次のステップに従います。

1. アーカイブするスナップショットについて、フルスナップショットサイズとその作成元のソースボリュームを確認します。describe-snapshots コマンドを使用し、アーカイブするスナップショットの ID を --snapshot-id に指定します。

```
$ aws ec2 describe-snapshots --snapshot-id snapshot_id
```

FullSnapshotSizeInBytes のレスポンス値はフルスナップショットサイズをバイト単位で示し、VolumeId のレスポンス値はソースボリュームの ID を示します。

たとえば、次のコマンドは、スナップショット snap-09c9114207084f0d9 に関する情報を返します。

```
$ aws ec2 describe-snapshots --snapshot-id snap-09c9114207084f0d9
```

次の出力例は、フルスナップショットサイズが 5678912341 バイト (5.28 GiB) で、ソースボリュームが vol-0f3e2c292c52b85c3 であることを示しています。

```
{
    "Snapshots": [
        {
            "Description": "",
            "Tags": [],
            "Encrypted": false,
            "VolumeId": "vol-0f3e2c292c52b85c3",
            "State": "completed",
            "VolumeSize": 8,
            "StartTime": "2021-11-16T08:29:49.840Z",
            "Progress": "100%",
            "OwnerId": "123456789012",
            "FullSnapshotSizeInBytes" : "5678912341",
            "SnapshotId": "snap-09c9114207084f0d9"
        }
    ]
}
```

2. ソースボリュームから作成されたすべてのスナップショットを検索します。describe-snapshots コマンドを使用します。volume-id フィルターを指定して、フィルター値に、前のステップで取得したボリューム ID を指定します。

```
$ aws ec2 describe-snapshots --filters "Name=volume-id, Values=volume_id"
```

たとえば、次のコマンドは、ボリューム vol-0f3e2c292c52b85c3 から作成されたすべてのスナップショットを返します。

```
$ aws ec2 describe-snapshots --filters "Name=volume-id,
 Values=vol-0f3e2c292c52b85c3"
```

以下のコマンド出力は、ボリューム vol-0f3e2c292c52b85c3 から作成された 3 つのスナップショットを示します。

```
{
    "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"
        }
    ]
}
```

3. 前のコマンドの出力を使用して、スナップショットを最も古いものから新しいものまで作成時刻順にソートします。各スナップショットの StartTime レスポンスパラメータには、そのスナップショットの作成時刻が UTC 時間形式で表示されます。

   例えば、前のステップで返されたスナップショットは、最も古いものから新しいものまで作成時刻順に整理され、次のようになります。

   1. snap-08ca60083f86816b0 (最古 — アーカイブしたいスナップショットの前に作成される)
   2. snap-09c9114207084f0d9 (アーカイブするスナップショット)
   3. snap-024f49fe8dd853fa8 (最新 — アーカイブしたいスナップショットの後に作成される)

4. アーカイブしたいスナップショットの直前と直後に作成されたスナップショットを特定します。この場合、スナップショット snap-09c9114207084f0d9 をアーカイブします。これは、3 つのスナップショットのセットで作成された 2 番目の増分スナップショットです。スナップショット snap-08ca60083f86816b0 は直前に作成され、スナップショット snap-024f49fe8dd853fa8 が直後に作成されます。

5. アーカイブしたいスナップショット内の参照されていないデータを検索します。まず、アーカイブしたいスナップショットの直前に作成されたスナップショットと、アーカイブするスナップショットの間で異なるブロックを見つけます。list-changed-blocks コマンドを使用します。--first-snapshot-id に、アーカイブしたいスナップショットの直前に作成されたスナップショットの ID を指定します。--second-snapshot-id に、アーカイブしたいスナップショットの ID を指定します。

   ```
   $ aws ebs list-changed-blocks --first-snapshot-id snapshot_created_before --second-snapshot-id snapshot_to_archive
   ```

   たとえば、次のコマンドは、スナップショット snap-08ca60083f86816b0 (アーカイブしたいスナップショットの前に作成されたスナップショット) とスナップショット

snap-09c9114207084f0d9 (アーカイブしたいスナップショット) の間で異なるブロックのブロックインデックスを表示します。

```
$ aws ebs list-changed-blocks --first-snapshot-id snap-08ca60083f86816b0 --second-
snapshot-id snap-09c9114207084f0d9
```

以下に、一部のブロックが省略されたコマンド出力を示します。

```
{
    "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
}
```

次に、同じコマンドを使用して、アーカイブするスナップショットとその直後に作成されたスナップショットとの間で異なるブロックを検索します。--first-snapshot-id に、アーカイブしたいスナップショットの ID を指定します。--second-snapshot-id に、アーカイブしたいスナップショットの直後に作成されたスナップショットの ID を指定します。

```
$ aws ebs list-changed-blocks --first-snapshot-id snapshot_to_archive --second-
snapshot-id snapshot_created_after
```

たとえば、次のコマンドは、スナップショット snap-024f49fe8dd853fa8 (アーカイブしたいスナップショットの後に作成されたスナップショット) とスナップショット snap-09c9114207084f0d9 (アーカイブしたいスナップショット) の間で異なるブロックのブロックインデックスを表示します。

```
$ aws ebs list-changed-blocks --first-snapshot-id snap-09c9114207084f0d9 --second-
snapshot-id snap-024f49fe8dd853fa8
```

以下に、一部のブロックが省略されたコマンド出力を示します。

```
{
    "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
}
```

6. 前のステップで両方のコマンドで返された出力を比較します。両方のコマンド出力に同じブロックインデックスが表示される場合は、ブロックに参照されていないデータが含まれていることを示します。

   例えば、前のステップのコマンド出力では、ブロック 4、5、13、14 がスナップショット `snap-09c9114207084f0d9` に固有のものであり、スナップショットの系統の中で他のスナップショットから参照されていないことを示しています。

   標準階層ストレージの削減を判断するには、両方のコマンド出力に表示されるブロック数に 512 KiB (スナップショットブロックサイズ) を掛けます。

   たとえば、9,950 ブロックインデックスが両方のコマンド出力に表示されている場合、標準階層ストレージが約 4.85 GiB (9,950 ブロック * 512 KiB = 4.85 GiB) 減少することを示します。

7. 参照されていないブロックを標準階層に 90 日間格納するためのストレージコストを決定します。この値を、ステップ 1 で説明した完全なスナップショットをアーカイブ階層に格納するコストと比較します。最低 90 日間はアーカイブ階層から完全なスナップショットを復元しないことを前提として、値を比較することで、コスト削減を判断できます。詳細については、「Amazon EBS スナップショットをアーカイブするための料金と請求」を参照してください。

Amazon EBS スナップショットのアーカイブに必要な IAM アクセス許可

デフォルトでは、ユーザーにはスナップショットのアーカイブを使う許可がありません。ユーザーがスナップショットのアーカイブを使用するには、特定のリソースと API アクションを使用する許可を付与する IAM ポリシーを作成する必要があります。詳細については、「IAM ユーザーガイド」の「IAM ポリシーの作成」を参照してください。

スナップショットのアーカイブを使用するには、次の許可をユーザーに付与する必要があります。

- `ec2:DescribeSnapshotTierStatus`
- `ec2:ModifySnapshotTier`
- `ec2:RestoreSnapshotTier`

コンソールユーザーには、`ec2:DescribeSnapshots` のような追加の許可が必要になる場合があります。

暗号化されたスナップショットをアーカイブおよび復元するには、次の追加の AWS KMS アクセス許可が必要です。

- kms:CreateGrant
- kms:Decrypt
- kms:DescribeKey

以下は、暗号化されたスナップショットと暗号化されていないスナップショットをアーカイブ、復元、表示する許可を IAM ユーザーに付与する IAM ポリシーの例です。これには、コンソールユーザーの ec2:DescribeSnapshots 許可が含まれます。一部の許可が不要な場合は、ポリシーから削除できます。

> **Tip**
>
> 最小権限のプリンシパルに従うには、kms:CreateGrant へのフルアクセスを許可しないでください。代わりに、次の例に示すように、 AWS サービスによってユーザーに代わって権限が作成された場合にのみ、 kms:GrantIsForAWSResource条件キーを使用して KMS キーに対する権限の作成をユーザーに許可します。

JSON

```
{
    "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
                }
            }
    }]
```

```
}
```

アクセス許可を付与するにはユーザー、グループ、またはロールにアクセス許可を追加します。

- のユーザーとグループ AWS IAM アイデンティティセンター:

  アクセス許可セットを作成します。「AWS IAM アイデンティティセンター ユーザーガイド」の「アクセス許可セットを作成する」の手順に従ってください。
- IAM 内で、ID プロバイダーによって管理されているユーザー:

  ID フェデレーションのロールを作成します。詳細については IAM ユーザーガイド の サードパーティー ID プロバイダー (フェデレーション) 用のロールを作成する を参照してください。
- IAM ユーザー:
  - ユーザーが担当できるロールを作成します。手順については IAM ユーザーガイド の IAM ユーザーのロールの作成 を参照してください。
  - (お奨めできない方法) ポリシーをユーザーに直接アタッチするか、ユーザーをユーザーグループに追加します。「IAM ユーザーガイド」の「ユーザー (コンソール) へのアクセス許可の追加」の指示に従います。

Amazon EBS スナップショットをアーカイブ

アカウントで所有している `completed` の状態にあるあらゆるスナップショットをアーカイブすることができます。`pending` または `error` の状態、または共有されているスナップショットにあるスナップショットはアーカイブできません。詳細については、「Amazon EBS スナップショットをアーカイブする際の考慮事項と制限」を参照してください。

スナップショットが 1 つ以上の AMI に関連付けられている場合は、スナップショットをアーカイブする前に、関連付けられた AMI を無効にする必要があります。詳細については、「AMI の無効化」を参照してください。

アーカイブされたスナップショットは、スナップショット ID、暗号化ステータス、 AWS Identity and Access Management (IAM) アクセス許可、所有者情報、リソースタグを保持します。ただし、スナップショットをアーカイブすると、高速スナップショット復元とスナップショット共有は自動的に無効になります。

アーカイブの処理中も、スナップショットを引き続き使用できます。スナップショットの階層化ステータスが `archival-complete` の状態の場合、スナップショットは使用できません。

Console

スナップショットをアーカイブするには

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. ナビゲーションペインで、[Snapshots] を選択します。
3. スナップショットのリストで、アーカイブするスナップショットを選択し、[Actions] (アクション)、[Archive snapshot] (スナップショットのアーカイブ) の順にクリックします。
4. 確定するには、[Archive snapshot] (スナップショットのアーカイブ) を選択します。

AWS CLI

スナップショットをアーカイブするには

modify-snapshot-tier AWS CLI コマンドを使用します。`--snapshot-id` に、アーカイブするスナップショットの ID を指定します。`--storage-tier` の場合、`archive` を指定します。

```
aws ec2 modify-snapshot-tier \
    --snapshot-id snap-0abcdef1234567890 \
    --storage-tier archive
```

以下は出力の例です。`TieringStartTime` レスポンスパラメータは、アーカイブプロセスが開始された日付と時刻を、UTC 時間形式 (YYYY-MM-DDTHH:MM:SSZ) で示します。

```
{
    "SnapshotId": "snap-0abcdef1234567890",
    "TieringStartTime": "2021-09-15T16:44:37.574Z"
}
```

PowerShell

スナップショットをアーカイブするには

Edit-EC2SnapshotTier コマンドレットを使用します。次の の例では、指定したスナップショットをアーカイブします。

```
Edit-EC2SnapshotTier `
    -SnapshotId snap-0abcdef1234567890 `
```

```
    -StorageTier "archive"
```

アーカイブされた Amazon EBS スナップショットを復元

アーカイブされたスナップショットを使用するには、まずそのスナップショットを標準階層に復元する必要があります。復元されたスナップショットには、アーカイブされる前と同じスナップショット ID、暗号化ステータス、IAM 許可、所有者情報、リソースタグがあります。復元後は、アカウント内の他のスナップショットと同じ方法で使用できます。復元されたスナップショットは、常に完全なスナップショットです。

スナップショットを復元するときに、そのスナップショットを永久にまたは一時的に復元することを選択できます。

スナップショットを永続的に復元すると、スナップショットはアーカイブ階層から標準階層に永続的に移動されます。スナップショットは、手動で再アーカイブするか、手動で削除するまで、復元され、使用可能な状態になります。スナップショットを永続的に復元すると、スナップショットはアーカイブ階層から削除されます。

スナップショットを一時的に復元すると、指定した復元期間中、スナップショットがアーカイブ階層から標準階層にコピーされます。スナップショットは復元されたままで、復元期間のみ使用できる状態になります。復元期間中、スナップショットのコピーはアーカイブ階層に残ります。期間が終了すると、スナップショットは標準階層から自動的に削除されます。復元期間中は、いつでも復元期間を増減したり、復元タイプを永続的に変更したりすることができます。詳細については、「一時的に復元された Amazon EBS スナップショットの復元期間を変更」を参照してください。

無効になっている AMI に関連付けられているスナップショットを復元してその AMI を使用する場合、まず関連付けられたすべてのスナップショットを完全に復元し、その後「無効化された AMI を再度有効にする」必要があります。関連付けられたスナップショットが一時的に復元された場合、AMI を有効にすることはできません。次のコマンドを使用して AMI に関連付けられているすべてのスナップショットを検索できます。

```
aws ec2 describe-images --image-id ami_id \
  --query Images[*].BlockDeviceMappings[*].Ebs[].SnapshotId[]
```

Console

アーカイブからスナップショットから復元するには

Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。

1. ナビゲーションペインで、[Snapshots] を選択します。
2. スナップショットのリストで、復元するアーカイブされたスナップショットを選択し、[Action] (アクション)、[Restore snapshot from archive] (アーカイブからスナップショットを復元する) の順にクリックします。
3. 実行する復元のタイプを指定します。[Restore type] (復元タイプ) で、以下のいずれかを実行します。
   - スナップショットを永続的に復元するには、[Permanent] (永続) を選択します。
   - スナップショットを一時的に復元するには、[Temporary] (一時的) 、[Temporary restore period] (一時復元期間) の順にクリックし、スナップショットを復元する日数を入力します。
4. 確定するには、[Restore snapshot] (スナップショットの復元) を選択します。

AWS CLI

アーカイブされたスナップショットを永続的に復元するには

`--permanent-restore` オプションを指定して restore-snapshot-tier AWS CLI コマンドを使用します。`--snapshot-id` に、復元するスナップショットの ID を指定します。

```
aws ec2 restore-snapshot-tier \
    --snapshot-id snap-0abcdef1234567890 \
    --permanent-restore
```

以下は出力の例です。

```
{
    "SnapshotId": "snap-0abcdef1234567890",
    "IsPermanentRestore": true
}
```

アーカイブされたスナップショットを一時的に復元するには

restore-snapshot-tier AWS CLI コマンドを使用します。`--permanent-restore` オプションを省略します。`--temporary-restore-days` に、スナップショットを復元する日数を指定します。許容範囲は 1～180 日です。値を指定していない場合、デフォルトで 1 日が適用されます。

次の例では、指定されたスナップショットを復元期間 5 日間で一時的に復元します。

```
aws ec2 restore-snapshot-tier \
    --snapshot-id snap-0abcdef1234567890 \
    --temporary-restore-days 5
```

以下は出力の例です。

```
{
    "SnapshotId": "snap-0abcdef1234567890",
    "RestoreDuration": 5,
    "IsPermanentRestore": false
}
```

PowerShell

アーカイブされたスナップショットを永続的に復元するには

Restore-EC2SnapshotTier コマンドレットを使用します。

```
Restore-EC2SnapshotTier `
    -SnapshotId snap-0abcdef1234567890 `
    -PermanentRestore $true
```

アーカイブされたスナップショットを一時的に復元するには

Restore-EC2SnapshotTier コマンドレットを使用します。

```
Restore-EC2SnapshotTier `
    -SnapshotId snap-0abcdef1234567890 `
    -TemporaryRestoreDays 5
```

一時的に復元された Amazon EBS スナップショットの復元期間を変更

スナップショットを一時的に復元する場合は、スナップショットをアカウントに復元する日数を指定する必要があります。復元期間が過ぎると、スナップショットは標準階層から自動的に削除されます。

一時的に復元されたスナップショットの復元期間は、いつでも変更できます。

復元期間を増減するか、復元タイプを一時的から永続的に変更するかを選択できます。

復元期間を変更すると、新しい復元期間は現在の日付から有効になります。たとえば、新しい復元期間を 5 日間と指定すると、スナップショットは現在の日付から 5 日間復元されます。

> **Note**
>
> 復元期間を 1 日に設定することで、一時的な復元を早期に終了できます。

復元タイプを一時から永続に変更すると、スナップショットコピーはアーカイブ階層から削除され、手動で再アーカイブまたは削除するまで、スナップショットはアカウントで引き続き使用できます。

Console

復元期間または復元タイプを変更するには

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. ナビゲーションペインで、[Snapshots] を選択します。
3. スナップショットのリストで、以前に一時的に復元したスナップショットを選択し、[Actions] (アクション)、[Restore snapshot from archive] (アーカイブからスナップショットを復元する) の順にクリックします。
4. [Restore type] (復元タイプ) で、以下のいずれかを実行します。
   - 復元タイプを一時的から永続的に変更するには、[Permanent] (永続) を選択します。
   - 復元期間を増減するには、[Temporary] (一時的) を保持し、次に[Temporary restore period] (一時復元期間) で、新しい復元期間を日単位で入力します。
5. 確定するには、[Restore snapshot] (スナップショットの復元) を選択します。

AWS CLI

復元期間または復元タイプを変更するには

restore-snapshot-tier コマンドを使用します。`--snapshot-id` に、以前に一時的に復元したスナップショットの ID を指定します。復元タイプを一時的から永続的に変更するには、`--permanent-restore` を指定し、`--temporary-restore-days` は省略します。復元期間を増減するには、`--permanent-restore` を省略します。そして `--temporary-restore-days` で、新しい復元期間を日単位で指定します。

例: 復元期間を増減する

次のコマンドでは、指定したスナップショットの復元期間を 10 日間に変更します。

```
aws ec2 restore-snapshot-tier \
    --snapshot-id snap-0abcdef1234567890 \
    --temporary-restore-days 10
```

以下は出力の例です。

```
{
    "SnapshotId": "snap-0abcdef1234567890",
    "RestoreDuration": 10,
    "IsPermanentRestore": false
}
```

例: 復元タイプを永続に変更する

次のコマンドでは、指定したスナップショットの復元タイプを一時的から永続的に変更します。

```
aws ec2 restore-snapshot-tier \
    --snapshot-id snap-0abcdef1234567890 \
    --permanent-restore
```

以下は出力の例です。

```
{
    "SnapshotId": "snap-0abcdef1234567890",
    "IsPermanentRestore": true
}
```

PowerShell

復元期間または復元タイプを変更するには

Restore-EC2SnapshotTier コマンドレットを使用します。-SnapshotId に、以前に一時的に復元したスナップショットの ID を指定します。復元タイプを一時的から永続的に変更するには、-PermanentRestore を指定し、-TemporaryRestoreDays は省略します。復元期間を増減するには、-PermanentRestore を省略します。そして -TemporaryRestoreDays で、新しい復元期間を日単位で指定します。

例: 復元期間を増減する

次のコマンドでは、指定したスナップショットの復元期間を 10 日間に変更します。

```
Restore-EC2SnapshotTier `
    -SnapshotId snap-0abcdef1234567890 `
    -TemporaryRestoreDays 10
```

例: 復元タイプを永続に変更する

次のコマンドでは、指定したスナップショットの復元タイプを一時的から永続的に変更します。

```
Restore-EC2SnapshotTier `
    -SnapshotId snap-0abcdef1234567890 `
    -PermanentRestore $true
```

Amazon EBS スナップショットアーカイブを表示

Console

スナップショットのストレージ階層情報を表示するには

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. ナビゲーションペインで、[Snapshots] を選択します。
3. スナップショットのリストで、スナップショットを選択し、[Storage tier] (ストレージ階層) タブを選択します。

   タブは以下の情報を提供します。

   - [Last tier change started on] (最後の階層変更の開始日時) — 前回のアーカイブまたは復元が開始された日時。
   - [Tier change progress] (階層変更の進行状況) — 前回のアーカイブまたは復元アクションの進行状況 (パーセンテージ)。
   - [Storage tier] (ストレージ階層) — スナップショットのストレージ階層。アーカイブされたスナップショットは常に `archive`、一時的に復元されたスナップショットを含め、標準階層に保存されたスナップショットは `standard` となります。
   - [Tiering status] (階層化ステータス) — 前回のアーカイブまたは復元アクションのステータス。
   - [Archive completed on] (アーカイブの完了日時) — アーカイブが完了した日時。

- [Temporary restore expires on] (一時的復元の有効期限) — 設定された一時的に復元されたスナップショットが期限切れになる日時

AWS CLI

アーカイブされたスナップショットに関するアーカイブ情報を表示するには

describe-snapshot-tier-status コマンドを使用します。snapshot-id フィルターを指定し、フィルター値にスナップショット ID を指定します。または、アーカイブされたすべてのスナップショットを表示するには、フィルターを省略します。

```
--filters "Name=snapshot-id, Values=snapshot_id"
```

出力には、以下のレスポンスパラメータが含まれます。

- Status— スナップショットのステータス。アーカイブされたスナップショットはすべて completed になります。completed の状態にあるスナップショットのみアーカイブすることができます。
- LastTieringStartTime — アーカイブプロセスが開始した日と時刻のUTC 時間形式 (YYYY-MM-DDTHH:MM:SSZ)。
- LastTieringOperationState — アーカイブプロセスの現在の状態。状態には、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 が含まれます。
- LastTieringProgress — スナップショットアーカイブプロセスの進行状況 (パーセント)。
- StorageTier — スナップショットのストレージ階層。アーカイブされたスナップショットは常に archive、一時的に復元されたスナップショットを含め、標準階層に保存されたスナップショットは standard となります。
- ArchivalCompleteTime — アーカイブプロセスが完了した日と終了時刻の UTC 時間形式 (YYYY-MM-DDTHH:MM:SSZ)。

例: スナップショットを説明する

次の例では、指定したスナップショットの情報が表示されます。

```
aws ec2 describe-snapshot-tier-status \
```

```
    --filters "Name=snapshot-id, Values=snap-0abcdef1234567890"
```

以下は出力の例です。

```
{
    "SnapshotTierStatuses": [
        {
            "Status": "completed",
            "ArchivalCompleteTime": "2021-09-15T17:33:16.147Z",
            "LastTieringProgress": 100,
            "Tags": [],
            "VolumeId": "vol-01234567890abcdef",
            "LastTieringOperationState": "archival-completed",
            "StorageTier": "archive",
            "OwnerId": "123456789012",
            "SnapshotId": "snap-0abcdef1234567890",
            "LastTieringStartTime": "2021-09-15T16:44:37.574Z"
        }
    ]
}
```

アーカイブ階層と標準階層のスナップショットを表示するには

describe-snapshots コマンドを使用します。--snapshot-ids に、スナップショットの ID を指定します。

```
aws ec2 describe-snapshots --snapshot-ids snap-0abcdef1234567890
```

以下は出力の例です。StorageTier レスポンスパラメータは、スナップショットが現在アーカイブされているかどうかを示します。archive は、スナップショットが現在アーカイブされ、アーカイブ階層に格納されていることを示します。standard は、スナップショットが現在アーカイブされておらず、標準階層に格納されていることを示します。

次の出力例では、Snap A がアーカイブされており、Snap B そして Snap C はアーカイブされていません。

また、RestoreExpiryTime レスポンスパラメータは、アーカイブから一時的に復元されるスナップショットに対してのみ返されます。これは、一時的に復元されたスナップショットが標準階層から自動的に削除される時期を示します。永続的に復元されるスナップショットに対しては返されません。

次の出力例では、Snap C が一時的に復元され、2021-09-19T21:00:00.000Z (2021 年 9 月 19 日、21:00 UTC) に標準階層から自動的に削除されます。


```
{
    "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": []
        }
    ]
}
```

アーカイブ階層または標準階層に格納されているスナップショットのみを表示するには

describe-snapshots コマンドを使用します。--filter オプションをフィルター名 storage-tier と共に含めます。フィルター値に archive または standard を指定します。次の例では、アーカイブされたスナップショットのみを表示します。

```
aws ec2 describe-snapshots --filters "Name=storage-tier,Values=archive"
```

PowerShell

アーカイブされたスナップショットに関するアーカイブ情報を表示するには

Get-EC2SnapshotTierStatus コマンドレットを使用します。

```
Get-EC2SnapshotTierStatus `
    -Filter @{Name="snapshot-id"; Values="snap-0abcdef1234567890"}
```

アーカイブ階層と標準階層のスナップショットを表示するには

Get-EC2Snapshot コマンドレットを使用します。

```
Get-EC2Snapshot `
    -SnapshotId snap-0abcdef1234567890
```

アーカイブ階層または標準階層に格納されているスナップショットのみを表示するには

Get-EC2Snapshot コマンドレットを使用します。次の例では、アーカイブされたスナップショットのみを表示します。

```
Get-EC2Snapshot `
    -Filter @{Name="storage-tier"; Values="archive"}
```

CloudWatch Events を使用して Amazon EBS スナップショットアーカイブをモニタリング

Amazon EBS は、スナップショットアーカイブアクションに関連するイベントを発行します。 AWS Lambda および Amazon CloudWatch Events を使用して、プログラムでイベント通知を処理できます。イベントは、ベストエフォートベースで出力されます。詳細については、Amazon EventBridge ユーザーガイドを参照してください。

利用できるイベントは次のとおりです。

- `archiveSnapshot` — スナップショットアーカイブアクションが成功または失敗したときに発行されます。

  スナップショットアーカイブアクションが成功したときに発生するイベントの例を次に示します。

```
{
   "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"
  }
```

  スナップショットアーカイブアクションが失敗した場合に発生するイベントの例を次に示します。

```
{
   "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 — 永続復元アクションが成功または失敗したときに出力されます。

  永続的な復元アクションが成功したときに発生するイベントの例を次に示します。

```
{
   "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"
  }
}
```

永続的な復元アクションが失敗した場合に発生するイベントの例を次に示します。

```
{
   "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 — 一時復元アクションが成功または失敗したときに出力されます。

  一時的な復元アクションが成功したときに発生するイベントの例を次に示します。

```
{
   "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"
  }
}
```

一時的な復元アクションが失敗した場合に発生するイベントの例を次に示します。

```
{
   "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` — 一時的に復元されたスナップショットの復元期間の有効期限が切れたときに発行されます。

以下に例を示します。

```
{
   "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"
  }
}
```

Amazon EBS スナップショットの削除

ボリュームの Amazon EBS スナップショットが不要になった場合は、これを削除できます。スナップショットを削除しても、ボリュームには影響しません。ボリュームを削除しても、そのボリュームが作成したスナップショットには影響しません。

トピック

- スナップショットの削除に関する考慮事項
- 増分スナップショットの削除の仕組み
- スナップショットを削除する
- マルチボリュームスナップショットを削除

スナップショットの削除に関する考慮事項

スナップショットの削除には、次の考慮事項が適用されます。

- 登録済みの AMI によって使用されている EBS ボリュームのルートデバイスのスナップショットを削除することはできません。この考慮事項は、登録されている AMI が非推奨または無効になっている場合でも適用されます。スナップショットを削除するには、まず AMI の登録を解除する必要があります。詳細については、「AMI の登録の解除」を参照してください。
- Amazon EC2 を使用して AWS Backup サービスによって管理されているスナップショットを削除することはできません。代わりに、 AWS Backup を使用して、バックアップボールト内の対応する復旧ポイントを削除します。詳細については、「AWS Backup デベロッパーガイド」の「バックアップの削除」を参照してください。
- スナップショットを手動で作成、保持、削除することも、Amazon Data Lifecycle Manager を使用してスナップショットを管理することもできます。詳細については、Amazon Data Lifecycle Managerを参照してください。
- 進行中のスナップショットを削除することはできますが、スナップショットが完了しなければ削除は実行されません。これには時間がかかる場合があります。同時スナップショットの制限に達していて、追加のスナップショットを取得しようとすると、`ConcurrentSnapshotLimitExceeded` エラーが発生することがあります。詳細については、Amazon Web Services 全般のリファレンスで Amazon EBS の「Service Quotas」を参照してください。
- ごみ箱の保持ルールに一致するスナップショットを削除すると、スナップショットはすぐに削除されるのではなく、ごみ箱に保持されます。詳細については、「ごみ箱」を参照してください。
- 無効化された EBS-backed の AMI に関連付けられているスナップショットは削除できません。詳細については、「AMI の無効化」を参照してください。
- 共有されているスナップショットは削除できません。
- 所有している共有スナップショットを削除すると、そのスナップショットが共有されているすべてのアカウントはそのスナップショットにアクセスできなくなります。

増分スナップショットの削除の仕組み

ボリュームのスナップショットを定期的に作成する場合、スナップショットは増分になります。最新のスナップショットを作成した時点から、デバイス上で変更があったブロックだけが、新しいスナップショットに保存されます。スナップショットの保存は増分ベースで行われるものの、最新のスナップショットさえあればボリュームを作成できるようにスナップショット削除プロセスは設計されています。

データが以前のスナップショットまたは一連のスナップショットに保持されているボリュームに存在し、後でボリュームから削除された場合、データは以前のスナップショットの一意のデータとみなされます。一意のデータは、一意のデータを参照するすべてのスナップショットが削除されない限り、一連のスナップショットから削除されません。

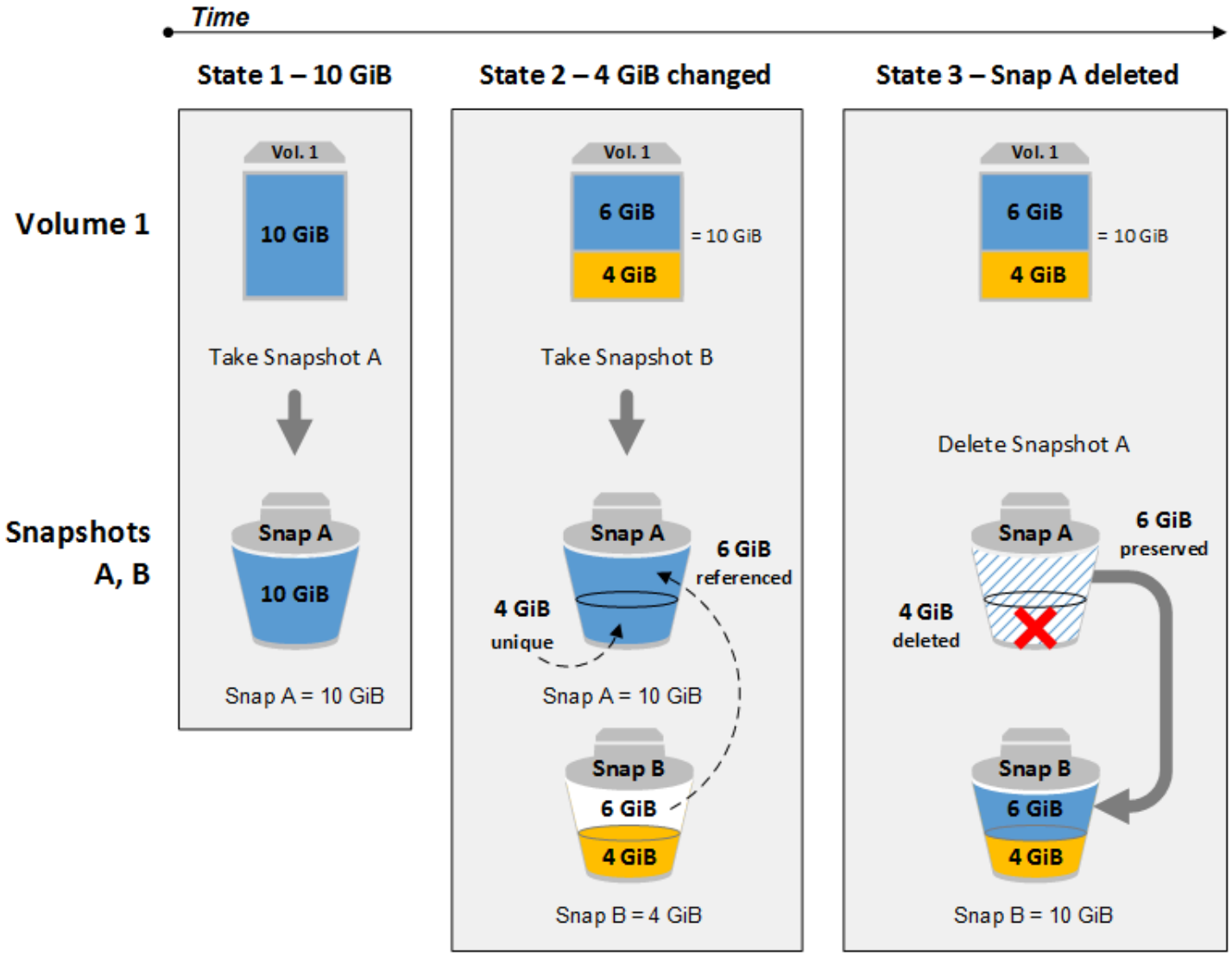
スナップショットを削除すると、そのスナップショットのみが参照するデータのみが削除されます。一意のデータは、それを参照するすべてのスナップショットが削除されるまで削除されません。ボリュームの過去のスナップショットを削除しても、そのボリュームのそれ以降のスナップショットからボリュームを作成する機能に影響することはありません。

スナップショットを削除しても、組織のデータストレージコストが減少しない場合があります。他のスナップショットはそのスナップショットのデータを参照する場合があります。参照されたデータは常に保持されます。後から作成したスナップショットが使用しているデータを含むスナップショットを削除すると、参照されるデータに関連付けられたコストは後から作成されたスナップショットに割り当てられます。スナップショットにデータを保存する方法の詳細については、Amazon EBS スナップショットの仕組みと次の例を参照してください。

次の図では、ボリューム 1 は 3 つの時点に示されています。スナップショットが最初の 2 つの状態をキャプチャし、3 つめでは、スナップショットが削除されています。

- 状態 1 では、ボリュームに 10 GiB のデータがあります。スナップ A がボリュームで作成された最初のスナップショットであるため、10 GiB のデータ全体をコピーする必要があるます。この状態では、10 GiB のスナップショットデータの保存に対して課金されます。
- 状態 2 では、ボリュームにはまだ 10 GiB のデータが含まれていますが、そのうち 4 GiB が変更されました。スナップ B は、スナップ A が取得された後に変更された 4 GiB のみを保存し、スナップ A に既に保存されている 6 GiB の変更されていないデータを参照します。この状態では、14 GiB のスナップショットデータ (スナップ A から 10 Gib + スナップ B から 4 GiB) の保存に対して課金されます。
- 状態 3 では、ボリュームは変更されませんが、スナップ A は削除されます。スナップ A に含まれる 6 GiB の変更されていないデータはスナップ B によって引き続き参照されるため、そのデータは保持され、スナップ B に関連付けられます。スナップ A の 4 GiB の一意のデータは、他のスナップショットによって参照されなくなったため削除されます。この状態では、10 GiB のスナップショットデータ (スナップ A から保持される 6 GiB のデータ + スナップ B にある 4 GiB のデータ) の保存に対して課金されます。

別のスナップショットによって一部のデータが参照されているスナップショットの削除



スナップショットを削除する

Console

スナップショットを削除するには

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. ナビゲーションペインで、[スナップショット] を選択します。
3. スナップショットを選択します。
4. [Actions] (アクション)、[Delete snapshot] (スナップショットの削除) の順に選択します。
5. 確認を求められたら、**delete**と入力し、[削除] を選択します。

AWS CLI

スナップショットを削除するには

delete-snapshot コマンドを使用します。

```
aws ec2 delete-snapshot --snapshot-id snap-0abcdef1234567890
```

PowerShell

スナップショットを削除するには

Remove-EC2Snapshot コマンドレットを使用します。

```
Remove-EC2Snapshot -SnapshotId snap-0abcdef1234567890
```

> **トラブルシューティングのヒント**
>
> スナップショットが現在 AMI によって使用されていることを示す `Failed to delete snapshot` エラーが表示された場合、スナップショットを削除する前に関連付けられた AMI の登録を解除する必要があります。AMI に関連付けられているスナップショットを削除することはできません。
> コンソールを使用していて、関連する AMI が無効になっている場合、無効になっている AMI を表示するには、[AMI] 画面の [無効化されたイメージ] フィルタを選択する必要があります。

マルチボリュームスナップショットを削除

マルチボリュームスナップショットを削除するには、スナップショットを作成したときにセットに適用したタグを使用して、マルチボリュームスナップショットセットのすべてのスナップショットを取得します。次に、スナップショットを個別に削除します。

マルチボリュームスナップショットセット内の個々のスナップショットの削除を妨げられることはありません。`pending state` の中にあるスナップショットを削除すると、そのスナップショットだけが削除されます。マルチボリュームスナップショットセット内の他のスナップショットは正常に完了します。

Amazon EBS 高速スナップショット復元

Amazon EBS 高速スナップショット復元 (FSR) を使用するとスナップショットからボリュームを作成でき、このボリュームは作成時に完全に初期化された状態になります。これにより、ブロックの初回アクセス時における I/O オペレーションのレイテンシーがなくなります。高速スナップショット復元を使用して作成されたすべてのボリュームでは、プロビジョンドパフォーマンスがすばやく実現されます。

開始するには、特定のアベイラビリティーゾーンで特定のスナップショットの高速スナップショット復元を有効にします。スナップショットとアベイラビリティーゾーンのペアごとに 1 つの高速スナップショット復元を参照します。高速スナップショット復元が有効になっているアベイラビリティーゾーンの 1 つで、対応するスナップショットからボリュームを作成すると、ボリュームは高速スナップショット復元を使用して復元されます。

スナップショットごとに高速スナップショット復元を明示的に有効にする必要があります。例えば、高速スナップショット復元が有効なスナップショットから復元されたボリュームを基に、新たにスナップショットを作成した場合、作成されたスナップショットの高速スナップショット復元は、自動的に有効化されません。高速スナップショット復元が有効化されているスナップショットをコピーすると、そのスナップショットコピーの高速スナップショット復元は自動で有効化されまん。

高速スナップショット復元のパフォーマンスを十分に活用して復元できるボリュームの数は、スナップショットのボリューム作成クレジットの数によって決まります。詳細については、「Amazon EBS 高速スナップショット復元ボリューム作成クレジット」を参照してください。

所有しているスナップショットに対しても、共有しているパブリックおよびプライベートスナップショットに対しても、高速スナップショット復元を有効にすることができます。

内容

- 考慮事項
- 料金と請求
- Amazon EBS 高速スナップショット復元ボリューム作成クレジット
- Amazon EBS スナップショットの高速スナップショット復元を設定
- Amazon EBS スナップショットの高速スナップショット復元の状態を確認
- 高速スナップショット復元を使用して復元した Amazon EBS ボリュームの表示

考慮事項

- 高速スナップショット復元は AWS Outposts、、ローカルゾーン、および Wavelength Zones ではサポートされていません。
- サイズが 16 TiB 以下のスナップショットで、スナップショットの高速復元を有効にできます。
- 最大 64,000 IOPS および 1,000 MiB/ 秒のスループットでプロビジョニングされたボリュームは、高速スナップショット復元の完全なパフォーマンス上の利点を享受します。64,000 IOPS または 1,000 MiB / 秒のスループットを超えるパフォーマンスでプロビジョニングされたボリュームの場合、完全なパフォーマンスを得るには、ボリュームを初期化する ことをお勧めします。
- 高速スナップショット復元は、リージョンあたり最大 5 個のスナップショットに対して有効にすることができます。このクォータは、所有しているスナップショットおよび共有しているスナップショットに適用されます。共有しているスナップショットに対して高速スナップショット復元を有効にすると、お客様の高速スナップショット復元のクォータにカウントされます。スナップショット所有者の高速スナップショット復元のクォータにはカウントされません。
- スナップショットに対する高速スナップショット復元の状態が変わると、Amazon EBS によって Amazon CloudWatch Events が発行されます。詳細については、EBS 高速スナップショット復元イベントを参照してください。

料金と請求

特定のアベイラビリティーゾーンでスナップショットの高速スナップショット復元を有効にしている時間中は、請求が発生します。料金は 1 時間を最小として時間単位で計算されます。

例えば、US-East-1a で 1 か月 (30 日間) にわたって 1 つのスナップショットのスナップショット高速復元を有効にすると、料金は 540 USD になります (1 スナップショット x 1 AZ x 720 時間 x 1 時間あたり $0.75)。us-east-1a、us-east-1b、us-east-1c で同じ期間にわたって 2 つのスナップショットの高速スナップショット復元を有効にすると、料金は 3240 USD になります (2 スナップショット x 3 AZ x 720 時間 x 1 時間あたり $0.75)。

共有されているパブリックまたはプライベートスナップショットに対して高速スナップショット復元を有効にすると、お客様のアカウントに請求されます。スナップショット所有者には請求されません。共有しているスナップショットがスナップショット所有者によって削除または共有解除されると、高速スナップショット復元がお客様のアカウントのスナップショットに対して無効になり、請求が停止されます。

詳細については、Amazon EBS の料金表を参照してください。

Amazon EBS 高速スナップショット復元ボリューム作成クレジット

高速スナップショット復元のフルパフォーマンスの利点を享受するボリュームの数は、スナップショットのボリューム作成クレジットの数によって決まります。アベイラビリティーゾーンごとにスナップショットあたり 1 つのクレジットバケットがあります。高速スナップショット復元を有効にしてスナップショットから作成するボリュームごとにクレジットバケットの 1 つのクレジットが消費されます。スナップショットから 1 つの初期化ボリュームを作成するには、バケットに少なくとも 1 つのクレジットが必要です。ボリュームの作成時、バケット内のクレジットが 1 つに満たない場合、この作成には、高速スナップショット復元のメリットを活用できません。

共有しているスナップショットに対して高速スナップショット復元を有効にすると、お客様のアカウントの共有スナップショット用に個別のクレジットバケットが割り当てられます。共有スナップショットからボリュームを作成する場合、クレジットはお客様のクレジットバケットから消費されます。スナップショット所有者のクレジットバケットからは消費されません。

クレジットバケットのサイズとリフィルレートは、スナップショットデータのサイズではなく、スナップショットのサイズ (ソースボリュームのサイズでもあります) に基づいています。例えば、150 GiB のデータを持つ 200 GiB ボリュームからスナップショットを作成し、それに対して高速スナップショット復元を有効にすると、クレジットバケットのサイズとリフィルレートは 200 GiB に基づいて決まります。

スナップショットの高速スナップショット復元を有効にすると、クレジットバケットはゼロクレジットから始まり、最大クレジット容量に達するまで設定されたレートで補充されます。また、クレジットを消費すると、クレジットバケットは、最大クレジット容量に達するまで時間の経過に伴って補充されます。

クレジットバケットの補充レートは次のように計算されます。

```
MIN (10, (1024 ÷ snapshot_size_gib))
```

クレジットバケットのサイズは、次のように計算されます。

```
MAX (1, MIN (10, (1024 ÷ snapshot_size_gib)))
```

例えば、スナップショットの高速スナップショット復元をサイズ 128 GiB を指定して有効にすると、補充レートは 0.1333 クレジット/分になります。

```
MIN (10, (1024 ÷ 128))
```

```
 = MIN (10, 8)
 = 8 credits per hour
 = 0.1333 credits per minute
```

クレジットバケットの最大サイズは、8 クレジットです。

```
MAX (1, MIN (10, (1024 ÷ 128)))
 = MAX (1, MIN (10, 8))
 = MAX (1, 8)
 = 8 credits
```

この例では、高速スナップショット復元を有効にすると、クレジットバケットはゼロクレジットで始まります。8 分後、クレジットバケットには初期化されたボリュームを 1 つ作成するのに十分なクレジット (0.1333 credits × 8 minutes = 1.066 credits) があります。クレジットバケットが満杯である場合、同時に 8 個の初期化ボリュームを作成できます (8 クレジット)。バケットが最大容量を下回ると、0.1333 クレジット/秒で補充されます。

CloudWatch メトリクスを使用して、クレジットバケットのサイズおよび各バケットで利用可能なクレジット数をモニタリングすることができます。詳細については、高速スナップショット復元のメトリクスを参照してください。

高速スナップショット復元を有効にしてスナップショットからボリュームを作成したら、describe-volumes を使用してボリュームを示し、高速スナップショット復元を使用して、ボリュームが初期化されたボリュームとして作成されたかどうかを、出力の fastRestored フィールドで確認することができます。

Amazon EBS スナップショットの高速スナップショット復元を設定

デフォルトでは、高速スナップショット復元はスナップショットに対して無効になっています。高速スナップショット復元は、所有しているスナップショットおよび共有しているスナップショットに対して有効または無効にすることができます。スナップショットに対して高速スナップショット復元を有効または無効にすると、その変更はお客様のアカウントにのみ適用されます。

> **Note**
>
> スナップショットに対して高速スナップショット復元を有効にすると、特定のアベイラビリティーゾーンで高速スナップショット復元が有効になっている時間 (分単位) に対してお客様のアカウントに請求されます。最低料金として 1 時間分の料金が請求されます。

所有しているスナップショットを削除すると、お客様のアカウントのそのスナップショットに対して高速スナップショット復元が自動的に無効になります。共有しているスナップショットに対して高速スナップショット復元を有効にし、スナップショット所有者がそのスナップショットを削除または共有解除した場合、お客様のアカウントの共有スナップショットに対して高速スナップショット復元が自動的に無効になります。

共有しているスナップショットに対して高速スナップショット復元を有効にしており、そのスナップショットがカスタム CMK を使用して暗号化されている場合、スナップショット所有者がカスタム CMK に対するアクセス許可を取り消しても、スナップショットに対して高速スナップショット復元は自動的に無効になりません。そのスナップショットに対する高速スナップショット復元は手動で無効にする必要があります。

スナップショットの高速スナップショット復元を有効にすると、スナップショットは最適化中 (optimizing) 状態になります。optimizing の状態にあるスナップショットを使用しながらボリュームを復元することで、パフォーマンスに関する一定のメリットが得られます。高速スナップショット復元によるパフォーマンス上の最大限のメリットは、状態が enabled になったスナップショットでのみ提供されます。

Console

高速スナップショット復元を設定するには

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. ナビゲーションペインで、[Snapshots] を選択します。
3. スナップショットを選択し、[Actions] (アクション)、[Manage fast snapshot restore] (高速スナップショット復元の管理) を選択します。
4. [高速スナップショット復元の設定] セクションには、選択したスナップショットの高速スナップショット復元を有効にできる、すべてのアベイラビリティーゾーンが一覧表示されます。[Current status] (現在のステータス) のボリュームは、各ゾーンで高速スナップショット復元が現在有効か無効かを示します。

   現在無効になっているゾーンで高速スナップショット復元を有効にするには、ゾーンを選択し、次に [Enable] (有効化) を選択し、[Enable] (有効化) を選択して確定します。

   現在高速スナップショット復元が有効になっているゾーンで高速復元を無効にするには、[Disable] (無効) を選択します。
5. 必要な変更を行ったら、[Close] (閉じる) を選択します。

AWS CLI

高速スナップショット復元を有効にするには

enable-fast-snapshot-restores コマンドを使用します。

```
aws ec2 enable-fast-snapshot-restores \
    --availability-zones us-east-1a us-east-1b \
    --source-snapshot-ids snap-0abcdef1234567890
```

高速スナップショット復元を無効にするには

disable-fast-snapshot-restores コマンドを使用します。

```
aws ec2 disable-fast-snapshot-restores \
    --availability-zones us-east-1a \
    --source-snapshot-ids snap-0abcdef1234567890
```

次の例では、describe-fast-snapshot-restores コマンドを使用して、無効な高速スナップショット復元を記述しています。

```
aws ec2 describe-fast-snapshot-restores \
    --filters Name=state,Values=disabled
```

PowerShell

高速スナップショット復元を有効にするには

Enable-EC2FastSnapshotRestore コマンドレットを使用します。

```
Enable-EC2FastSnapshotRestore `
    -AvailabilityZone us-east-1a us-east-1b `
    -SourceSnapshotId snap-0abcdef1234567890
```

高速スナップショット復元を無効にするには

Disable-EC2FastSnapshotRestore コマンドレットを使用します。

```
Disable-EC2FastSnapshotRestore `
    -AvailabilityZone us-east-1a `
```

```
    -SourceSnapshotId snap-0abcdef1234567890
```

次の例では、Get-EC2FastSnapshotRestore コマンドレットを使用して、無効な高速スナップショット復元を記述しています。

```
Get-EC2FastSnapshotRestore `
    -Filter @{Name="state"; Values="disabled"}
```

Amazon EBS スナップショットの高速スナップショット復元の状態を確認

スナップショットの高速スナップショット復元は、以下のいずれかの状態になります。

- enabling — 高速スナップショット復元の有効化がリクエストされました。
- optimizing — 高速スナップショット復元の有効化中です。スナップショットの最適化には TiB あたり 60 分を要します。最適化されたスナップショットにより、ボリュームのリストア時のパフォーマンスに関し、一定のメリットが得られます。
- enabled — 高速スナップショット復元は有効になっています。この状態にあり、かつ十分なボリューム作成クレジットを持つスナップショットは、ボリューム復元時にパフォーマンスの利点をフルに提供します。
- disabling — 高速スナップショット復元の無効化がリクエストされました。または、高速スナップショット復元の有効化のリクエストが失敗しました。
- disabled — 高速スナップショット復元は無効になっています。高速スナップショット復元は必要に応じて再度有効にすることができます。

所有しているスナップショットまたは共有しているスナップショットに対する高速スナップショット復元の状態を表示できます。

Console

高速スナップショット復元の状態を表示するには

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. ナビゲーションペインで、[スナップショット] を選択します。
3. スナップショットを選択します。
4. [Details] (詳細) タブの [Fast snapshot restore] (高速スナップショット復元) では、高速スナップショット復元の状態が表示されます。

AWS CLI

高速スナップショット復元が有効になっているスナップショットを表示するには

describe-fast-snapshot-restores コマンドを使用します。

```
aws ec2 describe-fast-snapshot-restores --filters Name=state,Values=enabled
```

以下は出力の例です。

```
{
    "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"
        }
    ]
}
```

PowerShell

高速スナップショット復元が有効になっているスナップショットを表示するには

Get-EC2FastSnapshotRestore コマンドレットを使用します。

```
Get-EC2FastSnapshotRestore `
    -Filter @{Name="state"; Values="enabled"}
```

高速スナップショット復元を使用して復元した Amazon EBS ボリュームの表示

該当するアベイラビリティーゾーンで高速スナップショット復元が有効になっているスナップショットからボリュームを作成すると、ボリュームは高速スナップショット復元を使用して復元されます。

AWS CLI

高速スナップショット復元が有効になっているスナップショットから作成したボリュームを表示するには

describe-volumes コマンドを使用します。

```
aws ec2 describe-volumes --filters Name=fast-restored,Values=true
```

以下は出力の例です。

```
{
    "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-0abcdef1234567890",
            "State": "available",
            "VolumeId": "vol-01234567890abcdef",
            "Iops": 100,
            "VolumeType": "gp2",
            "FastRestored": true
        }
    ]
}
```

PowerShell

高速スナップショット復元が有効になっているスナップショットから作成したボリュームを表示するには

Get-EC2Volume コマンドレットを使用します。

```
Get-EC2Volume -Filter @{Name="fast-restored"; Values="true"}
```

Amazon EBS スナップショットのロック

Amazon EBS スナップショットをロックして、偶発的または悪意のある削除から保護したり、WORM (Write Once Read Many) 形式で特定の期間保存したりできます。スナップショットがロックされている期間は、付与されている IAM 許可に関係なく、どのユーザーもスナップショットを削除することはできません。ロックされたスナップショットは、他のスナップショットと同じ方法で引き続き使用できます。

> **Note**
>
> スナップショットロックは、SEC 17a-4、CFTC、および FINRA 規制の対象となる環境での使用について、Cohasset Associates によって評価済みです。スナップショットロックとこれらの規制との関係の詳細については、「Cohasset Associates Compliance Assessment」を参照してください。

スナップショットは、コンプライアンスモードとガバナンスモードのいずれかを使用してロックでき、特定の期間または特定の日付までロックできます。詳細については、「ロックモード」および「ロック期間」を参照してください。

料金

スナップショットは、追加料金なしでロックおよびロック解除できます。ロックされたスナップショットについては、Amazon EBS スナップショットの標準ストレージコストをお支払いいただきます。

トピック

- Amazon EBS スナップショットロックの概念
- Amazon EBS スナップショットロックの考慮事項

- Amazon EBS スナップショットロックへのアクセスを制御
- Amazon EBS スナップショットをロック
- Amazon EBS スナップショットのロック解除
- Amazon EBS スナップショットロック設定を更新
- Amazon EBS スナップショットロックのモニタリング

Amazon EBS スナップショットロックの概念

次の内容は、スナップショットロックの使用を開始する際に理解しておくべき重要な概念を示します。

目次

- ロックモード
- ロック期間
- クーリングオフ期間
- ロック状態

ロックモード

次の 2 つのモードのいずれかでスナップショットをロックできます。

ガバナンスモード

スナップショットがロックされた後、適切な IAM 許可を持つユーザーは、スナップショットのロック解除ならびにロックモードおよびロック期間または有効期限の変更をいつでも行うことができます。ガバナンスモードでスナップショットをロックすると、そのスナップショットは直ちにロックされ、クーリングオフ期間はありません。ガバナンスモードでロックされたスナップショットを削除するには、まずスナップショットをロック解除するか、ロックの有効期限が切れるまで待つ必要があります。

ガバナンスモードを使用すると、特定のユーザーだけがスナップショットのロック解除やスナップショットのロック設定の変更を行えるようになるため、組織のデータガバナンス要件を満たすことができます。コンプライアンスモードでスナップショットをロックする前に、ガバナンスモードを使用してロック設定をテストすることもできます。

コンプライアンスモード

コンプライアンスモードでスナップショットをロックする場合、スナップショットをロックした直後に開始されるクーリングオフ期間をオプションで指定できます。クーリングオフ期間中、適切なアクセス許可を持つユーザーは、スナップショットのロック解除、ロックモードの変更、クーリングオフ期間の延長または短縮、およびロック期間または有効期限の延長または短縮を行うことができます。クーリングオフ期間が終了すると、スナップショットのロック解除、ロックモードの変更、またはロック期間もしくは有効期限の短縮はできなくなります。ただし、ロック期間または有効期限の延長については行うことができます。コンプライアンスモードでロックされたスナップショットをクーリングオフ期間が終了した後に削除するには、ロックの有効期限が切れるまで待つ必要があります。

> **Note**
>
> リクエストのクーリングオフ期間を省略すると、クーリングオフ期間を設定していないコンプライアンスモードでスナップショットをロックできます。この場合、ロックは直ちに有効になり、スナップショットのロック解除、ロックモードの変更、またはロック期間もしくは有効期限の短縮はできなくなります。ただし、ロック期間または有効期限の延長については行うことができます。

コンプライアンスモードを使用すると、コンプライアンス上の理由から特定の期間削除すべきではないスナップショットを保護できます。コンプライアンスモードには、次の利点があります。

- スナップショットの WORM (Write Once Read Many) 設定が有効になります。
- スナップショットを偶発的または悪意のある削除から保護する追加の防御レイヤーを提供します。
- 保持期間を適用して特権ユーザーによる早期削除を防止し、組織のデータ保護ポリシーと手順を満たします。

> **Note**
>
> ロックの有効期限が切れる前にコンプライアンスモードでロックされているスナップショットを削除する唯一の方法は、関連付けられた AWS アカウントを閉じることです。

ロック期間

ロック期間とは、スナップショットのロック状態を継続できる期間のことです。ロック期間は、次のいずれかで指定できますが、両方での指定はできません。

日数

ロック期間は、スナップショットのロック状態を継続する日数で指定できます。指定した日数が経過すると、スナップショットは自動的にロック解除されます。期間は、1 日から 36,500 日 (100 年) の範囲で設定できます。

ロックの有効期限

ロック期間は、将来の有効期限日によって決定できます。スナップショットのロック状態は、ロックの有効期限日を過ぎるまで継続します。ロックの有効期限日を過ぎると、スナップショットは自動的にロック解除されます。

クーリングオフ期間

クーリングオフ期間は、コンプライアンスモードでスナップショットをロックするときに指定できるオプションの期間です。クーリングオフ期間中、適切なアクセス許可を持つユーザーは、スナップショットのロック解除、ロックモードの変更、クーリングオフ期間の延長または短縮、およびロック期間の延長または短縮を行うことができます。クーリングオフ期間が終了すると、ユーザーは、付与されているアクセス許可に関係なく、スナップショットのロック解除、ロックモードの変更、クーリングオフ期間の復元、またはロック期間の短縮を行うことができなくなります。

クーリングオフ期間中は、スナップショットを削除できません。

指定した場合、クーリングオフ期間はスナップショットをロックした直後に開始されます。省略した場合、スナップショットは直ちに、クーリングオフ期間を設定していないコンプライアンスモードでロックされます。

クーリングオフ期間は 1~72 時間の範囲で設定できます。クーリングオフ期間を設定していないコンプライアンスモードでスナップショットを直ちにロックする場合は、リクエストのクーリングオフ期間を指定しないでください。

ロック状態

スナップショットロックは、次に示す状態のいずれかになります。

- `compliance-cooloff` – スナップショットはコンプライアンスモードでロックされていますが、まだクーリングオフ期間内です。スナップショットの削除はできませんが、ロックの解除や適切なアクセス許可を持つユーザーによるロック設定の変更は可能です。
- `governance` – スナップショットはガバナンスモードでロックされています。スナップショットの削除はできませんが、ロックの解除や適切なアクセス許可を持つユーザーによるロック設定の変更は可能です。

- compliance – スナップショットがクーリングオフ期間を設定していないコンプライアンスモードでロックされているか、クーリングオフ期間が切れています。スナップショットのロックを解除したり、削除したりすることはできません。ロック期間を延長できるのは、適切なアクセス許可を持つユーザーだけです。
- expired – スナップショットはコンプライアンスモードまたはガバナンスモードでロックされていますが、ロックの有効期限が切れています。スナップショットはロックされていないため、削除できます。

Amazon EBS スナップショットロックの考慮事項

Amazon EBS スナップショットをロックするときは、次の点に注意してください。

- スナップショットは、pending または completed 状態にある場合にのみロックできます。
  - スナップショットが pending 状態にあるときにロックし、一定期間ロックした場合、ロック期間はスナップショットが completed 状態に到達したときにのみ開始されます。スナップショットが pending 状態にあるときは削除できません。
  - スナップショットが pending 状態にあるときにロックし、何らかの理由でスナップショットの作成に失敗すると、ロックはキャンセルされます。
- クーリングオフ期間の終了後に、コンプライアンスモードでロックされているスナップショットのロック期間を延長した場合、別のクーリングオフ期間を指定することはできません。クーリングオフ期間を指定すると、リクエストは失敗します。
- アーカイブされたスナップショットはロックできます。また、ロックされたスナップショットはアーカイブできます。
- AMI に関連付けられているスナップショットはロックできます。
- ロックされたスナップショットが関連付けられている AMI を登録解除できます。
- ロックされたスナップショットの暗号化に使用された KMS キーを削除できます。
- によって作成されたスナップショットをロックしないことをお勧めします AWS Backup。 AWS Backup は、保持期間が終了する前にスナップショットが削除されないようにします。によって管理されるスナップショットのセキュリティレイヤーを追加するには AWS Backup、 AWS Backup ボールトロックを使用することをお勧めします。詳細については、「AWS Backup Vault Lock」を参照してください。
- スナップショットの作成中や AMI の登録中にスナップショットをロックすることはできません。
- AWS Outposts上で、ローカルの Amazon EBS スナップショットをロックすることはできません。

- ロックの有効期限が切れる前にコンプライアンスモードでロックされているスナップショットを削除する唯一の方法は、関連付けられた AWS アカウントを閉じることです。

  スナップショットをロックしている間に AWS アカウントを閉鎖すると、 はスナップショットをそのままにしてアカウントを 90 日間 AWS 停止します。90 日以内にアカウントを再開しない場合、 はロックされていてもスナップショット AWS を削除します。

Amazon EBS スナップショットロックへのアクセスを制御

デフォルトでは、 ユーザーにはスナップショットロックを使用する許可はありません。ユーザーがスナップショットロックを使用するには、特定のリソースと API アクションを使用する許可を付与する IAM ポリシーを作成する必要があります。詳細については、「IAM ユーザーガイド」の「IAM ポリシーの作成」を参照してください。

トピック

- 必要なアクセス許可
- 条件キーでアクセスを制限します。

必要なアクセス許可

スナップショットロックを使用するには、次の許可をユーザーに付与する必要があります。

- `ec2:LockSnapshot` – スナップショットをロックします。
- `ec2:UnlockSnapshot` – スナップショットのロックを解除します。
- `ec2:DescribeLockedSnapshots` – スナップショットロック設定を表示します。

以下は、スナップショットをロックおよびロック解除したり、スナップショットロック設定を表示したりする許可をユーザーに付与する IAM ポリシーの例です。これには、コンソールユーザーの `ec2:DescribeSnapshots` 許可が含まれます。一部の許可が不要な場合は、ポリシーから削除できます。

JSON

```
{
  "Version":"2012-10-17",
```

```
  "Statement": [
    {
      "Sid": "AllowSnapshotLockOperations",
      "Effect": "Allow",
      "Action": [
        "ec2:LockSnapshot",
        "ec2:UnlockSnapshot",
        "ec2:DescribeLockedSnapshots",
        "ec2:DescribeSnapshots"
      ],
      "Resource": [
        "arn:aws:ec2:*::snapshot/*",
        "arn:aws:ec2:*:111122223333:volume/*"
      ]
    }
  ]
}
```

アクセス許可を付与するにはユーザー、グループ、またはロールにアクセス許可を追加します。

- 以下のユーザーとグループ AWS IAM アイデンティティセンター:

  アクセス許可セットを作成します。「AWS IAM アイデンティティセンター ユーザーガイド」の「アクセス許可セットを作成する」の手順に従ってください。
- IAM 内で、ID プロバイダーによって管理されているユーザー:

  ID フェデレーションのロールを作成します。詳細については IAM ユーザーガイド の サードパーティー ID プロバイダー (フェデレーション) 用のロールを作成する を参照してください。
- IAM ユーザー:
  - ユーザーが担当できるロールを作成します。手順については IAM ユーザーガイド の IAM ユーザーのロールの作成 を参照してください。
  - (お奨めできない方法) ポリシーをユーザーに直接アタッチするか、ユーザーをユーザーグループに追加します。「IAM ユーザーガイド」の「ユーザー (コンソール) へのアクセス許可の追加」の指示に従います。

条件キーでアクセスを制限します。

条件キーを使用して、スナップショットをロックする方法を制限できます。

トピック

- ec2:SnapshotLockDuration
- ec2:CoolOffPeriod

ec2:SnapshotLockDuration

`ec2:SnapshotLockDuration` 条件キーを使用すると、スナップショットをロックするときにユーザーが指定できるロック期間を制限できます。

次のポリシー例では、ユーザーが指定できるロック期間を `10~50` 日に制限しています。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "AllowSnapshotLockWithDurationCondition",
      "Effect": "Allow",
      "Action": "ec2:LockSnapshot",
      "Resource": "arn:aws:ec2:*::snapshot/*",
      "Condition": {
        "NumericGreaterThan": {
          "ec2:SnapshotLockDuration": 10
        },
        "NumericLessThan": {
          "ec2:SnapshotLockDuration": 50
        }
      }
    }
  ]
}
```

ec2:CoolOffPeriod

`ec2:CoolOffPeriod` 条件キーを使用すると、ユーザーがクーリングオフ期間を設定していないコンプライアンスモードでスナップショットをロックできないようにすることができます。

以下のポリシー例では、コンプライアンスモードでスナップショットをロックする際、ユーザーがクーリングオフ期間を 48 時間より長く指定することを制限しています。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "AllowSnapshotLockWithCondition",
      "Effect": "Allow",
      "Action": "ec2:LockSnapshot",
      "Resource": "arn:aws:ec2:*::snapshot/*",
      "Condition": {
        "NumericGreaterThan": {
          "ec2:SnapshotTime": 48
        }
      }
    }
  ]
}
```

Amazon EBS スナップショットをロック

pending または completed 状態にあるスナップショットはロックできます。詳細については、「Amazon EBS スナップショットロックの考慮事項」を参照してください。

Console

スナップショットをロックするには

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. ナビゲーションペインで、[Snapshots] を選択します。
3. ロックするスナップショットを選択し、[アクション]、[スナップショット設定]、[スナップショットロックの管理] の順に選択します。
4. [スナップショットをロック] を選択します。

5. [ロックモード] には、[ガバナンスモード] または [コンプライアンスモード] のいずれかを選択します。詳細については、「ロックモード」を参照してください。
6. [ロック期間] では、次のいずれかの操作を行います。

   - スナップショットを特定の期間ロックするには、[スナップショットのロック期間] を選択し、期間を日単位または年単位で入力します。
   - スナップショットを特定の日付と時刻までロックするには、[スナップショットのロック期間終了日] を選択し、有効期限の日付と時刻を選択します。

   詳細については、「ロック期間」を参照してください。
7. (コンプライアンスモードのみ) [クーリングオフ期間] には、スナップショットのロックを解除してロック設定を変更できるクーリングオフ期間を指定します。詳細については、「クーリングオフ期間」を参照してください。
8. (コンプライアンスモードのみ) スナップショットをコンプライアンスモードでロックし、クーリングオフ期間が終了した後はスナップショットのロック解除ができなくなることを確認するには、[確認] を選択します。
9. [ロック設定を保存] を選択します。

AWS CLI

ガバナンスモードでスナップショットをロックするには

lock-snapshot コマンドを使用します。--lock-mode の場合、governance を指定します。スナップショットを特定の期間ロックするには、--lock-duration に期間を日単位で指定します。

```
aws ec2 lock-snapshot \
    --snapshot-id snap-0abcdef1234567890 \
    --lock-mode governance \
    --lock-duration 30
```

スナップショットを特定の日付までロックする場合は、--expiration-date にロックの有効期限が切れる日付と時刻を UTC タイムゾーンで指定します。

```
aws ec2 lock-snapshot \
    --snapshot-id snap-0abcdef1234567890 \
    --lock-mode governance \
```

```
    --expiration-date YYYY-MM-DDThh:mm:ss.sssZ
```

スナップショットをコンプライアンスモードでロックするには

lock-snapshot コマンドを使用します。--lock-mode の場合、compliance を指定します。--cool-off-period には、必要に応じてクーリングオフ期間を時間単位で指定します。スナップショットを特定の期間ロックするには、--lock-duration にスナップショットをロックする日数を指定します。

```
aws ec2 lock-snapshot \
    --snapshot-id snap-0abcdef1234567890 \
    --lock-mode compliance \
    --cool-off-period 24 \
    --lock-duration 30
```

スナップショットを特定の日付までロックする場合は、--expiration-date にロックの有効期限が切れる日付と時刻を UTC タイムゾーンで指定します。

```
aws ec2 lock-snapshot \
    --snapshot-id snap-0abcdef1234567890 \
    --lock-mode compliance \
    --expiration-date YYYY-MM-DDThh:mm:ss.sssZ
```

PowerShell

ガバナンスモードでスナップショットをロックするには

Lock-EC2Snapshot コマンドレットを使用します。必要に応じて、スナップショットロックの期間を日単位で指定できます。

```
Lock-EC2Snapshot `
    -SnapshotId snap-0abcdef1234567890 `
    -LockMode "governance" `
    -LockDuration 30
```

または、UTC タイムゾーンで、特定の日付までスナップショットをロックすることもできます。

```
Lock-EC2Snapshot `
    -SnapshotId snap-0abcdef1234567890 `
    -LockMode "governance" `
    -ExpirationDate YYYY-MM-DDThh:mm:ss.sssZ
```

スナップショットをコンプライアンスモードでロックするには

Lock-EC2Snapshot コマンドレットを使用します。必要に応じて、クーリングオフ期間を時間単位で指定できます。また、スナップショットロックの期間を日単位で指定することもできます。

```
Lock-EC2Snapshot `
    -SnapshotId snap-0abcdef1234567890 `
    -LockMode "compliance" `
    -CoolOffPeriod 24 `
    -LockDuration 30
```

または、UTC タイムゾーンで、特定の日付までスナップショットをロックすることもできます。

```
Lock-EC2Snapshot `
    -SnapshotId snap-0abcdef1234567890 `
    -LockMode "compliance" `
    -ExpirationDate YYYY-MM-DDThh:mm:ss.sssZ
```

Amazon EBS スナップショットのロック解除

スナップショットをロック解除できるのは、スナップショットがガバナンスモードでロックされている場合、またはコンプライアンスモードでロックされていても、まだクーリングオフ期間内である場合のみです。

Console

スナップショットをロック解除するには

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. ナビゲーションペインで、[Snapshots] を選択します。
3. ロックを解除するスナップショットを選択し、[アクション]、[スナップショット設定]、[スナップショットロックの管理] の順に選択します。
4. [スナップショットのロック解除] を選択し、もう一度 [スナップショットのロック解除] を選択して確定します。

AWS CLI

スナップショットをロック解除するには

unlock-snapshot コマンドを使用します。

```
aws ec2 unlock-snapshot --snapshot-id snap-0abcdef1234567890
```

PowerShell

スナップショットをロック解除するには

Unlock-EC2Snapshot コマンドレットを使用します。

```
Unlock-EC2Snapshot -SnapshotId snap-0abcdef1234567890
```

Amazon EBS スナップショットロック設定を更新

許可される更新内容はロック状態によって異なります。

- governance – ロックモードの変更およびロック期間または有効期限日の延長または短縮を行うことができます。
- compliance-cooloff – ロックモードの変更、クーリングオフ期間の延長または短縮、およびロック期間または有効期限日の延長または短縮を行うことができます。
- compliance – ロック期間または有効期限日の延長のみ行うことができます。

Console

スナップショットロック設定を更新するには

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. ナビゲーションペインで、[Snapshots] を選択します。
3. ロック設定を変更するスナップショットを選択し、[アクション]、[スナップショット設定]、[スナップショットロックの管理] の順に選択します。
4. 必要に応じて設定を更新し、[ロック設定を保存] を選択します。

AWS CLI

スナップショットロック設定を更新するには

lock-snapshot コマンドを使用します。スナップショットの ID と変更するオプションを指定します。次の例では、有効期限を変更しています。

```
aws ec2 lock-snapshot \
    --snapshot-id snap-0abcdef1234567890 \
    --lock-mode governance \
    --expiration-date YYYY-MM-DDThh:mm:ss.sssZ
```

PowerShell

スナップショットロック設定を更新するには

Lock-EC2Snapshot コマンドレットを使用します。スナップショットの ID と変更するオプションを指定します。次の例では、有効期限を変更しています。

```
Lock-EC2Snapshot `
    -SnapshoId snap-0abcdef1234567890 `
    -LockMode "governance" `
    -ExpirationDate YYYY-MM-DDThh:mm:ss.sssZ
```

Amazon EBS スナップショットロックのモニタリング

次のツールを使用して、Amazon EBS スナップショットロックに関連するアクションをモニタリングできます。

トピック

- を使用して Amazon EBS スナップショットロックをモニタリングする AWS CloudTrail
- Amazon EventBridge を使用した Amazon EBS スナップショットロックのモニタリング

を使用して Amazon EBS スナップショットロックをモニタリングする AWS CloudTrail

コンソールからの呼び出しや API へのコード呼び出しを含め、スナップショットロックの API コールをイベントとしてモニタリングできます。CloudTrail で収集した情報を使用して、作成されたリクエスト内容、リクエスト作成元の IP アドレス、リクエスト作成者、リクエスト作成日時、およびその他の詳細を確認できます。

詳細については、「AWS CloudTrailを使用して Amazon EC2 API コールをログに記録する」を参照してください。

Amazon EventBridge を使用した Amazon EBS スナップショットロックのモニタリング

Amazon EBS は、スナップショットロックアクションに関連するイベントを発行します。 AWS Lambda および Amazon EventBridge を使用して、プログラムでイベント通知を処理できます。イベントは、ベストエフォートベースで出力されます。詳細については、Amazon EventBridge ユーザーガイドを参照してください。

以下のイベントが発生します。

- ガバナンスモードまたはコンプライアンスモードで正常にロックされたスナップショット。

```
{
  "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"
  }
}
```

- スナップショットが pending 状態にあるときにロックされ、completed 状態に到達しなかった場合のロック失敗イベント。

```
{
```

```
  "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"
  }
}
```

- ロックの有効期限切れ

```
{
  "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
  }
}
```

- コンプライアンスモードでのロック後、クーリングオフ期間が終了。

```
{
  "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"
  }
}
```

Amazon EBS スナップショットのブロックパブリックアクセス

スナップショットがパブリックに共有されないようにするために、スナップショットのブロックパブリックアクセスを有効にします。リージョンでスナップショットのブロックパブリックアクセスを有効にすると、そのリージョンでスナップショットをパブリックに共有しようとする試みは自動的にブロックされます。これにより、スナップショットのセキュリティを強化し、スナップショットデータを不正アクセスや意図しないアクセスから保護することができます。

スナップショットのブロックパブリックアクセスは、次の 2 つのモードのいずれかで有効にできます。

- すべての共有をブロック – スナップショットのパブリック共有をすべてブロックします。アカウント内のユーザーは、新しいパブリック共有をリクエストできません。また、既にパブリックに共有されていたスナップショットはプライベートとして扱われ、一般公開されなくなります。
- 新しい共有をブロック – スナップショットの新しいパブリック共有のみをブロックします。アカウント内のユーザーは、新しいパブリック共有をリクエストできません。ただし、既にパブリックに共有されていたスナップショットは、引き続き一般公開されます。

考慮事項

スナップショットのブロックパブリックアクセスを使用する際には、次の点に注意してください。

- スナップショットのブロックパブリックアクセスを使用しても、プライベートスナップショットの共有は妨げられません。
- [すべての共有をブロック] モードでスナップショットのブロックパブリックアクセスを有効化すると、既にパブリックに共有されているスナップショットのアクセス許可は変更されません。代わりに、これらのスナップショットが一般に公開されたり、パブリックアクセスされたりすることを防ぎます。したがって、これらのスナップショットの属性は、一般公開されていないにもかかわらず、パブリックに共有されていることを示します。

  後でブロックパブリックアクセスを無効にしたり、新しい共有をブロックするようにモードを変更したりすると、これらのスナップショットは再び一般公開されます。
- スナップショットのブロックパブリックアクセスは、リージョンごとに設定が必要となります。この設定は、有効になっているリージョンのすべてのスナップショットに適用されます。スナップショットのパブリック共有を防ぐ必要があるリージョンごとに、スナップショットのブロックパブリックアクセスを有効にする必要があります。
- ブロックパブリックアクセスは、アカウントレベルの設定が必要となります。この設定は、管理者ユーザーを含むアカウントのすべてのユーザーに適用されます。スナップショットのブロックパブリックアクセスを組織レベルで有効にすることはできません。
- ブロックパブリックアクセス設定は、アカウントで直接、または宣言型ポリシーを使用して設定されます。宣言型ポリシーを使用すると、複数のリージョンと複数のアカウントで同時に設定を適用できます。宣言ポリシーが使用されている場合、アカウント内で直接設定を変更することはできません。このトピックでは、アカウント内で設定を直接設定する方法について説明します。宣言ポリシーの使用の詳細については、「AWS Organizations ユーザーガイド」の「宣言ポリシー」を参照してください。
- スナップショットのブロックパブリックアクセスを使用しても、EBS-backed AMI のパブリック共有を防ぐことはできません。スナップショットのブロックパブリックアクセスを有効にしても、

ユーザーは EBS-backed AMI をパブリックに共有できます。EBS-backed AMI がパブリックに共有されている場合、その AMI にアクセスできるユーザーは、関連付けられたスナップショットからボリュームを作成できます。AMI のパブリック共有を防ぐには、AMI のブロックパブリックアクセスを有効にします。

- スナップショットのブロックパブリックアクセスは、 のローカルスナップショットではサポートされていません AWS Outposts。

料金

スナップショットのブロックパブリックアクセスは、追加料金なしで有効にできます。

目次

- Amazon EBS スナップショットのパブリックアクセスをブロックするための IAM 許可
- Amazon EBS スナップショットのブロックパブリックアクセスの設定
- Amazon EBS スナップショットのブロックパブリックアクセス設定を表示
- Amazon EBS スナップショットのブロックパブリックアクセスを無効化
- EventBridge を使用して Amazon EBS スナップショットのブロックパブリックアクセスをモニタリング

Amazon EBS スナップショットのパブリックアクセスをブロックするための IAM 許可

デフォルトでは、ユーザーにはスナップショットのブロックパブリックアクセスを使用する許可はありません。ユーザーがスナップショットのブロックパブリックアクセスを使用するには、特定の API アクションを使用する許可を付与する IAM ポリシーを作成する必要があります。ポリシーを作成したら、ユーザー、グループ、ロールに許可を追加します。

スナップショットのブロックパブリックアクセスを使用するには、次の許可をユーザーに付与する必要があります。

- `ec2:EnableSnapshotBlockPublicAccess` – スナップショットのブロックパブリックアクセスを有効にし、モードを変更します。
- `ec2:DisableSnapshotBlockPublicAccess` – スナップショットのブロックパブリックアクセスを無効にします。
- `ec2:GetSnapshotBlockPublicAccessState` – リージョンのスナップショット設定のブロックパブリックアクセスを表示します。

IAM ポリシーの例を次に示します。一部の許可が不要な場合は、ポリシーから削除できます。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "ec2:EnableSnapshotBlockPublicAccess",
            "ec2:DisableSnapshotBlockPublicAccess",
            "ec2:GetSnapshotBlockPublicAccessState"
        ],
        "Resource": "*"
    }]
}
```

アクセス許可を付与するにはユーザー、グループ、またはロールにアクセス許可を追加します。

- 以下のユーザーとグループ AWS IAM アイデンティティセンター:

  アクセス許可セットを作成します。「AWS IAM アイデンティティセンター ユーザーガイド」の「アクセス許可セットを作成する」の手順に従ってください。
- IAM 内で、ID プロバイダーによって管理されているユーザー:

  ID フェデレーションのロールを作成します。詳細については IAM ユーザーガイド の サードパーティー ID プロバイダー (フェデレーション) 用のロールを作成する を参照してください。
- IAM ユーザー:
  - ユーザーが担当できるロールを作成します。手順については IAM ユーザーガイド の IAM ユーザーのロールの作成 を参照してください。
  - (お奨めできない方法) ポリシーをユーザーに直接アタッチするか、ユーザーをユーザーグループに追加します。「IAM ユーザーガイド」の「ユーザー (コンソール) へのアクセス許可の追加」の指示に従います。

Amazon EBS スナップショットのブロックパブリックアクセスの設定

スナップショットのブロックパブリックアクセスを有効にして、リージョン内でのスナップショットのパブリック共有を防ぎます。この機能を有効にすると、リージョン内でスナップショットをパブリックに共有するリクエストはブロックされます。

Important

[すべての共有をブロック] モードでスナップショットのブロックパブリックアクセスを有効化すると、既にパブリックに共有されているスナップショットのアクセス許可は変更されません。代わりに、これらのスナップショットが一般に公開されたり、パブリックアクセスされたりすることを防ぎます。したがって、これらのスナップショットの属性は、一般公開されていないにもかかわらず、パブリックに共有されていることを示します。
後でブロックパブリックアクセスを無効にしたり、新しい共有をブロックするようにモードを変更したりすると、これらのスナップショットは再び一般公開されます。

Note

この設定は、アカウントレベルで直接、または宣言ポリシーを使用して設定されます。スナップショットのパブリック共有を防止する各 AWS リージョン で設定する必要があります。宣言型ポリシーを使用すると、複数のリージョンと複数のアカウントで同時に設定を適用できます。宣言ポリシーが使用されている場合、アカウント内で直接設定を変更することはできません。このトピックでは、アカウント内で設定を直接設定する方法について説明します。宣言ポリシーの使用の詳細については、「AWS Organizations ユーザーガイド」の「宣言ポリシー」を参照してください。

Console

スナップショットのブロックパブリックアクセスを設定するには

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. ナビゲーションペインで [EC2 ダッシュボード] を選択し、[アカウントの属性] (右側) で [データ保護とセキュリティ] を選択します。
3. [EBS スナップショットのブロックパブリックアクセス] セクションで [管理] を選択します。
4. [パブリックアクセスをブロック] を選択し、次のオプションのいずれかを選択します。

- [パブリックアクセスをすべてブロック] – スナップショットのパブリック共有をすべてブロックします。アカウント内のユーザーは、新しいパブリック共有をリクエストできません。また、既にパブリックに共有されていたスナップショットはプライベートとして扱われ、一般公開されなくなります。
- [新しいパブリック共有をブロック] – スナップショットの新しいパブリック共有のみをブロックします。アカウント内のユーザーは、新しいパブリック共有をリクエストできません。ただし、既にパブリックに共有されていたスナップショットは、引き続き一般公開されます。

5. [更新] を選択します。

AWS CLI

スナップショットのブロックパブリックアクセスを有効にまたは変更するには

enable-snapshot-block-public-access コマンドを使用します。`--state` に、次のいずれかの値を指定します。

- `block-all-sharing` – スナップショットのパブリック共有をすべてブロックします。アカウント内のユーザーは、新しいパブリック共有をリクエストできません。また、既にパブリックに共有されていたスナップショットはプライベートとして扱われ、一般公開されなくなります。
- `block-new-sharing` – スナップショットの新しいパブリック共有のみをブロックします。アカウント内のユーザーは、新しいパブリック共有をリクエストできません。ただし、既にパブリックに共有されていたスナップショットは、引き続き一般公開されます。

特定のリージョンのスナップショットで、ブロックパブリックアクセスを有効化または変更するには

```
aws ec2 enable-snapshot-block-public-access \
    --state block-new-sharing \
    --region us-east-1
```

以下は出力の例です。

```
{
    "State": "block-new-sharing"
```

```
}
```

全リージョンのスナップショットで、ブロックパブリックアクセスを有効化または変更するには

```
echo -e "Region   \t Public Access State" ; \
echo -e "-------------- \t ----------------------" ; \
for region in $(
    aws ec2 describe-regions \
        --region us-east-1 \
        --query "Regions[*].[RegionName]" \
        --output text
    );
    do (output=$(
        aws ec2 enable-snapshot-block-public-access \
            --region $region \
            --state block-new-sharing \
            --output text)
        echo -e "$region \t $output"
    );
done
```

以下は出力の例です。

```
Region           Public Access State
--------------   ----------------------
ap-south-1       block-new-sharing
eu-north-1       block-new-sharing
eu-west-3        block-new-sharing
...
```

PowerShell

スナップショットのブロックパブリックアクセスを有効にまたは変更するには

Enable-EC2SnapshotBlockPublicAccess コマンドを使用します。-State に、次のいずれかの値を指定します。

- block-all-sharing – スナップショットのパブリック共有をすべてブロックします。アカウント内のユーザーは、新しいパブリック共有をリクエストできません。また、既にパブリックに共有されていたスナップショットはプライベートとして扱われ、一般公開されなくなります。

- block-new-sharing – スナップショットの新しいパブリック共有のみをブロックします。アカウント内のユーザーは、新しいパブリック共有をリクエストできません。ただし、既にパブリックに共有されていたスナップショットは、引き続き一般公開されます。

特定のリージョンのスナップショットで、ブロックパブリックアクセスを有効化または変更するには

```
Enable-EC2SnapshotBlockPublicAccess `
    -Region us-east-1 `
    -State block-new-sharing
```

以下は出力の例です。

```
Value
-----
block-new-sharing
```

全リージョンのスナップショットで、ブロックパブリックアクセスを有効化または変更するには

```
(Get-EC2Region -Region us-east-1).RegionName | `
    ForEach-Object {
    [PSCustomObject]@{
        Region            = $_
        PublicAccessState = (
            Enable-EC2SnapshotBlockPublicAccess `
                -Region $_ `
                -State block-new-sharing)
    }
} | Format-Table -AutoSize
```

以下は出力の例です。

```
Region         PublicAccessState
------         -----------------
ap-south-1     block-new-sharing
eu-north-1     block-new-sharing
eu-west-3      block-new-sharing
...
```

Amazon EBS スナップショットのブロックパブリックアクセス設定を表示

ブロックパブリックアクセスは、アカウントのリージョンごとに、次に示すステータスのいずれかになります。

- すべての共有をブロック – スナップショットのパブリック共有はすべてブロックされます。アカウント内のユーザーは、新しいパブリック共有をリクエストできません。また、既にパブリックに共有されていたスナップショットはプライベートとして扱われ、一般公開されません。
- 新しい共有をブロック – スナップショットの新しいパブリック共有のみがブロックされます。アカウント内のユーザーは、新しいパブリック共有をリクエストできません。ただし、既にパブリックに共有されていたスナップショットは、引き続き一般公開されます。
- ブロック解除 – パブリック共有はブロックされません。ユーザーはスナップショットをパブリックに共有できます。

Console

スナップショットのブロックパブリックアクセスの設定を表示するには

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. ナビゲーションペインで [EC2 ダッシュボード] を選択し、[アカウントの属性] (右側) で [データ保護とセキュリティ] を選択します。
3. [EBS スナップショットのブロックパブリックアクセス] セクションには、現在の設定が表示されます。

AWS CLI

スナップショットのブロックパブリックアクセスの設定を表示するには

get-snapshot-block-public-access-state コマンドを使用します。

- 特定のリージョンの場合

```
aws ec2 get-snapshot-block-public-access-state
```

この出力例では、ManagedBy フィールドは設定を構成したエンティティを示し、account は設定がアカウントで直接構成されたことを示します。declarative-policy という値は、

その設定が宣言的ポリシーによって構成されたことを意味します。詳細については、「AWS Organizations IAM ユーザーガイド」の「管理されたポリシー」を参照してください。

```
{
    "State": "unblocked",
    "ManagedBy": "account"
}
```

- 全リージョン

```
echo -e "Region   \t Public Access State" ; \
echo -e "-------------- \t ----------------------" ; \
for region in $(
    aws ec2 describe-regions \
        --region us-east-1 \
        --query "Regions[*].[RegionName]" \
        --output text
    );
    do (output=$(
        aws ec2 get-snapshot-block-public-access-state \
            --region $region \
            --output text)
        echo -e "$region \t $output"
    );
done
```

以下は出力の例です。

```
Region           Public Access State
--------------   ----------------------
ap-south-1       unblocked
eu-north-1       unblocked
eu-west-3        unblocked
```

PowerShell

スナップショットのブロックパブリックアクセスの設定を表示するには

Get-EC2SnapshotBlockPublicAccessState コマンドレットを使用します。

- 特定のリージョンの場合

```
Get-EC2SnapshotBlockPublicAccessState -Region us-east-1
```

以下は出力の例です。

```
Value
-----
block-new-sharing
```

- 全リージョン

```
(Get-EC2Region -Region us-east-1).RegionName | `
    ForEach-Object {
    [PSCustomObject]@{
        Region            = $_
        PublicAccessState = (Get-EC2SnapshotBlockPublicAccessState -Region $_)
    }
} | Format-Table -AutoSize
```

以下は出力の例です。

```
Region           Public Access State
--------------   ----------------------
ap-south-1       unblocked
eu-north-1       unblocked
eu-west-3        unblocked
...
```

Amazon EBS スナップショットのブロックパブリックアクセスを無効化

スナップショットのブロックパブリックアクセスを無効にして、リージョン内でのスナップショットのパブリック共有を許可します。この機能を無効にすると、ユーザーはリージョン内でスナップショットをパブリックに共有できるようになります。

> **Important**
>
> [すべての共有をブロック] モードでスナップショットのブロックパブリックアクセスを有効化すると、既にパブリックに共有されているスナップショットのアクセス許可は変更されません。代わりに、これらのスナップショットが一般に公開されたり、パブリックアクセスさ

れたりすることを防ぎます。したがって、これらのスナップショットの属性は、一般公開されていないにもかかわらず、パブリックに共有されていることを示します。
ブロックパブリックアクセスを無効にすると、これらのスナップショットは再び一般公開されます。

> **Note**
>
> この設定は、アカウントレベルで直接、または宣言ポリシーを使用して設定されます。スナップショットのパブリック共有を許可する各 AWS リージョン で設定する必要があります。宣言型ポリシーを使用すると、複数のリージョンと複数のアカウントで同時に設定を適用できます。宣言ポリシーが使用されている場合、アカウント内で直接設定を変更することはできません。このトピックでは、アカウント内で設定を直接設定する方法について説明します。宣言ポリシーの使用の詳細については、「AWS Organizations ユーザーガイド」の「宣言ポリシー」を参照してください。

Console

スナップショットのブロックパブリックアクセスを無効にするには

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. ナビゲーションペインで [EC2 ダッシュボード] を選択し、[アカウントの属性] (右側) で [データ保護とセキュリティ] を選択します。
3. [EBS スナップショットのブロックパブリックアクセス] セクションで [管理] を選択します。
4. [パブリックアクセスをブロック] をオフにして、[更新] を選択します。

AWS CLI

スナップショットのブロックパブリックアクセスを無効にするには

disable-snapshot-block-public-access コマンドを使用します。

- 特定のリージョンの場合

```
aws ec2 disable-snapshot-block-public-access --region us-east-1
```

以下は出力の例です。

```
{
    "State": "unblocked"
}
```

- 全リージョン

```
echo -e "Region   \t Public Access State" ; \
echo -e "-------------- \t ----------------------" ; \
for region in $(
    aws ec2 describe-regions \
        --region us-east-1 \
        --query "Regions[*].[RegionName]" \
        --output text
    );
    do (output=$(
        aws ec2 disable-snapshot-block-public-access \
            --region $region \
            --output text)
        echo -e "$region \t $output"
    );
done
```

以下は出力の例です。

```
Region           Public Access State
--------------   ----------------------
ap-south-1       unblocked
eu-north-1       unblocked
eu-west-3        unblocked
```

PowerShell

スナップショットのブロックパブリックアクセスを無効にするには

Disable-EC2SnapshotBlockPublicAccess コマンドレットを使用します。

- 特定のリージョンの場合

```
Disable-EC2SnapshotBlockPublicAccess -Region us-east-1
```

以下は出力の例です。

```
Value
-----
unblocked
```

- 全リージョン

```
(Get-EC2Region -Region us-east-1).RegionName | `
    ForEach-Object {
    [PSCustomObject]@{
        Region            = $_
        PublicAccessState = (Disable-EC2SnapshotBlockPublicAccess -Region $_)
    }
} | `
Format-Table -AutoSize
```

以下は出力の例です。

```
Region         PublicAccessState
------         -----------------
ap-south-1     unblocked
eu-north-1     unblocked
eu-west-3      unblocked
...
```

EventBridge を使用して Amazon EBS スナップショットのブロックパブリックアクセスをモニタリング

Amazon EBS は、スナップショットのパブリックアクセスブロックに関連するイベントを発行します。 AWS Lambda と Amazon EventBridge を使用すると、イベント通知をプログラムで処理できます。イベントは、ベストエフォートベースで出力されます。詳細については、Amazon EventBridge ユーザーガイドを参照してください。

以下のイベントが発生します。

- [すべての共有をブロック] モードで、スナップショットのブロックパブリックを有効にする

```
{
```

```
  "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"
  }
}
```

- [新しい共有をブロック] モードで、スナップショットのブロックパブリックアクセスを有効にする

```
{
  "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"
  }
}
```

- スナップショットのブロックパブリックアクセスを無効にする

```
{
  "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"
  }
}
```

Amazon EBS local snapshots on Outposts

Amazon EBS スナップショットは、EBS ボリュームのポイントインタイムコピーです。

デフォルトでは、AWS Outpost 上の EBS ボリュームのスナップショットは、Outpost で使用するリージョンにある Amazon S3 内に保存されます。また、Outposts の Amazon EBS ローカルスナップショットを使用すれば、Outpost 上のボリュームのスナップショットを、その Outpost 自体の Amazon S3 にローカルに保存することもできます。これにより、スナップショットのデータを Outpost とオンプレミスの両方で保持することが可能になります。さらに、 AWS Identity and Access Management (IAM) ポリシーとアクセス許可を使用して、スナップショットデータが を離れないようにデータレジデンシー強制ポリシーを設定できますOutpost。これは、 リージョンによってまだ提供されておらず、データレジデンシー要件がある国または AWS リージョンに居住している場合に特に便利です。

このトピックでは、Outposts の Amazon EBS ローカルスナップショット の使用に関する情報を提供します。Amazon EBS スナップショットおよび AWS リージョンでのスナップショットの操作の詳細については、「」を参照してくださいAmazon EBS スナップショット。

詳細については、「AWS Outposts ファミリー」および「AWS Outposts Family Documentation」を参照してください。

トピック

- よくある質問
- 前提条件
- 考慮事項
- IAM によるアクセスの制御
- ローカルスナップショット を使用する

よくある質問

1. ローカルスナップショット とは何ですか?

   デフォルトでは、Outpost 上のボリュームの Amazon EBS スナップショットは、Outpost で使用するリージョンにある Amazon S3 内に保存されます。Outposts 上の Amazon S3 で Outpost がプロビジョニングされた場合は、スナップショットを Outpost それ自体に、ローカルで保存することを選択できます。ローカルにあるのは増分スナップショットです。つまり、最後にスナップショットを作成した時点から、ボリューム上で変更のあるブロックだけが保存されます。これらのスナップショットからは、それらが置かれているのと同じ Outpost 上で、いつでもボリュームを復元することが可能です。Amazon EBS スナップショットの詳細については、Amazon EBS スナップショットを参照してください。

2. ローカルスナップショット を使用する理由は何ですか?

   スナップショットは、データをバックアップするのに便利な手段です。ローカルスナップショットでは、スナップショットデータはすべて Outpost にローカルに保存されます。このデータは、ご使用の設備の外部に移動することはありません。これは、 リージョンによってまだ提供されておらず、居住要件がある国または AWS リージョンに居住している場合に特に便利です。

   また、ローカルスナップショットを使用することで、帯域幅が制約された環境でリージョンと Outpost の間の通信を行う場合に使用される帯域幅を削減できます。

3. スナップショットのデータレジデンシーを強制的に Outpost にするにはどうすればよいですか?

   AWS Identity and Access Management (IAM) ポリシーを使用して、ローカルスナップショットを操作するときにプリンシパル (AWS アカウント、IAM ユーザー、IAM ロール) が持つアクセス許可を制御し、データレジデンシーを適用できます。プリンシパルがOutpostボリュームとインスタンスからスナップショットを作成し、そのスナップショットを AWS リージョンに保存できないようにするポリシーを作成できます。Outpost からリージョンへのスナップショットとイメージのコピーは、現在サポートされていません。詳細については、「IAM によるアクセスの制御」を参照してください。

4. マルチボリュームの Crash-consistent な ローカルスナップショット はサポートされていますか?

   はい。Outpost 上のインスタンスから、マルチボリュームの Crash-consistent なローカルスナップショットを作成できます。

5. ローカルスナップショット を作成するにはどうしたらよいですか?

   スナップショットは、 AWS Command Line Interface (AWS CLI) または Amazon EC2 コンソールを使用して手動で作成できます。詳細については、ローカルスナップショット を使用する を

参照してください。また、Amazon Data Lifecycle Manager を使用することで、ローカルスナップショット のライフサイクルを自動化することもできます。詳細については、Outpost のスナップショットを自動化するを参照してください。

6. Outpost がリージョンへの接続を失った場合でも、ローカルスナップショットを作成、使用、または削除できますか?

いいえ。スナップショットのヘルスにとって重要なアクセス、承認、ログ記録、モニタリングサービスなどは、リージョンにより提供されるため、Outpost はそのリージョンに接続されている必要があります。接続が失われると、新らしい ローカルスナップショット の作成、既存のローカルスナップショット からのボリュームの作成やインスタンスの起動、あるいは ローカルスナップショット の削除はできなくなります。

7. ローカルスナップショット の削除後に Amazon S3 ストレージ容量が使用可能になるまでどれくらい待つ必要がありますか?

Amazon S3ストレージ容量は、それを参照している ローカルスナップショット とボリュームを削除した後、72 時間以内に使用可能になります。

8. で Amazon S3 容量が不足しないようにするにはどうすればよいですかOutpost?

ストレージ容量の不足を避けるためには、Amazon CloudWatch アラームを使用して Amazon S3 ストレージ容量を監視し、必要がなくなったスナップショットとボリュームを削除することをお勧めします。Amazon Data Lifecycle Manager を使用してローカルスナップショットのライフサイクルを自動化する場合は、スナップショット保持ポリシーが必要以上に長くスナップショットを保持しないようにしてください。

9. Outpost でローカル Amazon S3 の容量が不足するとどうなりますか ?

Outpost でローカル Amazon S3 の容量が不足した場合、Amazon Data Lifecycle Manager は Outpost にローカルスナップショットを正常に作成できません。Amazon Data Lifecycle Manager は Outpost にローカルスナップショットを作成しようとしますが、スナップショットはすぐに `error` の状態になり、最終的に Amazon Data Lifecycle Manager によって削除されます。`SnapshotsCreateFailed` Amazon CloudWatch メトリクスを使用して、スナップショットのライフサイクルポリシーでスナップショット作成の失敗をモニタリングすることをお勧めします。詳細については、CloudWatch を使用して、Data Lifecycle Manager のポリシーをモニタリングを参照してください。

10. ローカルスナップショット とその ローカルスナップショット で保存された AMI はスポットインスタンスおよびスポットフリートで使用できますか?

    いいえ。ローカルスナップショット、もしくは ローカルスナップショット でバックアップされた AMI を使用して、スポットインスタンスまたはスポットフリートを起動することはできません。

11. ローカルスナップショット とその ローカルスナップショット で保存された AMI は Amazon EC2 Auto Scaling で使用できますか?

    はい。スナップショットと同じ Outpost にあるサブネットで Auto Scaling グループを起動する際は、ローカルスナップショットと、ローカルスナップショットで保存された AMI を使用できます。Amazon EC2 Auto Scaling グループのサービスリンクロールには、スナップショットの暗号化用に、KMS キー を使用するためのアクセス権限が必要です。

    ローカルスナップショットまたはローカルスナップショットにバックアップされた AMIs を使用して、 AWS リージョンで Auto Scaling グループを起動することはできません。

前提条件

スナップショットを Outpost に保存するには、Outposts 上で S3 を使用してプロビジョニングされた Outpost が必要です。Outposts 上の S3 の詳細については、「Amazon S3 ユーザーガイド」の「Amazon S3 on Outposts とは」を参照してください。

考慮事項

ローカルスナップショット を使用する際には、次の点に注意してください。

- ローカルスナップショットを使用するには、 が AWS リージョンに接続しているOutpost必要があります。
- スナップショットメタデータは、 に関連付けられた AWS リージョンに保存されますOutpost。これにはスナップショットのデータ自体は含まれません。
- Outpost に保存されたスナップショットは、デフォルトで暗号化されます。非暗号化スナップショットはサポートされていません。ローカルスナップショットOutpostから にボリュームを作成すると、ボリュームはソーススナップショットと同じ KMS キーを使用して暗号化されます。ただし、ローカルスナップショットをコピーするときに暗号化キーを変更できます。
- Outpost からローカルスナップショットを削除したあと、削除されたスナップショットで使用していた Amazon S3 のストレージ容量は、72 時間以内に使用できるようになります。詳細については、「ローカルスナップショット を削除する」を参照してください。

- Outpost からローカルスナップショットをエクスポートすることはできません。
- ローカルスナップショット では、高速スナップショット復元を有効化できません。
- EBS direct API は ローカルスナップショット ではサポートされていません。
- ローカルスナップショットまたは AMIs を から AWS リージョンOutpost、ある から別の Outpost 、または 内でコピーすることはできませんOutpost。ただし、 AWS リージョンから Outpost に、スナップショットをコピーすることは可能です。詳細については、「AWS リージョンから にスナップショットをコピーする Outpost」を参照してください。
- AWS リージョンから にスナップショットをコピーするとOutpost、データはサービスリンク経由で転送されます。複数のスナップショットを同時にコピーすると、Outpost で実行されている他のサービスに影響を与える可能性があります。
- ローカルスナップショット は共有できません。
- データレジデンシー要件が満たされていることを確認するには、IAM ポリシーを使用する必要があります。詳細については、「IAM によるアクセスの制御」を参照してください。
- ローカルスナップショット は増分バックアップです。最新のスナップショットが作成された後に変更された、ボリューム内のブロックのみが保存されます。各 ローカルスナップショット には、データを新しい EBS ボリュームに復元するために必要な、(スナップショットが作成された瞬間の) 情報がすべて含まれます。詳細については、Amazon EBS スナップショットの仕組みを参照してください。
- IAM ポリシーを使用して、CopySnapshot と CopyImage アクションに対し、データの所在場所を強制的に指定することはできません。

IAM によるアクセスの制御

AWS Identity and Access Management (IAM) ポリシーを使用して、ローカルスナップショットを操作するときにプリンシパル (AWS アカウント、IAM ユーザー、IAM ロール) が持つアクセス許可を制御できます。次に、ローカルスナップショット で特定のアクションを実行するためのアクセス許可を、付与または拒否する際に使用していただけるポリシーの例を示しました。

> **Important**
>
> Outpost からリージョンへのスナップショットとイメージのコピーは、現在サポートされていません。そのため現時点では、IAM ポリシーを使用して、CopySnapshot と CopyImage アクションにデータの所在場所を指定することはできません。

トピック

- スナップショットのデータの所在場所を強制的に指定する
- プリンシパルによる ローカルスナップショット の削除を防止する

スナップショットのデータの所在場所を強制的に指定する

次のポリシー例では、すべてのプリンシパルが 上のボリュームとインスタンスからスナップショットを作成しOutpostarn:aws:outposts:us-east-1:123456789012:outpost/op-1234567890abcdef、スナップショットデータを AWS リージョンに保存することを禁止しています。プリンシパルは引き続き ローカルスナップショット を作成できます。このポリシーにより、すべてのスナップショットが、その Outpost 上に保持されることが保証されます。

JSON

```
{
    "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": "*"
        }
    ]
}
```

プリンシパルによる ローカルスナップショット の削除を防止する

次のポリシー例では、すべてのプリンシパルに対し、Outpost `arn:aws:outposts:us-east-1:123456789012:outpost/op-1234567890abcdef0` に格納されているローカルスナップショットの削除を禁止しています。

JSON

```
{
    "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": "*"
        }
    ]
}
```

ローカルスナップショット を使用する

このセクションでは、ローカルスナップショット の使用方法について説明します。

トピック

- スナップショットの保存に関するルール
- Outpost 上のボリュームからローカルスナップショットを作成する
- ローカルスナップショット からの AMI の作成
- AWS リージョンから にスナップショットをコピーする Outpost
- AWS リージョンから に AMIs をコピーする Outpost
- ローカルスナップショット からボリュームを作成する
- ローカルスナップショット によってバックアップされた AMI からインスタンスを起動する
- ローカルスナップショット を削除する
- Outpost のスナップショットを自動化する

スナップショットの保存に関するルール

スナップショットの保存には、次のルールが適用されます。

- ボリュームの最新のスナップショットを Outpost に保存した後は、後続するすべてのスナップショットも同じ Outpost に保存する必要があります。
- ボリュームの最新のスナップショットが AWS リージョンに保存されている場合、連続するすべてのスナップショットを同じリージョンに保存する必要があります。そのボリュームから ローカルスナップショット の作成を開始するには、次の操作を行います。

  1. AWS リージョンにボリュームのスナップショットを作成します。
  2. AWS リージョンOutpostから にスナップショットをコピーします。
  3. ローカルスナップショット から新しいボリュームを作成します。
  4. そのボリュームを Outpost 上のインスタンスにアタッチします。

  Outpost の新しいボリュームの場合、そこで作成されるスナップショットは Outpost または AWS リージョンに保存できます。その後、連続するすべてのスナップショットは、同じロケーションに保存する必要があります。

- で作成されたスナップショットOutpostや AWS リージョンOutpostから にコピーされたスナップショットを含むローカルスナップショットは、同じ にボリュームを作成する場合にのみ使用できますOutpost。
- リージョンにあるスナップショットから Outpost にボリュームを作成する場合、その新しいボリュームの後に続くスナップショットはすべて、同じリージョンに置かれる必要があります。
- ローカルスナップショットから Outpost にボリュームを作成する場合、その新しいボリュームの後に続くスナップショットはすべて、同じ Outpost 上に置かれる必要があります。

Outpost 上のボリュームからローカルスナップショットを作成する

Outpost 上のボリュームからローカルスナップショットを作成することができます。スナップショットは、ソースボリュームと同じ Outpost に保存することも、Outpost で使用しているリージョンに保存することもできます。

ローカルスナップショットは、同じ Outpost にボリュームを作成するためにのみ使用できます。

詳細については、Amazon EBS スナップショットの作成を参照してください。

ローカルスナップショット からの AMI の作成

ローカルスナップショットと、Outpost のリージョンに格納されているスナップショットの組み合わせを使用して、Amazon マシンイメージ (AMI) を作成できます。例えば、`us-east-1` に Outpost がある場合には、その Outpost 上でローカルスナップショットによりバックアップされたデータボリュームと、`us-east-1` リージョン内のスナップショットによりバックアップされたルートボリュームを持つ、AMI を作成できます。

ローカルスナップショットOutpostをサポートする のインスタンスから直接 AMI を作成することもできます。AMI 用に作成されたバッキングスナップショットは、インスタンスOutpostと同じ またはその親リージョンに保存できます。

> **Note**
>
> - 複数の Outposts に保存されたバックアップ用のスナップショットを含む AMI を作成することはできません。
> - ローカルスナップショットによってバックアップされた AMI は、同じ Outpost でインスタンスを起動するためにのみ使用できます。

リージョンに置かれたスナップショットから Outpost に AMI を作成するには

1. リージョンから Outpost にスナップショットをコピーします。詳細については、「AWS リージョンから にスナップショットをコピーする Outpost」を参照してください。
2. Amazon EC2 コンソールまたは register-image コマンドを使用して、Outpost にコピーしたスナップショットから AMI を作成します。詳細については、スナップショットから AMI を作成するを参照してください。

のインスタンスから直接 AMI を作成するには Outpost

create-image コマンドを `--snapshot-location`パラメータとともに使用して、バッキングスナップショットの保存場所を制御します。

- スナップショットをインスタンスOutpostと同じ にローカルに保存するには、 を指定しますlocal。
- の親リージョンにスナップショットを保存するにはOutpost、 を指定しますregional。

ソースインスタンスがローカルスナップショットOutpostをサポートする にある場合は、 `--snapshot-location`パラメータが必要です。これを省略すると、リクエストはInvalidParameterValueエラーで失敗します。

次の例では、AMI を作成し、インスタンスOutpostと同じ にバッキングスナップショットを保存します。

```
aws ec2 create-image \
    --instance-id i-1234567890abcdef0 \
    --name "My Outpost image" \
    --snapshot-location local
```

Outpost のインスタンスからリージョンに AMI を作成するには

create-image コマンドを使用して、 `--snapshot-location`パラメータregionalに を指定します。

次の例では、AMI を作成し、 の親リージョンにバッキングスナップショットを保存しますOutpost。

```
aws ec2 create-image \
    --instance-id i-1234567890abcdef0 \
```

```
    --name "My Regional image" \
    --snapshot-location regional
```

AWS リージョンから にスナップショットをコピーする Outpost

AWS リージョンから にスナップショットをコピーできますOutpost。これは、スナップショットが Outpost で使用しているリージョンにある場合にのみ実行できます。スナップショットが別のリージョンにある場合は、まずスナップショットを Outpost で使用しているリージョンにコピーした上で、そのリージョンから Outpost にコピーする必要があります。

> **Note**
>
> Outpost とリージョン間、異なる Outpost 間、または同じ Outpost 内で、ローカルスナップショットをコピーすることはできません。

詳細については、「Amazon EBS スナップショットのコピー」を参照してください。

AWS リージョンから に AMIs をコピーする Outpost

AMIs、 AWS リージョンから にコピーできますOutpost。AMI をリージョンから Outpost にコピーすると、その AMI に関連付けられたすべてのスナップショットも、リージョンから Outpost にコピーされます。

AMI に関連付けられたスナップショットが Outpost のリージョンに置かれている場合にのみ、その AMI をリージョンから Outpost にコピーすることができます。スナップショットが別のリージョンにある場合は、最初に AMI を Outpost で使用するリージョンにコピーし、そのリージョンから AMI を Outpost にコピーする必要があります。

> **Note**
>
> Outpost とリージョン間、異なる Outpost 間、または同一の Outpost 内で、AMI をコピーすることはできません。

copy-image AWS CLI コマンドのみOutpostを使用して、リージョンから に AMIs をコピーできます。

ローカルスナップショット からボリュームを作成する

ローカルスナップショットから Outpost にボリュームを作成できます。このボリュームは、ソーススナップショットと同じ Outpost に作成する必要があります。Outpost のリージョンにボリュームを作成するために、ローカルスナップショットを使用することはできません。

ローカルスナップショット からボリュームを作成する場合、異なる KMS キー を使用して、そのボリュームを再暗号化することはできません。ローカルスナップショット から作成されたボリュームは、ソーススナップショットと同じ KMS キーを使用して暗号化する必要があります。

詳細については、「Amazon EBS ボリュームの作成」を参照してください。

ローカルスナップショット によってバックアップされた AMI からインスタンスを起動する

ローカルスナップショット によってバックアップされた AMI からインスタンスを起動できます。インスタンスは、ソース AMI と同じ Outpost で起動する必要があります。詳細については、「User guide for first-generation AWS Outposts racks」の「Launch an instance on your Outpost rack」をご参照ください。

ローカルスナップショット を削除する

ローカルスナップショットを、Outpost から削除することができます。スナップショットを Outpost から削除した場合、削除したスナップショットで使用されていた Amazon S3 のストレージ容量は、そのスナップショットを参照していたスナップショットとボリュームの削除から、72 時間以内に使用できるようになります。

Amazon S3 ストレージ容量はすぐには利用できないので、Amazon CloudWatch アラームを使用して Amazon S3 ストレージ容量を監視することをお勧めします。ストレージ容量の不足を避けるため、必要がなくなったスナップショットとボリュームを削除します。

スナップショットの削除の詳細については、スナップショットを削除するを参照してください。

Outpost のスナップショットを自動化する

Amazon Data Lifecycle Manager のスナップショットライフサイクルポリシーを作成すると、Outpost 上にあるボリュームとインスタンスのために、スナップショットの作成、コピー、保持、削除を自動化できます。スナップショットをリージョンに保存するか、それらを Outpost でローカルに保存するかを選択できます。さらに、 AWS リージョンで作成および保存されたスナップショットを に自動的にコピーできますOutpost。

次の表に、サポートされている機能の概要を示します。

リソースのロケーション	スナップショットの送信先	リージョン間のコピー	へのクロスリージョンコピー Outpost	高速スナップショット復元	クロスアカウントの共有
リージョン	リージョン	✓	✓	✓	✓
Outpost	リージョン	✓	✓	✓	✓
Outpost	Outpost	✗	✗	✗	✗

考慮事項

- 現行では、Amazon EBS スナップショットライフサイクルポリシーのみがサポートされています。EBS-backed AMI ポリシー、およびクロスアカウント共有イベントポリシーは、サポートされていません。
- ポリシーにより、リージョン内のボリュームまたはインスタンス用のスナップショットを管理する場合、そのスナップショットは、ソースリソースと同じリージョンに作成されます。
- ポリシーが、Outpost 上にあるボリュームまたはインスタンス用のスナップショットを管理している場合は、ソース Outpost、またはその Outpost が使用しているリージョンのどちらかに、スナップショットを作成できます。
- リージョン内のスナップショットと、Outpost 上にあるスナップショットの両方を、1 つのポリシーで管理することはできません。リージョンと Outpost でスナップショットを自動化する必要がある場合は、個別にポリシーを作成する必要があります。
- 高速スナップショット復元は、Outpost で作成されたスナップショット、または Outpost にコピーされたスナップショットではサポートされていません。
- Outpost で作成されたスナップショットでは、クロスアカウント共有はサポートされていません。
- プレスクリプトとポストスクリプトは、 で作成されたスナップショットではサポートされていませんOutpost。

ローカルスナップショット を管理するスナップショットライフサイクルの作成の詳細については、Automating snapshot lifecycles (スナップショットライフサイクルの自動化)を参照してください。

ローカルゾーン内のローカルスナップショット

Amazon EBS スナップショットは、EBS ボリュームのポイントインタイムコピーです。

AWS ローカルゾーン内の EBS ボリュームのスナップショットは、同じローカルゾーンの Amazon S3 またはそのローカルゾーンの親リージョンに保存できます。ローカルゾーンにスナップショットを保存すると、スナップショットデータが特定の国、州、または市区町村で処理および保存されるため、データレジデンシーのニーズを満たすことができます。IAM を使用してデータレジデンシー強制ポリシーを設定して、スナップショットデータがローカルゾーンから出ないようにすることもできます。

ローカルゾーンは、 AWS インフラストラクチャをエンドユーザーやビジネスセンターに近づけることで、1 桁ミリ秒のレイテンシーまたはローカルデータ処理を必要とするアプリケーションに最適です。さらに、規制やコンプライアンスに敏感なワークロードのデータレジデンシー要件を満たすことができます。詳細については、AWS 「ローカルゾーンとは」を参照してください。

ローカルスナップショットは現在、Amazon S3 をサポートするローカルゾーンでサポートされています。詳細については、「AWS Local Zones features」を参照してください。

トピック

- よくある質問
- 考慮事項
- IAM によるアクセスの制御

よくある質問

1. ローカルゾーン内のローカルスナップショットとは何ですか?

   ローカルゾーン内のローカルスナップショットとは、ローカルゾーン内の Amazon S3 に保存されるスナップショットのことです。デフォルトでは、ローカルゾーン内のボリュームの Amazon EBS スナップショットは、親リージョンにある Amazon S3 内に保存されます。ローカルゾーンが Amazon S3 をサポートしている場合は、代わりにスナップショットをローカルゾーンにローカルに保存することができます。 AWS リージョンのスナップショットと同様に、ローカルゾーンのローカルスナップショットは増分です。つまり、最新のスナップショットの後に変更されたボリュームのブロックのみが保存されます。これらのスナップショットを使用して、Amazon EBS ボリュームを同じローカルゾーン内でいつでも復元できます。

2. ローカルスナップショットを使用する理由は何ですか?

   ローカルゾーンのローカルスナップショットを使用して、スナップショットデータが国、州、市区町村などの特定の地理的場所に保存されるようにすることで、データレジデンシーまたはデータ分離の要件を満たすことができます。

3. スナップショットのデータレジデンシーを強制的にローカルゾーンにするにはどうすればよいですか?

   AWS Identity and Access Management (IAM) ポリシーを使用して、ローカルゾーンでローカルスナップショットを操作するときにプリンシパル (AWS アカウント、IAM ユーザー、IAM ロール) が持つアクセス許可を制御し、データレジデンシーを適用できます。たとえば、ユーザーがローカルゾーンのボリュームからスナップショットを作成し、それらのスナップショットを AWS リージョンに保存できないようにするポリシーを作成できます。詳細については、「IAM によるアクセスの制御」を参照してください。

4. マルチボリュームの Crash-consistent なローカルスナップショットはサポートされていますか?

   はい。ローカルゾーン内のインスタンスから、ローカルゾーンにマルチボリュームの Crash-consistent なローカルスナップショットを作成できます。

5. ローカルゾーンにローカルスナップショットを作成するにはどうしたらよいですか?

   Local Zones でローカルスナップショットを手動で作成するには、 AWS CLI または Amazon EC2 コンソールを使用します。詳細については、EBS ボリュームのスナップショットを作成するを参照してください。また、Amazon Data Lifecycle Manager を使用することで、ローカルゾーン内のローカルスナップショットのライフサイクルを自動化することもできます。詳細については、EBS スナップショット用の Amazon Data Lifecycle Manager カスタムポリシーを作成 を参照してください。

6. ローカルゾーン内のローカルスナップショットをコピーできますか?

   はい。スナップショットは、リージョンから同じリージョン内のローカルゾーン、ローカルゾーンからそのリージョン、あるローカルゾーンから同じローカルゾーングループ内の別のローカルゾーンにコピーできます。

7. ローカルゾーン内のローカルスナップショットからデータを復元するにはどうすればよいですか?

   ローカルゾーン内のローカルスナップショットを使用して、同じローカルゾーンにのみ Amazon EBS ボリュームを作成できます。

8. ローカルゾーン内のローカルスナップショットはどのように暗号化されますか?

   ローカルゾーン固有の暗号化動作Local Zones、Outposts、Wavelength Zones での Amazon EBS 暗号化については、「」を参照してください。Amazon EBS リソースがデフォルトで暗号化され

ているローカルゾーンでは、ローカルスナップショットから作成されたボリュームは常にスナップショットと同じ AWS KMS キーを使用して暗号化されます。ただし、ローカルスナップショットをコピーするときに暗号化キーを変更できます。

9. ローカルゾーン内のローカルスナップショットを使用して EBS-backed AMI を作成できますか?

   はい。スナップショットの送信先をローカルゾーンとして指定することで、EBS-backed AMI の作成時にローカルゾーン内のローカルスナップショットを使用できます。詳細については、「Amazon EBS-backed AMI を作成する」を参照してください。

10. ローカルゾーン内のローカルスナップショットを共有できますか?

    はい。ローカルゾーンのローカルスナップショットを、その AWS アカウントでローカルゾーンの使用を有効にした他のアカウントと共有できます。

11. ボリュームのローカルスナップショットを作成してから、親リージョンでスナップショットを作成するように切り替えることはできますか?

    いいえ。ボリュームのローカルスナップショットを作成した後、親リージョンでそのボリュームのスナップショットを連続して作成することはできません。すべてのスナップショットは増分であるため、ボリュームの最新のスナップショットがローカルスナップショットである場合、そのボリュームの連続するスナップショットはすべて、ローカルスナップショットである必要があります。

考慮事項

ローカルゾーン内のローカルスナップショットを使用する際には、次の点に注意してください。

- ローカルスナップショットは現在、Amazon S3 をサポートするローカルゾーンでサポートされています。
- 次の機能は、ローカルゾーン内のローカルスナップショットでは使用できません。
  - VM Import/Export アクション
  - 高速スナップショット復元
  - EBS direct API
  - ごみ箱
  - スナップショットアーカイブ
- IAM ポリシーを使用して、データレジデンシーの要件を強制する必要があります。詳細については、「IAM によるアクセスの制御」を参照してください。

- ボリュームの最新のスナップショットがローカルスナップショットである場合、連続するすべてのスナップショットはローカルスナップショットである必要があります。同様に、ボリュームの最新のスナップショットが AWS リージョンに保存されている場合、連続するすべてのスナップショットを同じリージョンに保存する必要があります。

IAM によるアクセスの制御

AWS Identity and Access Management (IAM) ポリシーを使用して、ローカルゾーンでローカルスナップショットを操作するときにプリンシパル (AWS アカウント、IAM ユーザー、IAM ロール) が持つアクセス許可を制御できます。次のポリシー例は、ローカルゾーン内のローカルスナップショットで特定のアクションを実行するためのアクセス許可を付与または拒否する際に使用できます。

トピック

- ローカルスナップショットのデータレジデンシーを強制的にローカルゾーンにする
- ローカルゾーンでローカルスナップショットを共有できないようにする
- プリンシパルがローカルゾーン内のローカルスナップショットを削除できないようにする

ローカルスナップショットのデータレジデンシーを強制的にローカルゾーンにする

次のポリシー例では、ローカルゾーン内のボリュームとインスタンスからローカルゾーン内のローカルスナップショットのみを作成するようにユーザーを制限しています。これにより、ユーザーはローカルゾーン内のボリュームとインスタンスからリージョンにスナップショットを作成できなくなります。

JSON

```
{
    "Version":"2012-10-17",
    "Statement":
    [
        {
            "Effect": "Deny",
            "Action":
            [
                "ec2:CreateSnapshot",
                "ec2:CreateSnapshots"
            ],
            "Resource": "*",
```

```
            "Condition":
            {
                "StringEquals":
                {
                    "ec2:Location": "regional",
                    "ec2:SourceAvailabilityZone": "local_zone"
                }
            }
        },
        {
            "Effect": "Deny",
            "Action":
            [
                "ec2:CreateSnapshot",
                "ec2:CreateSnapshots"
            ],
            "Resource": "*",
            "Condition":
            {
                "StringEquals":
                {
                    "ec2:SourceAvailabilityZone": "local_zone"
                },
                "Null":
                {
                    "ec2:Location": "true"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action":
            [
                "ec2:CreateSnapshot",
                "ec2:CreateSnapshots"
            ],
            "Resource": "*"
        }
    ]
}
```

ローカルゾーンでローカルスナップショットを共有できないようにする

次のポリシー例では、すべてのユーザーがローカルゾーンでローカルスナップショットを共有できないようにしています。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "DenySnapshotModifyInLocalZone",
      "Effect": "Deny",
      "Action": [
        "ec2:ModifySnapshotAttribute"
      ],
      "Resource": "arn:aws:ec2:*::snapshot/*",
      "Condition": {
        "StringEquals": {
        "ec2:AvailabilityZone": "use1-atl2-az1"
        }
      }
    },
    {
      "Sid": "AllowSnapshotModifyElsewhere",
      "Effect": "Allow",
      "Action": [
        "ec2:ModifySnapshotAttribute"
      ],
      "Resource": "arn:aws:ec2:*::snapshot/*"
    }
  ]
}
```

プリンシパルがローカルゾーン内のローカルスナップショットを削除できないようにする

次のポリシー例では、すべてのユーザーがローカルゾーン内のローカルスナップショットを削除できないようにしています。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "DenySnapshotDeleteInLocalZone",
      "Effect": "Deny",
      "Action": [
        "ec2:DeleteSnapshot"
      ],
      "Resource": "arn:aws:ec2:*::snapshot/*",
      "Condition": {
        "StringEquals": {
        "ec2:AvailabilityZone": "use1-atl2-az1"
        }
      }
    },
    {
      "Sid": "AllowSnapshotDeleteElsewhere",
      "Effect": "Allow",
      "Action": [
        "ec2:DeleteSnapshot"
      ],
      "Resource": "arn:aws:ec2:*::snapshot/*"
    }
  ]
}
```

Amazon EBS 暗号化

Amazon EBS 暗号化は、Amazon EC2 インスタンスに関連付けられた EBS リソースの簡単な暗号化ソリューションとして使用できます。Amazon EBS 暗号化では、独自のキー管理インフラストラクチャを構築、保守、保護する必要はありません。Amazon EBS 暗号化は、暗号化されたボリュームとスナップショットを作成するときに、 AWS KMS keys を使用します。

暗号化オペレーションは EC2 インスタンスをホストするサーバー上で実行され、インスタンスとそれに接続された EBS ストレージ間でのデータの保存と転送中のデータの両方のセキュリティを保証します。

1 つのインスタンスに対し、暗号化されたボリュームと暗号化されていないボリュームの両方を、同時にアタッチできます。すべての Amazon EC2 インスタンスタイプは Amazon EBS 暗号化をサポートしています。

内容

- Amazon EBS 暗号化の仕組み
- Local Zones、Outposts、Wavelength Zones での Amazon EBS 暗号化
- Amazon EBS 暗号化の要件
- デフォルトで Amazon EBS の暗号化を有効化
- EBS リソースの暗号化
- Amazon EBS 暗号化に使用される AWS KMS キーをローテーションする
- Amazon EBS 暗号化の例

Amazon EBS 暗号化の仕組み

EC2 インスタンスのブートボリュームとデータボリュームの両方を暗号化できます。

暗号化された EBS ボリュームを作成し、サポートされるインスタンスタイプにアタッチする場合、以下のタイプのデータが暗号化されます。

- ボリューム内の保存データ
- ボリュームとインスタンスの間で移動されるすべてのデータ
- ボリュームから作成されたすべてのスナップショット
- それらのスナップショットから作成されたすべてのボリューム

Amazon EBS は、業界標準の AES-256 データ暗号化を使用して、ボリュームをデータキーで暗号化します。データキーは によって生成され AWS KMS 、 によって AWS KMS キー AWS KMS で暗号化されてから、ボリューム情報とともに保存されます。Amazon EBS は、Amazon EBS リソースを作成する各リージョン AWS マネージドキー に一意の を自動的に作成します。KMS キーのエイリアスは aws/ebs です。デフォルトでは、Amazon EBS は暗号化にこの KMS キー を使用します。または、作成した対称カスタマーマネージド暗号化キーを使用することもできます。独自の KMS キーを使用することにより、KMS キー の作成、更新、無効化ができるなど、より高い柔軟性が得られます。

Amazon EC2 は と連携して AWS KMS 、暗号化されたボリュームを作成するスナップショットが暗号化されているかどうかに応じて、EBS ボリュームをわずかに異なる方法で暗号化および復号化します。

暗号化されたスナップショットに対する EBS 暗号化の動作

所有している暗号化されたスナップショットから暗号化されたボリュームを作成すると、Amazon EC2 は AWS KMS と連携して EBS ボリュームを次のように暗号化および復号します。

1. Amazon EC2 は GenerateDataKeyWithoutPlaintext リクエストを に送信し AWS KMS、ボリューム暗号化用に選択した KMS キーを指定します。
2. ボリュームがスナップショットと同じ KMS キーを使用して暗号化されている場合、 はスナップショットと同じデータキー AWS KMS を使用し、同じ KMS キーで暗号化します。ボリュームが別の KMS キーを使用して暗号化されている場合、 は新しいデータキー AWS KMS を生成し、指定した KMS キーで暗号化します。暗号化されたデータキーは Amazon EBS に送信され、ボリュームメタデータとともに保存されます。
3. 暗号化されたボリュームをインスタンスにアタッチすると、Amazon EC2 は CreateGrant リクエストを AWS KMS に送信し、データキーを復号化できるようにします。
4. AWS KMS は暗号化されたデータキーを復号し、復号されたデータキーを Amazon EC2 に送信します。
5. Amazon EC2 は、Nitro ハードウェア内のプレーンテキストデータキーを使用して、ボリュームのディスク I/O を暗号化します。プレーンテキストデータキーは、ボリュームがインスタンスにアタッチされる限り、メモリ内で維持されます。

暗号化されていないスナップショットに対する EBS 暗号化の動作

暗号化されていないスナップショットから暗号化されたボリュームを作成する場合、Amazon EC2 は AWS KMS と連携して、次のように EBS ボリュームを暗号化および復号化します。

1. Amazon EC2 は CreateGrant リクエストを に送信し AWS KMS、スナップショットから作成されたボリュームを暗号化できるようにします。
2. Amazon EC2 は GenerateDataKeyWithoutPlaintext リクエストを に送信し AWS KMS、ボリューム暗号化用に選択した KMS キーを指定します。
3. AWS KMS は新しいデータキーを生成し、ボリューム暗号化用に選択した KMS キーで暗号化し、暗号化されたデータキーを Amazon EBS に送信してボリュームメタデータとともに保存します。
4. Amazon EC2 は、暗号化されたデータキーを復号 AWS KMS するために Decrypt リクエストを に送信し、ボリュームデータの暗号化に使用します。
5. 暗号化されたボリュームをインスタンスにアタッチすると、Amazon EC2 は CreateGrant リクエストを AWS KMSに送信し、データキーを復号化できるようにします。
6. 暗号化されたボリュームをインスタンスにアタッチすると、Amazon EC2 は暗号化されたデータキーを指定して AWS KMSDecrypt リクエストを に送信します。
7. AWS KMS は暗号化されたデータキーを復号し、復号されたデータキーを Amazon EC2 に送信します。
8. Amazon EC2 は、Nitro ハードウェア内のプレーンテキストデータキーを使用して、ボリュームのディスク I/O を暗号化します。プレーンテキストデータキーは、ボリュームがインスタンスにアタッチされる限り、メモリ内で維持されます。

詳細については、AWS Key Management Service デベロッパーガイドのAmazon Elastic Block Store (Amazon EBS) で AWS KMSを使用する方法およびAmazon EC2 の例 2を参照してください。

使用できない KMS キーがデータキーに及ぼす影響

KMS キーが使用できなくなると、その影響はほぼ即時に表れます (最終的な一貫性の対象となります)。KMS キーのキーステータスは新しい条件を反映して変化し、暗号化オペレーションで KMS キーを使用するすべてのリクエストは失敗します。

KMS キーを使用不可にするアクションを実行しても、EC2 インスタンスまたはアタッチされた EBS ボリュームに対して直ちに影響が及ぶことはありません。Amazon EC2 は、ボリュームがインスタンスにアタッチされている間、KMS キーではなくデータキーを使用してすべてのディスク I/O を暗号化します。

ただし、暗号化された EBS ボリュームが EC2 インスタンスからデタッチされると、Amazon EBS は Nitro ハードウェアからデータキーを削除します。次回、暗号化された EBS ボリュームが EC2 インスタンスにアタッチされると、アタッチメントは失敗します。これは、Amazon EBS は KMS キー

を使用してボリュームの暗号化されたデータキーを復号できないためです。EBS ボリュームを再度使用するには、KMS キーを再度使用可能にする必要があります。

> **Tip**
>
> 使用不可にする KMS キーから生成されたデータキーで暗号化された EBS ボリュームのデータにアクセスする必要がなくなった場合は、KMS キーを使用不可にする前に EC2 インスタンスから EBS ボリュームをデタッチすることをお勧めします。

詳細については、「AWS Key Management Service デベロッパーガイド」の「AWS KMS keys」を参照してください。

Local Zones、Outposts、Wavelength Zones での Amazon EBS 暗号化

Amazon EBS の暗号化動作は、リソースを作成する場所のタイプによって異なります。

ローカルゾーン

次のローカルゾーンでは、すべての Amazon EBS リソースがデフォルトで暗号化されます。暗号化されていないボリュームやスナップショットを作成することはできません。KMS キーを指定しない場合、Amazon EBS はアカウントのデフォルトの暗号化キーを使用してリソースを暗号化します。デフォルトでは、これは AWS マネージドキー () ですaws/ebs。アカウントのデフォルトの暗号化キーを変更するには、「」を参照してくださいデフォルトの暗号化キー。

- 北米: us-east-1-atl-1a、us-east-1-bos-1a、us-east-1-chi-1a、、us-east-1-dfw-1a、us-east-1-iah-1a、us-east-1-mci-1a、us-east-1-mia-1a、、us-east-1-msp-1aus-east-1-nyc-1aus-east-1-phl-1a、、us-east-1-qro-1a、、us-east-2-mci-1a、、us-east-2-mci-1b、us-west-2-den-1aus-west-2-hnl-1a、us-west-2-las-1aus-west-2-las-1bus-west-2-pdx-1aus-west-2-phx-1a、、、 us-west-2-sea-1a
- アジアパシフィック: ap-northeast-1-tpe-1a、ap-south-1-ccu-1a、ap-south-1-ccu-2a、、ap-south-1-del-1a、ap-south-1-del-2a、ap-southeast-1-bkk-1a、ap-southeast-1-han-1aap-southeast-1-mnl-1aap-southeast-2-akl-1a、、 ap-southeast-2-per-1a

- 欧州: eu-central-1-ath-1a、eu-central-1-ham-1a、eu-central-1-ist-1a、eu-central-1-waw-1a、eu-north-1-cph-1a、 eu-north-1-hel-1a
- 南米: us-east-1-bue-1a、us-east-1-lim-1a、 us-east-1-scl-1a
- 中東: me-south-1-mct-1a
- アフリカ: af-south-1-los-1a

他のすべてのローカルゾーンでは、Amazon EBS 暗号化は親リージョンと同じ動作に従います。

Outposts と Wavelength Zones

および Wavelength Zones のすべての Amazon EBS AWS Outposts リソースは、デフォルトで暗号化されます。暗号化されていないボリュームやスナップショットを作成することはできません。KMS キーを指定しない場合、Amazon EBS はアカウントのデフォルトの暗号化キーを使用してリソースを暗号化します。デフォルトでは、これは AWS マネージドキー () ですaws/ebs。アカウントのデフォルトの暗号化キーを変更するには、「」を参照してくださいデフォルトの暗号化キー。

これらのゾーンでローカルスナップショットからボリュームを作成すると、ボリュームはソーススナップショットと同じ KMS キーを使用して暗号化されます。ただし、ローカルスナップショットをコピーするときに暗号化キーを変更できます。

Amazon EBS 暗号化の要件

開始する前に、以下の要件が満たされていることを確認します。

要件

- サポートされるボリュームタイプ
- サポートされるインスタンスタイプ
- ユーザーのアクセス許可
- インスタンスの権限

サポートされるボリュームタイプ

暗号化は、すべての EBS ボリュームタイプでサポートされます。暗号化されたボリュームでは、暗号化されていないボリュームと同じ IOPS パフォーマンスが期待できます。遅延に対する影響は最小限に抑えられます。暗号化されていないボリュームにアクセスするのと同じ方法で、暗号化されたボ

リュームにアクセスできます。暗号化と復号は透過的に処理され、ユーザーやアプリケーションから追加の操作を必要としません。

サポートされるインスタンスタイプ

Amazon EBS 暗号化は、すべての現行世代および前世代のインスタンスタイプで利用できます。

ユーザーのアクセス許可

EBS 暗号化に KMS キーを使用する場合、KMS キーポリシーは、必要な AWS KMS アクションにアクセスできるすべてのユーザーがこの KMS キーを使用して EBS リソースを暗号化または復号できるようにします。EBS 暗号化を使用するには、次のアクションを呼び出す許可をユーザーに付与する必要があります。

- `kms:CreateGrant`
- `kms:Decrypt`
- `kms:DescribeKey`
- `kms:GenerateDataKeyWithoutPlaintext`
- `kms:ReEncrypt`

> **Tip**
>
> 最小権限のプリンシパルに従うには、`kms:CreateGrant` へのフルアクセスを許可しないでください。代わりに、次の例に示すように、 AWS サービスによってユーザーに代わって権限が作成された場合にのみ、 `kms:GrantIsForAWSResource`条件キーを使用して、ユーザーが KMS キーに権限を作成できるようにします。

JSON

```
{
    "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
                }
            }
        }
    ]
}
```

詳細については、「 AWS Key Management Service デベロッパーガイド」の「デフォルトキーポリシー」セクションの AWS 「アカウントへのアクセスを許可し、IAM ポリシーを有効にする」を参照してください。

インスタンスの権限

インスタンスが暗号化された AMI、ボリューム、またはスナップショットと通信しようとすると、インスタンスの ID 専用ロールに KMS キーグラントが発行されます。ID 専用ロールは、インスタンスがユーザーに代わって暗号化された AMI、ボリューム、またはスナップショットを操作するために使用する IAM ロールです。

ID のみのロールは、手動で作成または削除する必要はなく、ポリシーも関連付けられていません。また、ID のみのロール認証情報にはアクセスできません。

> **Note**
>
> ID 専用ロールは、インスタンス上のアプリケーションが Amazon S3 オブジェクトや Dynamo DB テーブルなどの他の AWS KMS 暗号化されたリソースにアクセスするために使用されません。これらのオペレーションは、Amazon EC2 インスタンスロールの認証情報、またはインスタンスで設定したその他の AWS 認証情報を使用して実行されます。

ID のみのロールには、サービスコントロールポリシー (SCP) と KMS キーポリシーが適用されます。SCP キーまたは KMS キーが KMS キーへの ID 専用ロールアクセスを拒否すると、暗号化されたボリュームで、または暗号化された AMI やスナップショットを使用して EC2 インスタンスを起動できないことがあります。

aws:SourceIp、、、または aws:SourceVpce AWS グローバル条件キーを使用してネットワークの場所に基づいてアクセスを拒否する SCP aws:VpcSourceIpaws:SourceVpcまたはキーポリシーを作成する場合は、これらのポリシーステートメントがインスタンスのみのロールに適用されないようにする必要があります。ポリシーの例については、「データペリメータポリシーの例」を参照してください。

ID 専用ロール ARN は次の形式を使用します:

```
arn:aws-partition:iam::account_id:role/aws:ec2-infrastructure/instance_id
```

キーグラントがインスタンスに発行されると、キーグラントはそのインスタンス固有のロール割り当てセッションに発行されます。被付与者のプリンシパル ARN は以下の形式を使用します:

```
arn:aws-partition:sts::account_id:assumed-role/aws:ec2-infrastructure/instance_id
```

デフォルトで Amazon EBS の暗号化を有効化

作成した新しい EBS ボリュームとスナップショットコピーの暗号化を強制するように AWS アカウントを設定できます。例えば、Amazon EBS は、インスタンスの起動時に作成された EBS ボリュームと、暗号化されていないスナップショットからコピーしたスナップショットを暗号化します。暗号化されていない EBS リソースから暗号化された EBS リソースへの移行の例については、暗号化されていないリソースの暗号化を参照してください。

デフォルトでは、暗号化は既存の EBS ボリュームまたはスナップショットには影響しません。

考慮事項

- デフォルトでの暗号化はリージョン固有の設定です。リージョンに対して有効にした場合、そのリージョン内の個々のボリュームまたはスナップショットに対して無効にすることはできません。
- デフォルトで Amazon EBS 暗号化は、すべての現行世代および前世代のインスタンスタイプでサポートされています。
- スナップショットをコピーして、新しい KMS キーで暗号化すると、完全な (増分ではない) コピーが作成されます。その結果、追加のストレージコストが発生します。

Console

リージョンの暗号化をデフォルトで有効にするには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションバーから、使用するリージョンを選択します。
3. ナビゲーションペインから設定を選択し、データ保護とセキュリティタブを選択します。
4. [EBS 暗号化] セクションで、[管理] を選択します。
5. [Enable] (有効化) を選択します。は、デフォルトの暗号化キーとしてユーザーに代わってaws/ebs作成されたエイリアス AWS マネージドキー で保持するか、対称カスタマーマネージド暗号化キーを選択します。
6. [Update EBS encryption] (EBS 暗号化を更新する) を選択します。

AWS CLI

デフォルトの暗号化設定を表示するには

get-ebs-encryption-by-default コマンドを使用します。

- 特定のリージョンの場合

```
aws ec2 get-ebs-encryption-by-default --region region
```

- アカウントの全リージョンの場合

```
echo -e "Region      \t Encrypt \t Key"; \
echo -e "----------- \t ------- \t -------" ; \
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 $region | jq
 '.KmsKeyId');
    echo -e "$region \t $default \t\t $kms_key";
done
```

暗号化をデフォルトで有効にするには

enable-ebs-encryption-by-default コマンドを使用します。

- 特定のリージョンの場合

```
aws ec2 enable-ebs-encryption-by-default --region region
```

- アカウントの全リージョンの場合

```
echo -e "Region      \t Encrypt \t Key"; \
echo -e "----------- \t ------- \t -------" ; \
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 $region | jq
 '.KmsKeyId');
    echo -e "$region \t $default \t\t $kms_key";
done
```

暗号化をデフォルトで無効にするには

disable-ebs-encryption-by-default コマンドを使用します。

- 特定のリージョンの場合

```
aws ec2 disable-ebs-encryption-by-default --region region
```

- アカウントの全リージョンの場合

```
echo -e "Region      \t Encrypt \t Key"; \
echo -e "----------- \t ------- \t -------" ; \
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 $region | jq
 '.KmsKeyId');
    echo -e "$region \t $default \t\t $kms_key";
```

```
done
```

PowerShell

デフォルトの暗号化設定を表示するには

Get-EC2EbsEncryptionByDefault コマンドレットを使用します。

- 特定のリージョンの場合

```
Get-EC2EbsEncryptionByDefault -Region region
```

- アカウントの全リージョンの場合

```
(Get-EC2Region).RegionName |
    ForEach-Object {
    [PSCustomObject]@{
        Region                    = $_
        EC2EbsEncryptionByDefault = Get-EC2EbsEncryptionByDefault -Region $_
        EC2EbsDefaultKmsKeyId     = Get-EC2EbsDefaultKmsKeyId -Region $_
    } } |
    Format-Table -AutoSize
```

暗号化をデフォルトで有効にするには

Enable-EC2EbsEncryptionByDefault コマンドレットを使用します。

- 特定のリージョンの場合

```
Enable-EC2EbsEncryptionByDefault -Region region
```

- アカウントの全リージョンの場合

```
(Get-EC2Region).RegionName |
    ForEach-Object {
    [PSCustomObject]@{
        Region                    = $_
        EC2EbsEncryptionByDefault = Enable-EC2EbsEncryptionByDefault -Region $_
        EC2EbsDefaultKmsKeyId     = Get-EC2EbsDefaultKmsKeyId -Region $_
    } } |
```

```
    Format-Table -AutoSize
```

暗号化をデフォルトで無効にするには

Disable-EC2EbsEncryptionByDefault コマンドレットを使用します。

- 特定のリージョンの場合

```
Disable-EC2EbsEncryptionByDefault -Region region
```

- アカウントの全リージョンの場合

```
(Get-EC2Region).RegionName |
    ForEach-Object {
    [PSCustomObject]@{
        Region                      = $_
        EC2EbsEncryptionByDefault = Disable-EC2EbsEncryptionByDefault -Region $_
        EC2EbsDefaultKmsKeyId     = Get-EC2EbsDefaultKmsKeyId -Region $_
    } } |
    Format-Table -AutoSize
```

既存のスナップショットまたは暗号化されたボリュームに関連付けられている KMS キー を変更することはできません。ただし、スナップショットコピーオペレーション中に別の KMS キー を関連付けて、コピーしたスナップショットを新しい KMS キー で暗号化できます。

デフォルトの暗号化キー

Amazon EBS は、Amazon EBS リソースを作成するリージョン AWS マネージドキー ごとに一意のを自動的に作成します。この KMS キーのエイリアスは ですaws/ebs。これはデフォルトの暗号化キーです。デフォルトの暗号化キーをカスタマーマネージドキーに変更できます。

リージョンのデフォルトの暗号化キーを変更するには、次のいずれかの方法を使用します。

Console

リージョンのデフォルトの暗号化キーを変更するには

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. ナビゲーションバーから、使用するリージョンを選択します。

3. ナビゲーションペインから設定を選択し、データ保護とセキュリティタブを選択します。
4. [EBS 暗号化] セクションで、[管理] を選択します。
5. デフォルトの暗号化キーで、使用する KMS キー ID を選択します。
6. [Update EBS encryption] (EBS 暗号化を更新する) を選択します。

AWS CLI

modify-ebs-default-kms-key-id コマンドを使用します。

```
aws ec2 modify-ebs-default-kms-key-id --kms-key-id key-id --region region
```

PowerShell

Edit-EC2EbsDefaultKmsKeyId コマンドレットを使用します。

```
Edit-EC2EbsDefaultKmsKeyId -KmsKeyId key-id -Region region
```

EBS リソースの暗号化

EBS ボリュームを暗号化するには、デフォルトでの暗号化を使用するか、暗号化するボリュームを作成するときに暗号化を有効にします。

ボリュームを暗号化する場合、ボリュームの暗号化に使用する対称暗号化 KMS キーを指定できます。KMS キー が指定されていない場合、暗号化に使用される KMS キー はソーススナップショットの暗号化状態とその所有権によって異なります。詳細については、暗号化結果の表を参照してください。

> **Note**
>
> API または を使用して KMS キー AWS CLI を指定する場合は、 が KMS キーを非同期的に AWS 認証することに注意してください。無効な KMS キー ID、エイリアス、または ARN を指定すると、アクションは完了したように見える場合がありますが、最終的には失敗します。

既存のスナップショットまたはボリュームに関連付けられている KMS キー を変更することはできません。ただし、スナップショットコピーオペレーション中に別の KMS キー を関連付けて、コピーしたスナップショットを新しい KMS キー で暗号化できます。

作成時の空のボリュームの暗号化

新しい空の EBS ボリュームを作成するときは、特定のボリューム作成オペレーションで暗号化を有効にすることで暗号化できます。デフォルトで EBS 暗号化を有効にしたボリュームでは、EBS 暗号化用のデフォルト KMS キー を使用した暗号化が、自動的に実行されます。または、ボリュームの作成オペレーションごとに異なる対称暗号化 KMS キーを指定することもできます。ボリュームは最初に使用可能になった時点で暗号化されているため、データは常に保護されています。詳細な手順については、Amazon EBS ボリュームの作成を参照してください。

デフォルトでは、ボリュームの作成時に選択した KMS キー が、ボリュームから作成したスナップショットとそれらの暗号化されたスナップショットから復元したボリュームを暗号化します。暗号化されたボリュームまたはスナップショットから暗号化を削除することはできません。つまり、暗号化されたスナップショット、または暗号化されたスナップショットのコピーから復元されたボリュームは、常に暗号化されます。

暗号化されたボリュームのパブリックスナップショットはサポートされていませんが、暗号化されたスナップショットを特定のアカウントと共有できます。詳細な手順については、Amazon EBS スナップショットを他の AWS アカウントと共有するを参照してください。

暗号化されていないリソースの暗号化

暗号化されていない既存のボリュームまたはスナップショットを直接暗号化することはできません。

暗号化されていないボリュームを暗号化するには、そのボリュームのスナップショットを作成し、そのスナップショットを使用して新しい暗号化されたボリュームを作成します。詳細については、「スナップショットの作成」および「ボリュームの作成」を参照してください。

暗号化されていないスナップショットを暗号化するには、そのスナップショットの暗号化されたコピーを作成します。詳細については、「スナップショットをコピーする」を参照してください。

アカウントでデフォルトで暗号化を有効にすると、暗号化されていないスナップショットから作成されたボリュームとスナップショットコピーは常に暗号化されます。それ以外の場合は、リクエストで暗号化パラメータを指定する必要があります。詳細については、「デフォルトで暗号化の有効化」を参照してください。

Amazon EBS 暗号化に使用される AWS KMS キーをローテーションする

暗号化のベストプラクティスでは、暗号化キーの広範な再利用を推奨していません。

Amazon EBS の暗号化に使用する新しい暗号化マテリアルを作成するには、カスタマーマネージドキーを作成し、アプリケーションを変更してその新しい KMS キーを使用するか、または、既存のカスタマーマネージドキーの自動キーローテーションを有効にすることができます。

カスタマーマネージドキーの自動キーローテーションを有効にすると、 は KMS キーの新しい暗号化マテリアルを毎年 AWS KMS 生成します。 は、暗号化マテリアルのすべての以前のバージョン AWS KMS を保存し、その KMS キーマテリアルで以前に暗号化されたボリュームとスナップショットを引き続き復号して使用できます。 AWS KMS は、KMS キーを削除するまでローテーションされたキーマテリアルを削除しません。

ローテーションされたカスタマーマネージドキーを使用して新しいボリュームまたはスナップショットを暗号化すると、 は現在 (新しい) キーマテリアル AWS KMS を使用します。ローテーションされたカスタマーマネージドキーを使用してボリュームまたはスナップショットを復号化する場合、AWS KMS はそれを暗号化するために使用された暗号化マテリアルバージョンを使用します。ボリュームまたはスナップショットが以前のバージョンの暗号化マテリアルで暗号化されている場合、AWS KMS は引き続きその以前のバージョンを使用して復号します。 AWS KMS は、キーローテーション後に新しい暗号化マテリアルを使用するように以前に暗号化されたボリュームまたはスナップショットを再暗号化しません。これらは、最初に暗号化された暗号化マテリアルで暗号化されたままです。ローテーションされたカスタマーマネージドキーは、コードを変更せずにアプリケーションや AWS サービスで安全に使用できます。

> **Note**
>
> - 自動キーローテーションは、 が AWS KMS 作成するキーマテリアルを持つ対称カスタマーマネージドキーでのみサポートされます。
> - AWS KMS は AWS マネージドキー 毎年自動的にローテーションします。 AWS マネージドキーのキーローテーションを有効化または無効化することはできません。

詳細については、「AWS Key Management Service デベロッパーガイド」の「How key rotation works」を参照してください。

Amazon EBS 暗号化の例

暗号化された EBS リソースを作成すると、ボリューム作成パラメータまたは AMI やインスタンスのブロックデバイスマッピングで別の カスタマーマネージド型キー を指定しない限り、アカウントの EBS 暗号化のデフォルト KMS キー によって暗号化されます。

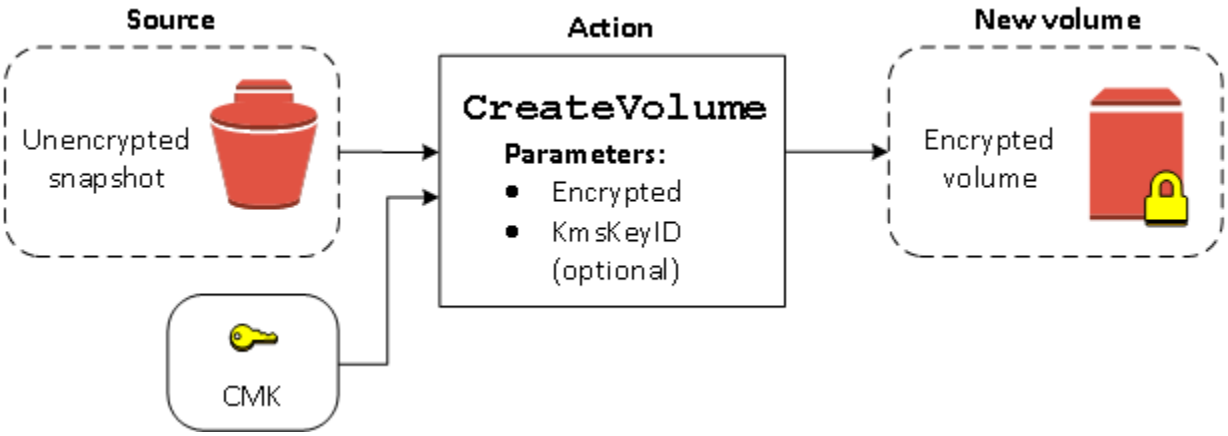
次の例では、ボリュームとスナップショットの暗号化状態を管理する方法を示します。暗号化のケースの完全なリストについては、暗号化の結果の表を参照してください。

例

- 暗号化されていないボリュームを復元する (デフォルトでの暗号化が有効になっていない場合)
- 暗号化されていないボリュームを復元する (デフォルトでの暗号化が有効になっている場合)
- 暗号化されていないスナップショットをコピーする (デフォルトでの暗号化が有効になっていない場合)
- 暗号化されていないスナップショットをコピーする (デフォルトでの暗号化が有効になっている場合)
- 暗号化ボリュームを再暗号化する
- 暗号化スナップショットを再暗号化する
- 暗号化されたボリュームと暗号化されていないボリュームとの間でデータを移行する
- 暗号化の結果

暗号化されていないボリュームを復元する (デフォルトでの暗号化が有効になっていない場合)

デフォルトでの暗号化を有効にしないと、暗号化されていないスナップショットから復元されたボリュームは、デフォルトで暗号化されません。ただし、`Encrypted` パラメータと、必要に応じて `KmsKeyId` パラメータを設定して、結果のボリュームを暗号化することができます。以下の図は、そのプロセスを示したものです。

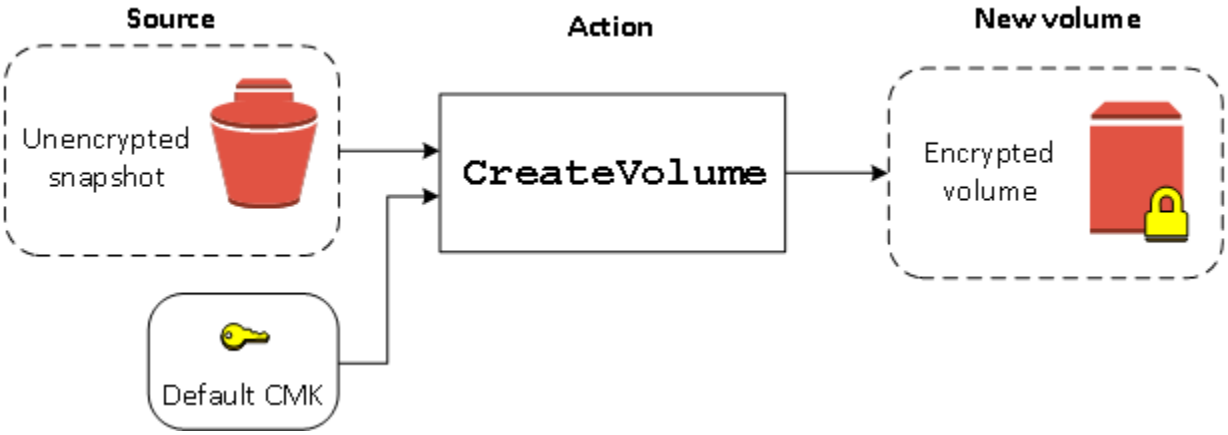


KmsKeyId パラメータを省略すると、結果のボリュームは EBS 暗号化のデフォルト KMS キー を使用して暗号化されます。ボリュームを別の KMS キー に暗号化するには、KMS キー ID を指定する必要があります。

詳細については、「Amazon EBS ボリュームの作成」を参照してください。

暗号化されていないボリュームを復元する (デフォルトでの暗号化が有効になっている場合)

デフォルトでの暗号化を有効にした場合、暗号化されていないスナップショットから復元されたボリュームには暗号化が必須であり、デフォルトの KMS キー を使用するために暗号化パラメータは必要ありません。以下の図に、このデフォルトの簡単なケースを示しています。

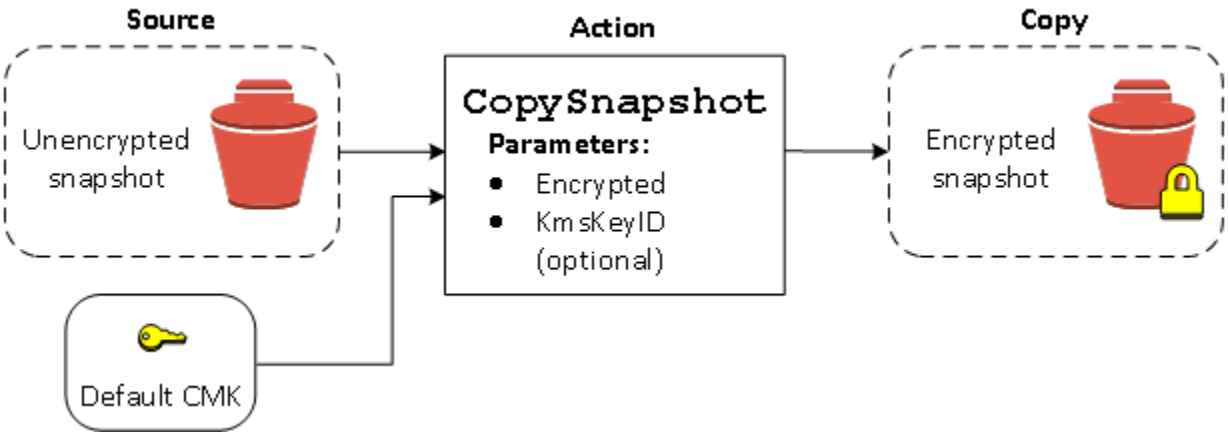


復元したボリュームを対称カスタマーマネージド型暗号化キーに暗号化する場合は、暗号化されていないボリュームを復元する (デフォルトでの暗号化が有効になっていない場合) に示すように Encrypted と KmsKeyId の両方のパラメータを指定する必要があります。

暗号化されていないスナップショットをコピーする (デフォルトでの暗号化が有効になっていない場合)

デフォルトでの暗号化を有効にしないと、暗号化されていないスナップショットのコピーは、デフォルトで暗号化されません。ただし、Encrypted パラメータと、必要に応じて KmsKeyId パラメータを設定して、結果のスナップショットを暗号化することができます。KmsKeyId を省略すると、結果のスナップショットはデフォルトの KMS キー に暗号化されます。ボリュームを別の対称暗号化 KMS キーに暗号化するには、KMS キー ID を指定する必要があります。

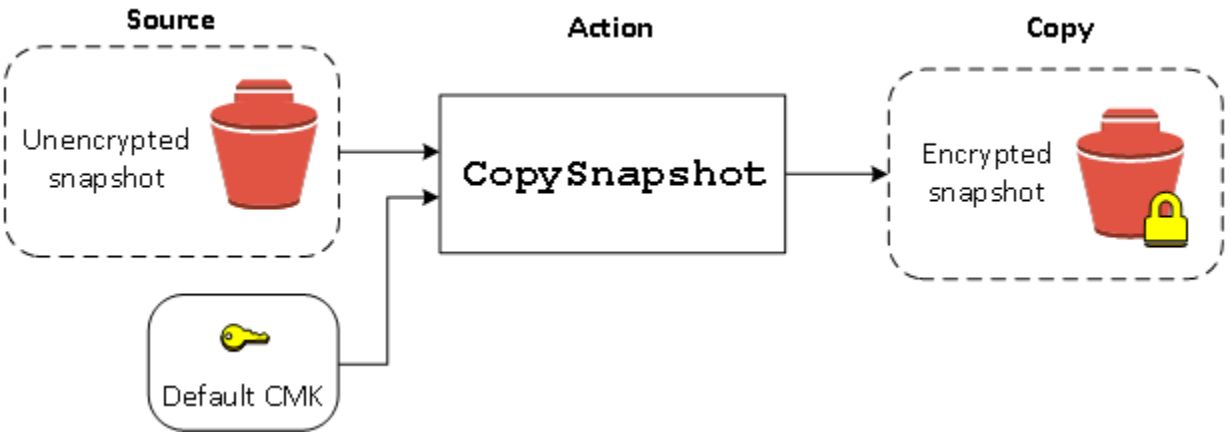
以下の図は、そのプロセスを示したものです。



EBS ボリュームを暗号化するには、暗号化されていないスナップショットを暗号化されたスナップショットにコピーし、その暗号化されたスナップショットからボリュームを作成することができます。詳細については、Amazon EBS スナップショットのコピーを参照してください。

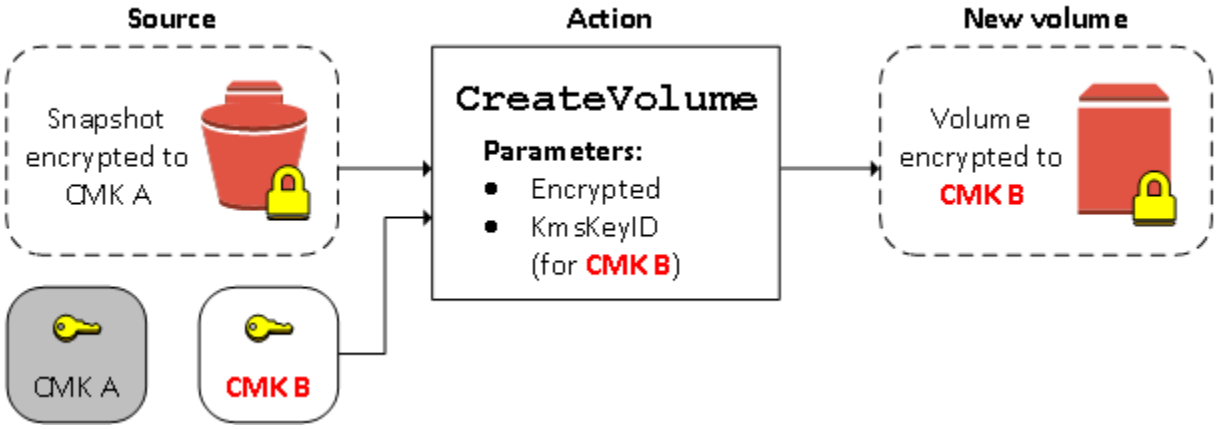
暗号化されていないスナップショットをコピーする (デフォルトでの暗号化が有効になっている場合)

デフォルトでの暗号化を有効にした場合、暗号化されていないスナップショットのコピーには暗号化が必須であり、デフォルトの KMS キー を使用する場合は、暗号化パラメータは必要ありません。このデフォルトのケースを次の図に示します。



暗号化ボリュームを再暗号化する

CreateVolume アクションが暗号化されたスナップショットに対して実行されるときは、別の KMS キー でそれを再暗号化することができます。以下の図は、そのプロセスを示したものです。この例では、KMS キー A と KMS キー B の 2 つの KMS キー を所有しています。ソーススナップショットは KMS キー A によって暗号化されています。ボリュームの作成中に、パラメータとして指定された KMS キー B の KMS キー ID を使用して、ソースデータは自動的に復号され、次に KMS キー B によって再暗号化されます。

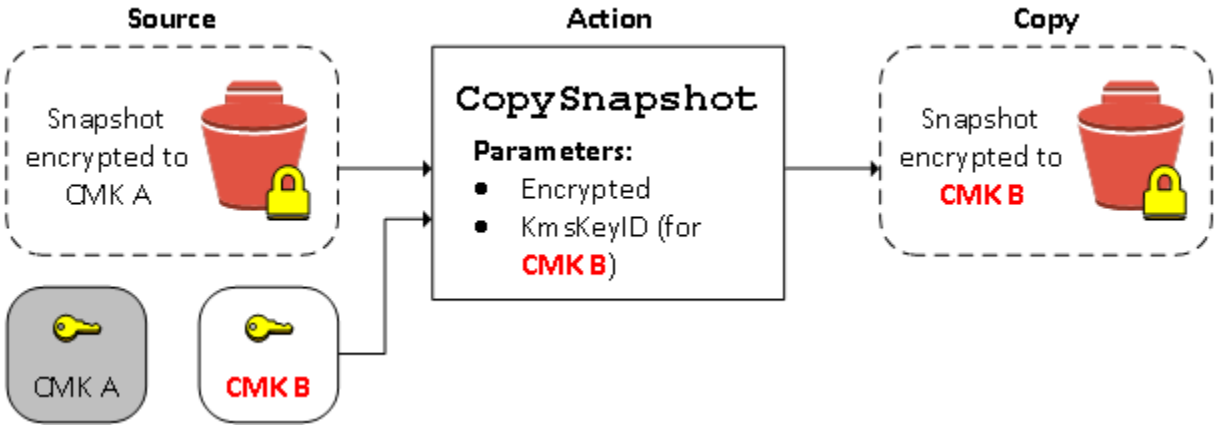


詳細については、「Amazon EBS ボリュームの作成」を参照してください。

暗号化スナップショットを再暗号化する

スナップショットをコピー時に暗号化する機能により、既に暗号化された自己所有のスナップショットに新しい対称暗号化 KMS キーを適用できます。結果として作成されたコピーから復元されたボリュームには、新しい KMS キー を使用してのみアクセスすることができます。以下の図は、そのプロセスを示したものです。この例では、KMS キー A と KMS キー B の 2 つの KMS キー を所有しています。ソーススナップショットは KMS キー A によって暗号化されています。コピー中に、パ

ラメータとして指定された KMS キー B の KMS キー ID を使用して、ソースデータは自動的に KMS キー B によって再暗号化されます。



関連するシナリオでは、共有されているスナップショットのコピーに新しい暗号化パラメータを適用するよう選択できます。デフォルトでは、コピーは、スナップショットの所有者によって共有された KMS キー を使用して暗号化されます。ただし、管理する別の KMS キー を使用して、共有スナップショットのコピーを作成することをお勧めします。これにより、元の KMS キー が侵害された場合や、所有者が何らかの理由で KMS キー を無効にした場合に、ボリュームへのアクセスが保護されます。詳細については、暗号化とスナップショットのコピーを参照してください。

暗号化されたボリュームと暗号化されていないボリュームとの間でデータを移行する

暗号化されているボリュームと暗号化されていないボリュームの両方に対してアクセス許可がある場合は、これらの間で自由にデータを転送できます。EC2 では、暗号化と復号化のオペレーションが透過的に実行されます。

Linux インスタンス

例えば、rsync コマンドを使用してデータをコピーします。次のコマンドでは、移行元のデータは `/mnt/source` にあり、移行先のボリュームは `/mnt/destination` にマウントされています。

```
[ec2-user ~]$ sudo rsync -avh --progress /mnt/source/ /mnt/destination/
```

Windows インスタンス

例えば、robocopy コマンドを使用してデータをコピーします。次のコマンドでは、移行元のデータは `D:\` にあり、移行先のボリュームは `E:\` にマウントされています。

```
PS C:\> robocopy D:\sourcefolder E:\destinationfolder /e /copyall /eta
```

非表示のフォルダで問題が発生することを回避するために、ボリューム全体をコピーするのではなく、フォルダを使用することをお勧めします。

暗号化の結果

次の表に、設定可能な組み合わせごとの暗号化の結果を示します。

EBS 暗号化が有効になっていますか?	デフォルトで暗号化が有効になっていますか?	ボリュームのソース	デフォルト (カスタマーマネージド型キーの指定なし)	カスタム (カスタマーマネージド型キーの指定あり)
いいえ	いいえ	新しい (空の) ボリューム	暗号化されていない	該当なし
いいえ	いいえ	所有する暗号化されていないスナップショット	暗号化されていない	
いいえ	いいえ	お客様が所有する暗号化されたスナップショット	同じキーで暗号化されている	
いいえ	いいえ	お客様と共有されている暗号化されていないスナップショット	暗号化されていない	
いいえ	いいえ	お客様と共有されている暗号化されたスナップショット	デフォルトのカスタマーマネージド型キーで暗号化*	
はい	いいえ	新しいボリューム	デフォルトのカスタマーマネージド型キーで暗号化	指定したカスタマーマネージド型キーにより暗号化**
はい	いいえ	所有する暗号化されていないスナップショット	デフォルトのカスタマーマネージド型キーで暗号化	

EBS 暗号化が有効になっていますか?	デフォルトで暗号化が有効になっていますか?	ボリュームのソース	デフォルト (カスタマーマネージド型キーの指定なし)	カスタム (カスタマーマネージド型キーの指定あり)
はい	いいえ	お客様が所有する暗号化されたスナップショット	同じキーで暗号化されている	
はい	いいえ	お客様と共有されている暗号化されていないスナップショット	デフォルトのカスタマーマネージド型キーで暗号化	
はい	いいえ	お客様と共有されている暗号化されたスナップショット	デフォルトのカスタマーマネージド型キーで暗号化	
いいえ	はい	新しい (空の) ボリューム	デフォルトのカスタマーマネージド型キーで暗号化	該当なし
いいえ	はい	所有する暗号化されていないスナップショット	デフォルトのカスタマーマネージド型キーで暗号化	
いいえ	はい	お客様が所有する暗号化されたスナップショット	同じキーで暗号化されている	
いいえ	はい	お客様と共有されている暗号化されていないスナップショット	デフォルトのカスタマーマネージド型キーで暗号化	
いいえ	はい	お客様と共有されている暗号化されたスナップショット	デフォルトのカスタマーマネージド型キーで暗号化	
はい	はい	新しいボリューム	デフォルトのカスタマーマネージド型キーで暗号化	指定したカスタマーマネージド型キーにより暗号化

EBS 暗号化が有効になっていますか?	デフォルトで暗号化が有効になっていますか?	ボリュームのソース	デフォルト (カスタマーマネージド型キーの指定なし)	カスタム (カスタマーマネージド型キーの指定あり)
はい	はい	所有する暗号化されていないスナップショット	デフォルトのカスタマーマネージド型キーで暗号化	
はい	はい	お客様が所有する暗号化されたスナップショット	同じキーで暗号化されている	
はい	はい	お客様と共有されている暗号化されていないスナップショット	デフォルトのカスタマーマネージド型キーで暗号化	
はい	はい	お客様と共有されている暗号化されたスナップショット	デフォルトのカスタマーマネージド型キーで暗号化	

* これは、 AWS アカウントとリージョンの EBS 暗号化に使用されるデフォルトのカスタマーマネージドキーです。デフォルトでは、これは EBS AWS マネージドキー に固有のものです。または、カスタマーマネージドキーを指定できます。

** これは、ボリュームの起動時に指定されたカスタマーマネージド型キーです。このカスタマーマネージドキーは、 AWS アカウントとリージョンのデフォルトのカスタマーマネージドキーの代わりに使用されます。

Amazon EBS ボリュームパフォーマンス

Amazon EBS のパフォーマンスは、I/O 特性やインスタンスとボリュームの設定などを含むいくつかの要因に左右されます。Amazon EBS 製品および Amazon EC2 製品の詳細ページに記載されているガイダンスに従うと、通常は良好なパフォーマンスを実現することができます。ただし、ピークパフォーマンスを実現するには、多少のチューニングを行う必要な場合があります。最適な設定を決定するには、ベンチマーキングに加えて、実際のワークロードからの情報でパフォーマンスをチューニングすることをお勧めします。EBS ボリュームの基本操作について理解したら、必要とする I/O パフォーマンスと、Amazon EBS パフォーマンスを向上させるオプションを確認し、そのパフォーマンス要件に対応できるようにすることをお勧めします。

AWS EBS ボリュームタイプのパフォーマンスの更新は、既存のボリュームにすぐには反映されない場合があります。古いボリュームで完全なパフォーマンスを確認するためには、最初に `ModifyVolume` アクションの実行が必要になる場合があります。詳細については、「Elastic Volumes オペレーションを使用して Amazon EBS ボリュームを変更する」を参照してください。

内容

- Amazon EBS パフォーマンスのヒント
- Amazon EBS 最適化
- Amazon EBS ボリュームの初期化
- 設定可能なインスタンス帯域幅の重み付け
- Amazon EBS I/O の特性およびモニタリング
- Amazon EBS および RAID の構成
- Amazon EBS ボリュームのベンチマーク

Amazon EBS パフォーマンスのヒント

ここに示すヒントは、さまざまなユーザーシナリオで、EBS ボリュームから最適なパフォーマンスを得るためのベストプラクティスを表しています。

EBS 最適化インスタンスを使用する

EBS 最適化スループットがサポートされていないインスタンスでは、インスタンスと EBS ボリュームの間のトラフィックが、ネットワークトラフィックと競合する場合があります。EBS 最適化インスタンスでは、これら 2 種類のトラフィックを分離した状態が維持されます。EBS 最適化インスタ

ンスの設定によっては、追加コストが発生する場合 (C3、R3、M3 など) と、追加コストなしで常に EBS 最適化状態になる場合 (M4、C4、C5、D2 など) があります。詳細については、「Amazon EBS 最適化」を参照してください。

インスタンス帯域幅を設定する

サポートされているインスタンスタイプでは、ebs-1 の帯域幅の重み付けを使用して、Amazon EBS の帯域幅を 25% 増やすようにインスタンス帯域幅の重み付けを設定できます。この機能を使用して EBS と VPC ネットワーク間のインスタンスのネットワークリソース割り当てを最適化することで、I/O 集約型ワークロードの EBS パフォーマンスを改善できる可能性があります。詳細については、「設定可能なインスタンス帯域幅の重み付け」を参照してください。

パフォーマンスの計算方法を理解する

EBS ボリュームのパフォーマンスを測定する場合、関連する測定単位と、パフォーマンスの計算方法を理解することが重要です。詳細については、Amazon EBS I/O の特性およびモニタリングを参照してください。

ワークロードを理解する

EBS ボリュームの最大パフォーマンス、I/O 操作の数およびサイズ、各アクションが完了するまでの所要時間は、互いに関連しています。これらの各要因 (パフォーマンス、I/O、レイテンシー) は相互に影響を与えます。また、アプリケーションが異なると、影響を受ける要因もさまざまに異なります。詳細については、「Amazon EBS ボリュームのベンチマーク」を参照してください。

スナップショットからボリュームを初期化する際のパフォーマンス低下に注意する

スナップショットから作成された新しい EBS ボリュームの各データブロックに初めてアクセスするときには、レイテンシーが著しく増加します。このパフォーマンスヒットは、以下のいずれかの方法で回避できます。

- 各ブロックへのアクセスが、ボリュームの本番環境への移行前に起こるようにする。このプロセスは、初期化と呼ばれます (以前は事前ウォーミングと呼ばれていました)。詳細については、「作成後にボリュームを手動で初期化する」を参照してください。
- スナップショットで高速スナップショット復元を有効にして、スナップショットから作成された EBS ボリュームが作成時に完全に初期化され、プロビジョニングされたすべてのパフォーマンスが即座に提供されるようにします。詳細については、「Amazon EBS 高速スナップショット復元」を参照してください。

HDD パフォーマンスが低下する要因

スループット最適化 HDD (st1) または Cold HDD (sc1) ボリュームのスナップショットを作成すると、スナップショットの進行中はボリュームのベースライン値までパフォーマンスが低下する可能性があります。この動作は、これらのボリュームタイプに固有です。パフォーマンスが制限される他の要因としては、インスタンスでのサポート範囲を超えるスループットの強要、スナップショットから作成したボリュームの初期化中のパフォーマンス低下、ボリュームに対する大量の小さなランダム I/O などがあります。HDD ボリュームのスループットを計算する方法については、Amazon EBS ボリュームの種類を参照してください。

アプリケーションから送られる I/O リクエスト数が十分でない場合も、パフォーマンスに影響します。これは、ボリュームのキュー長や I/O サイズを確認することで監視できます。このキュー長とは、アプリケーションからボリュームへの I/O リクエストのうち処理待ちのものの数です。最大限の安定性を確保するために、HDD-Backed ボリュームで 1 MiB のシーケンシャル I/O を実行する際には、キュー長 (整数に四捨五入) を 4 以上に保つ必要があります。安定したパフォーマンスの確保については、Amazon EBS I/O の特性およびモニタリングを参照してください。

st1 および sc1 (Linux インスタンスインスタンスのみ) で高いスループットの読み取りが多いワークロードに先読みを増やす

一部のワークロードでは読み取りが多く、オペレーティングシステムのページキャッシュを通じて (例えば、ファイルシステムから) ブロックデバイスへのアクセスが行われます。この場合、最大スループットを実現するには、先読みを 1 MiB に設定することをお勧めします。これは HDD ボリュームにのみ適用されるブロックデバイス単位の設定です。

ブロックデバイスに対する現在の先読み値を調べるには、次のコマンドを使用します。

```
$ sudo blockdev --report /dev/<device>
```

ブロックデバイス情報は次の形式で返されます。

```
RO    RA   SSZ   BSZ   StartSec            Size   Device
rw   256   512  4096       4096      8587820544   /dev/<device>
```

表示されているデバイスについては、先読み値として 256 (デフォルト値) が報告されています。この数値にセクターサイズ (512 バイト) を乗算すると、先読みバッファのサイズ (この場合は 128 KiB) を得ることができます。バッファ値を 1 MiBに設定するには、次のコマンドを使用します。

```
$ sudo blockdev --setra 2048 /dev/<device>
```

最初のコマンドをもう一度実行して、先読み設定が 2,048 になったことを確認します。

この設定は、ワークロードがサイズの大きなシーケンシャル I/O で構成される場合にのみ使用してください。ワークロードの内容として、サイズの小さなランダム I/O がほとんどであれば、この設定を使用すると逆にパフォーマンスが低下します。一般的に、サイズの小さい I/O やランダム I/O が大部分を占めるワークロードの場合は、st1 や sc1 ボリュームではなく、汎用 SSD (gp2 および gp3) ボリュームの使用を検討してください。

最新の Linux カーネルを使用する (Linux インスタンスのみ)

間接記述子がサポートされている最新の Linux カーネルを使用します。Linux カーネル 3.8 以降には、すべてこのサポートがあり、現行世代の EC2 インスタンスも同様です。平均 I/O サイズが 44 KiB 前後であれば、間接記述子がサポートされていないインスタンスやカーネルを使用している可能性があります。Amazon CloudWatch のメトリクスから平均 I/O サイズを得る方法については、Amazon EBS I/O の特性およびモニタリングを参照してください。

st1 または sc1 ボリュームで最大スループットを達成するには、256 の値を、xen_blkfront.max パラメータ (Linux カーネルバージョン 4.6 未満の場合)または xen_blkfront.max_indirect_segments パラメータ (Linux カーネルバージョン 4.6 以降の場合) に適用することを推奨します。適切なパラメータは、OS の起動コマンドラインで設定できます。

例えば、Amazon Linux AMI では、/boot/grub/menu.lst に記述されている GRUB 設定で kernel 行の末尾に追加できます。

```
kernel /boot/vmlinuz-4.4.5-15.26.amzn1.x86_64 root=LABEL=/ console=ttyS0
 xen_blkfront.max=256
```

後のカーネルの場合、コマンドは次のようになります。

```
kernel /boot/vmlinuz-4.9.20-11.31.amzn1.x86_64 root=LABEL=/ console=tty1 console=ttyS0
 xen_blkfront.max_indirect_segments=256
```

この設定を有効にするには、インスタンスを再起動する必要があります。

詳細については、「準仮想化 AMI 向けの GRUB の設定」を参照してください。他の Linux ディストリビューションでは (特に GRUB ブートローダーが使用されていない場合)、カーネル パラメータの調整に別のアプローチが必要になることがあります。

EBS I/O の特性の詳細については、このトピックのAmazon EBS: パフォーマンスを考慮した設計プレゼンテーションを参照してください。

RAID 0 を使用してインスタンスのリソース使用率を最大化する

一部のインスタンスタイプでは、単一の EBS ボリュームをプロビジョニングする場合よりも I/O スループットを増やすことができます。複数のボリュームを RAID 0 設定で結合し、これらのインスタンス用の利用可能な帯域幅を使用できます。詳細については、「Amazon EBS および RAID の構成」を参照してください。

Amazon EBS ボリュームのパフォーマンスをモニタリングする

Amazon CloudWatch、ステータスチェック、EBS の詳細なパフォーマンス統計を使用して、Amazon EBS ボリュームのパフォーマンスをモニタリングおよび分析できます。詳細については、「Amazon EBS の Amazon CloudWatch メトリクス」および「Amazon EBS の詳細なパフォーマンス統計」を参照してください。

Amazon EBS 最適化

Amazon EBS 最適化インスタンスは、最適化された設定スタックを使用し、Amazon EBS I/O 用に専用のキャパシティを追加で提供します。このように最適化することで、Amazon EBS I/O と、インスタンスからのその他のトラフィックとの間の競合を最小に抑え、EBS ボリュームの最高のパフォーマンスを実現します。

EBS 最適化インスタンスは、Amazon EBS 用に専用の帯域幅を用意します。EBS 最適化インスタンスにアタッチした場合:

- io2 Block Express ボリュームは、16KiB の I/O オペレーションで平均 500 マイクロ秒未満のレイテンシーを実現するように設計されています。io2Block Express ボリュームでは汎用ボリュームと比較して外れ値のレイテンシーも改善されるため、I/O が 800 マイクロ秒を超える頻度が 10 分の 1 以下に減少します。io1 および io2 ボリュームは、1 年の 99.9% の期間、プロビジョンド IOPS パフォーマンスの少なくとも 90% のボリュームを提供するように設計されています。
- gp2 ボリュームと gp3 ボリュームは、1 年の 99% の期間、プロビジョンド IOPS パフォーマンスの少なくとも 90% のボリュームを提供するように設計されています。

- st1 および sc1 ボリュームは、1 年の 99% の期間、想定されるスループットパフォーマンスの少なくとも 90% のボリュームを提供します。

毎時間、予測合計スループットの 99% 達成を目標に、準拠しない期間はほぼ均一に分散されています。詳細については、「Amazon EBS ボリュームの種類」を参照してください。

ワークロードの I/O サイズは、I/O サイズの増大に伴いレイテンシーが増加するにつれて、観測された平均レイテンシーに影響を及ぼします。例えば、io2 Block Express ボリュームは、16KiB の I/O 操作で平均 500 マイクロ秒未満のレイテンシーを実現するように設計されています。

詳細については、「Amazon EC2 ユーザーガイド」の「Amazon EBS 最適化インスタンス」を参照してください。

Amazon EBS ボリュームの初期化

EBS スナップショットまたは別の EBS ボリューム (ボリュームコピー) から Amazon EBS ボリュームを作成する場合、データブロックにアクセスする前に、データブロックをボリュームに書き込む必要があります。スナップショットから作成されたボリュームの場合は、データブロックを Amazon S3 から新しいボリュームにダウンロードする必要があります。ボリュームコピーの場合は、データブロックをソースボリュームからボリュームコピーにコピーする必要があります。このプロセスはボリューム初期化と呼ばれます。この間、初期化されるボリュームでは、I/O レイテンシーが増加し、パフォーマンスが低下する可能性があります。フルボリュームのパフォーマンスは、すべてのブロックがダウンロードされてボリュームに書き込まれた場合にのみ達成されます。

> **Note**
>
> 空のボリュームは、作成直後に最大パフォーマンスを実現し、初期化は必要ありません。

デフォルトのボリューム初期化レートは初期化プロセス全体で変動するため、完了時間が予測できなくなる可能性があります。ボリューム初期化に関連するパフォーマンスへの影響を最小限に抑えるため、次のオプションを使用できます。

トピック

- ボリューム初期化に Amazon EBS プロビジョンドレートを使用する
- 高速スナップショット復元が有効になっているスナップショットを使用する
- 作成後にボリュームを手動で初期化する

- Amazon EBS ボリューム初期化のステータスをモニタリングする

> **Note**
>
> ボリュームの初期化と高速スナップショット復元の Amazon EBS プロビジョンドレートは、ボリュームコピーではサポートされていません。詳細については、「Initialization」を参照してください。

ボリューム初期化に Amazon EBS プロビジョンドレートを使用する

スナップショットから Amazon EBS ボリュームを作成する場合、オプションで、100 ~ 300 MiB/秒の範囲のボリューム初期化の Amazon EBS プロビジョンドレート (ボリューム初期化レート) を指定できます。ボリューム初期化レートを指定すると、スナップショットブロックは Amazon S3 からダウンロードされ、作成後に指定されたレートでボリュームに書き込まれます。これにより、予測可能な時間内に完全に初期化され、完全にパフォーマンスを発揮するボリュームを作成できます。

ボリューム初期化レートは、複数のボリュームを同時に作成していて、すべてのボリュームを予測可能な時間内に初期化する必要がある場合に特に便利です。

> **Note**
>
> ボリューム初期化の Amazon EBS プロビジョンドレートは、すべての Amazon EBS ボリュームタイプ、および Amazon EC2 Mac インスタンスを含むすべての Amazon EC2 インスタンスタイプでサポートされています。

ボリューム初期化レートは以下の場合に指定できます。

- 個々のボリューム作成リクエストの場合
- インスタンス起動リクエストの EBS ボリュームブロックデバイスマッピングの場合
- 起動テンプレートの EBS ボリュームブロックデバイスマッピングの場合
- ルートボリューム置換タスクによって作成された EBS ボリュームの場合
- Amazon EKS クラスター (EBS CSI ドライバーによって作成) および Amazon ECS クラスターの EBS ボリュームの場合

トピック

- 仕組み
- 考慮事項
- クォータ
- 料金
- ボリュームの初期化レートを指定する

仕組み

ボリューム初期化レートでボリュームを作成すると、スナップショットブロックは指定したレートで Amazon S3 からボリュームにダウンロードされます。

ボリューム初期化にかかる時間は、以下によって異なります。

- スナップショットのデータサイズ (作成されるボリュームのサイズではありません)。

> **Tip**
>
> スナップショットのデータサイズを確認するには、describe-snapshots コマンド出力の `FullSnapshotSizeInBytes` フィールド、またはコンソールの [フルスナップショットサイズ] フィールドを確認します。

- 指定したボリューム初期化レート

例えば、10 GiB のデータを持つスナップショットを使用して 20 GiB ボリュームを作成し、300 MiB/秒のボリューム初期化レートを指定すると、ボリュームは約 34.1 秒 (10 GiB / 300 MiB/秒 = 34.1 秒) で完全に初期化されます。同様に、同じスナップショットとボリューム初期化レートで同時に 10 個のボリュームを作成すると、10 個のボリュームすべてが 34.1 秒で完全に初期化されます。

考慮事項

- ボリューム初期化レートは 100 ~ 300 MiB/秒の間で指定できます。
- ボリューム初期化レートを指定する場合、料金と完了時間は、スナップショットデータのサイズ (ボリュームのサイズではありません) と指定したレートによって決まります。詳細については、「料金」を参照してください。
- Amazon EBS は、お客様が 99% の時間用に指定したボリューム初期化レートの 10% 以内の平均レートを提供します。

- ボリューム初期化レートを指定し、高速スナップショット復元が有効になっているスナップショットを使用した場合、Amazon EBS では高速スナップショット復元ではなく指定されたレートが使用されます。高速スナップショット復元を使用したい場合は、ボリューム初期化レートを指定しないでください。
- 容量の制約またはクォータの超過により、Amazon EBS が指定されたボリューム初期化レートでボリュームを初期化できない場合、リクエストは失敗します。
- ローカルゾーン AWS Outpostsまたは Wavelength Zones で作成されたボリュームのボリューム初期化レートを指定することはできません。

クォータ

同時ボリューム作成リクエスト間でリクエストできる累積ボリューム初期化レートには、5,000 MiB/秒の制限があります。例えば、100 MiB/秒のレートで 50 件の同時ボリューム作成リクエスト (50 件の同時リクエスト x 100 MiB/秒のレート)、または 200 MiB/秒のレートで 25 件の同時リクエスト (25 件の同時リクエスト x 200 MiB/秒のレート) を実行できます。この制限はリージョンごとに適用されます。リクエストがこの制限を超えると、リクエストは失敗します。その場合は進行中のリクエストの一部が完了するまで待つか、クォータの引き上げをリクエストします。詳細については、「Amazon EBS のクォータ」を参照してください。

料金

ボリュームの初期化レートでボリュームを作成すると、選択した初期化階層に基づいて GiB あたりの料金が課金されます。レートはリージョンによって異なります。選択した初期化階層のレートを確認するには、Amazon EBS 料金を開き、リージョンを選択して、ボリューム初期化のプロビジョンドレートセクションを見つけます。

ボリュームのサイズではなく、スナップショットの完全なデータサイズに基づいて課金されます。例えば、50 GiB の書き込まれたデータを含む 100 GiB ボリュームのスナップショットを作成する場合、スナップショットの全データサイズは 50 GiB です。そのスナップショットを使用してボリュームを作成し、ボリュームの初期化レートを指定する場合、料金は 50 GiB に基づきます。

> **Tip**
>
> 完全なスナップショットデータサイズを確認するには、describe-snapshots コマンド出力の `FullSnapshotSizeInBytes`フィールド、またはコンソールのフルスナップショットサイズフィールドを確認します。

計算式は次のとおりです。

```
Rate for the selected initialization tier in the Region × full snapshot data size
```

ボリュームが `active` 状態になるとすぐに、全額が請求されます。失敗したリクエストについては請求されません。

初期化が完了する前にボリュームを削除しても、リクエストされたボリューム初期化レートに対して課金されます。

ボリュームの初期化レートを指定する

次のいずれかの方法を使用して、ボリュームの初期化レートでスナップショットからボリュームを作成します。

Console

ボリュームを作成するには

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. ナビゲーションペインで、ボリュームを選択し、ボリュームの作成を選択します。
3. [Volume type] (ボリュームタイプ) に、作成するボリュームのタイプを選択します。
4. サイズ には、ソーススナップショットのボリュームサイズ以上のサイズを入力します。
5. [アベイラビリティーゾーン] では、ボリュームを作成するアベイラビリティーゾーンを選択します。
6. スナップショット ID で、使用するスナップショットを選択します。
7. ボリュームの初期化レートには、100-300 MiB/秒のレートを入力します。レートを指定すると、追加料金が発生します。デフォルトの初期化レートまたは高速スナップショット復元を使用するには、選択したスナップショットに対して有効になっている場合は、このフィールドを空のままにします。

> **Note**
>
> レートと完全なスナップショットデータサイズが初期化時間にどのように影響するかについては、「」を参照してください仕組み。料金情報については、料金 を参照してください。

8. その他の必要なボリューム設定を設定し、必要に応じてタグを追加します。
9. [Create volume] (ボリュームの作成) を選択します。
10. ボリュームグリッドで、ボリュームの初期化状態と進行状況を表示します。

AWS CLI

ボリュームを作成するには

create-volume コマンドを使用します。次の の例では、gp3ボリュームの初期化レートが 300 MiB/秒の指定されたアベイラビリティーゾーンのスナップショットからボリュームを作成します。リージョン、アベイラビリティーゾーン、スナップショット ID の例をリソースの値に置き換えます。

```
aws ec2 create-volume \
    --region us-east-1 \
    --availability-zone us-east-1a \
    --volume-type gp3 \
    --snapshot-id snap-0abcdef1234567890 \
    --volume-initialization-rate 300
```

を省略すると--size、ボリュームはソーススナップショットのボリュームサイズを使用します。を指定する場合--size、値はソーススナップショットのボリュームサイズ以上である必要があります。

PowerShell

ボリュームを作成するには

New-EC2Volume コマンドレットを使用します。次の の例では、gp3ボリュームの初期化レートが 300 MiB/秒の指定されたアベイラビリティーゾーンのスナップショットからボリュームを作成します。コマンドは、返されたボリュームオブジェクトを に保存します$volume。リージョン、アベイラビリティーゾーン、スナップショット ID の例をリソースの値に置き換えます。

```
$volume = New-EC2Volume `
    -Region us-east-1 `
    -AvailabilityZone us-east-1a `
    -VolumeType gp3 `
    -SnapshotId snap-0abcdef1234567890 `
    -VolumeInitializationRate 300
```

モニタリングの初期化

ボリュームの初期化情報の更新には最大 5 分かかる場合があります。初期化をモニタリングする手順については、「」を参照してくださいAmazon EBS ボリューム初期化のステータスをモニタリングする。

高速スナップショット復元が有効になっているスナップショットを使用する

高速スナップショット復元が有効になっているスナップショットからボリュームを作成すると、ボリュームは作成時に完全に初期化され、すぐに完全なパフォーマンスを提供します。高速スナップショット復元の使用方法の詳細については、「Amazon EBS 高速スナップショット復元」を参照してください。

作成後にボリュームを手動で初期化する

作成後に Amazon EBS ボリュームを手動で初期化することで、ボリューム初期化によるパフォーマンスへの影響を最小限に抑えることができます。

次の手順を使用して、作成後に Amazon EBS ボリュームを手動で初期化できます。

> **Important**
>
> スナップショットから作成された Provisioned IOPS SSD ボリュームを初期化している間は、ボリュームのパフォーマンスが想定レベルの 50% を下回る場合があります。このため、ボリュームは [I/O Performance (I/O 性能)] ステータスチェックで `warning` 状態が表示されます。これは想定の動作です。初期化中の Provisioned IOPS SSD ボリュームの `warning` 状態は無視してかまいません。詳細については、「Amazon EBS ボリュームステータスチェック」を参照してください。

Linux インスタンス

Linux で、スナップショットから作成されたボリュームを初期化するには

1. 新しく復元されたボリュームを Linux インスタンスにアタッチします。
2. インスタンスのブロックデバイスを一覧表示するには、lsblk コマンドを使用します。

```
$ lsblk
```

```
NAME  MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvdf  202:80   0  30G  0 disk
xvda1 202:1    0   8G  0 disk /
```

ここでは、新しいボリューム/dev/xvdf がアタッチされていますが、マウントされていないことがわかります (MOUNTPOINT 列の下にリストされているパスがないため)。

3. デバイスのすべてのブロックを読み取るには、dd ユーティリティまたは fio ユーティリティを使用します。Linux システムにデフォルトでインストールされているのは dd コマンドですが、マルチスレッドの読み取りが可能な fio の方が、処理速度が大幅に速くなります。

> **Note**
>
> このステップは、使用している EC2 インスタンスの帯域幅、ボリュームに対してプロビジョニングされている IOPS、そしてボリュームのサイズに応じて、数分から数時間かかることがあります。

[dd] `if` (入力ファイル) パラメータは、初期化するドライブに設定します。`of` (出力ファイル) パラメータは、Linux ヌル仮想デバイス `/dev/null` に設定する必要があります。`bs` パラメータは、読み取り操作のブロックサイズに設定します。最適なパフォーマンスを得るには、この値を 1 MB に設定します。

> **Important**
>
> dd を誤って使用すると、ボリュームのデータが失われる場合があります。以下のコマンド例に正確に従ってください。`if=/dev/`*`xvdf`* パラメータのみ、読み出しているデバイスの名前によって異なります。

```
$ sudo dd if=/dev/xvdf of=/dev/null bs=1M
```

[fio] システムに fio がインストールされている場合、ボリュームを初期化するには次のコマンドを使用します。`--filename` (入力ファイル) パラメータは、初期化するドライブに設定します。

```
$ sudo fio --filename=/dev/xvdf --rw=read --bs=1M --iodepth=32 --ioengine=libaio --
direct=1 --name=volume-initialize
```

Amazon Linux に fio をインストールするには、次のコマンドを使用します。

```
sudo yum install -y fio
```

Ubuntu に fio インストールするには、次のコマンドを使用します。

```
sudo apt-get install -y fio
```

この操作が終了すると、読み取り操作のレポートが表示されます。これでボリュームを使用する準備ができました。詳細については、「Amazon EBS ボリュームを使用できるようにする」を参照してください。

Windows インスタンス

どちらのツールを使用する前も、次のようにシステム上のディスクについての情報を収集してください。

システムディスクに関する情報を収集するには

1. システムで使用可能なディスクを一覧表示するには、wmic コマンドを使用します。

   ```
   wmic diskdrive get size,deviceid
   ```

   出力例を次に示します。

   ```
   DeviceID            Size
   \\.\PHYSICALDRIVE2  80517265920
   \\.\PHYSICALDRIVE1  80517265920
   \\.\PHYSICALDRIVE0  128849011200
   \\.\PHYSICALDRIVE3  107372805120
   ```

2. dd または fio を使用して、初期化するディスクを識別します。C: ドライブは、\\.\PHYSICALDRIVE0 にあります。使用するドライブ番号がはっきりしない場合は、diskmgmt.msc ユーティリティを使用して、ドライブ文字とディスクドライブ番号を比較できます。

Use the dd utility

dd をインストールおよび使用してボリュームを初期化するには、次の手順を実行します。

重要な考慮事項

- ボリュームの初期化には、使用している EC2 インスタンスの帯域幅、ボリュームに対してプロビジョニングされている IOPS、そしてボリュームのサイズに応じて、数分から数時間かかることがあります。
- dd を誤って使用すると、ボリュームのデータが失われる場合があります。この手順を正確に実行してください。

Windows 向け dd をインストールするには

Windows 用 dd プログラムは、Linux や Unix システムで共通して使用できる dd プログラムと同様の環境を提供します。このプログラムを使用すると、スナップショットから作成された Amazon EBS ボリュームを初期化することができます。最新のベータバージョンでは、`/dev/null` 仮想デバイスをサポートしています。以前のバージョンをインストールする場合は、代わりに `nul` 仮想デバイスを使用できます。詳細なドキュメントは、http://www.chrysocome.net/dd から入手できます。

1. Windows 用の最新バイナリバージョンの dd を、http://www.chrysocome.net/dd からダウンロードします。
2. (オプション) 簡単に覚えられる場所に、コマンドラインユーティリティ用のフォルダを作成します (`C:\bin` など)。コマンドラインユーティリティ用にフォルダを既に指定した場合、次の手順を実行する代わりに、そのフォルダを使用できます。
3. バイナリパッケージを解凍し、`dd.exe` ファイルをコマンドラインユーティリティのフォルダ (`C:\bin` など) にコピーします。
4. どこからでもフォルダ内のプログラムを実行できるようにするため、Path 環境変数にコマンドラインユーティリティのフォルダを追加します。

    a. [スタート] を選択し、[PC] のコンテキスト (右クリック) メニューを開いて、[プロパティ] を選択します。

    b. [システムの詳細設定]、[環境変数] の順に選択します。

    c. [システム環境変数] で [Path] 変数を選択し、[編集] を選択します。

    d. [変数値] として、既存の値の最後に、セミコロンとコマンドラインユーティリティフォルダの場所 (**`;C:\bin\`**) を追加します。

e. [OK] をクリックして、[システム変数の編集] ウィンドウを閉じます。

5. 新しいコマンドプロンプトウィンドウを開きます。前の手順を行っても、現在開いているコマンドプロンプトウィンドウの環境変数は更新されません。前の手順を完了した時点で開くコマンドプロンプトウィンドウが更新されます。

Windows 用 dd を使ってボリュームを初期化するには

指定したデバイスのすべてのブロックの読み取り (および /dev/null 仮想デバイスへの出力の送信) を実行するには、次のコマンドを実行します。このコマンドは、既存のデータを安全に初期化します。

```
dd if=\\.\PHYSICALDRIVEn of=/dev/null bs=1M --progress --size
```

dd がボリュームの末尾を超えて読み取りを行おうとすると、エラーが表示されることがあります。このエラーを無視しても問題ありません。

以前のバージョンの dd コマンドを使用した場合、そのコマンドは /dev/null デバイスをサポートしていません。代わりに、次のように nul デバイスを使用することができます。

```
dd if=\\.\PHYSICALDRIVEn of=nul bs=1M --progress --size
```

Use the fio utility

fio をインストールおよび使用してボリュームを初期化するには、次の手順を実行します。

Windows 用 fio をインストールするには

Windows 用 fio プログラムは、Linux や Unix システムで共通して使用できる fio プログラムと同様の環境を提供します。このプログラムを使用すると、スナップショットから作成された Amazon EBS ボリュームを初期化することができます。詳細については、https://github.com/axboe/fioを参照してください。

1. fio MSI インストーラをダウンロードするには、最新リリースの [アセット] を展開して、MSI インストーラを選択します。
2. fio をインストールします。

Windows 用 fio を使用してボリュームを初期化するには

1. 次のようなコマンドを実行してボリュームを初期化します。

```
fio --filename=\\.\PHYSICALDRIVEn  --rw=read --bs=1M --iodepth=32 --direct=1 --
name=volume-initialize
```

2. この操作が終了すると、新規ボリュームを使用する準備が完了します。詳細については、「Amazon EBS ボリュームを使用できるようにする」を参照してください。

Amazon EBS ボリューム初期化のステータスをモニタリングする

スナップショットまたは別のボリューム (ボリュームコピー) からボリュームを作成するときに、ボリューム初期化のステータスをモニタリングして、初期化プロセスが完了したかどうかを判断できます。ボリューム初期化は、次のオプションを使用してモニタリングできます。

トピック

- AWS CLI、 AWS Tools for PowerShell、および Amazon EC2 コンソール
- Amazon EventBridge

AWS CLI、 AWS Tools for PowerShell、および Amazon EC2 コンソール

ボリュームの作成後はいつでも AWS CLI、、 AWS Tools for PowerShell、および Amazon EC2 コンソールを使用して、ボリュームの初期化のステータスを確認できます。以下の情報が提供されます。

- 初期化タイプ (AWS CLI および AWS Tools for PowerShellのみ) — 使用されるボリューム初期化のタイプを示します。高速スナップショット復元とデフォルトのボリューム初期化`default`には 、ボリューム初期化`provisioned-rate`には Amazon EBS プロビジョンドレート、ボリュームコピーの初期化`volume-copy`には 。
- 完了までの推定時間 (AWS CLI および の AWS ツールPowerShellのみ) — ボリュームの初期化に Amazon EBS プロビジョンドレートを使用して作成されたボリュームの場合のみ。ボリューム初期化が完了するまでの推定残り時間を秒単位で表します。
- 進行状況 — ボリュームの初期化プロセスの進行状況のパーセンテージ (0 ~ 100)。高速スナップショット復元で初期化されたボリュームの場合、進行状況は作成直後に 100% に達します。
- [初期化の状態] — ボリューム初期化の全体的な状態 (`initializing` または `completed`)。高速スナップショット復元で初期化されたボリュームの場合、状態は作成直後に `completed` に移行します。

> **Note**
>
> ボリューム初期化の情報が更新されるまでに最大 5 分かかる場合があります。

次のタブを使用して、ボリュームのボリューム初期化ステータスを確認します。

Console

ボリューム初期化のステータスをモニタリングするには

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. ナビゲーションペインの [ボリューム] を選択します。
3. ボリューム初期化のステータスを確認するボリュームを選択します。
4. グリッドの [初期化の状態] フィールドと [詳細] タブには、初期化状態 (進行状況のパーセンテージ) の形式で進行状況情報が表示されます。例えば、初期化中 (75%) のように表示されます。

   初期化状態は [初期化中] と [完了] のいずれかになります。

AWS CLI

ボリューム初期化のステータスをモニタリングするには

describedescribe-volume-status AWS CLI コマンドを使用して、初期化ステータスを表示します。出力には、ボリューム初期化の Amazon EBS プロビジョンドレートで作成されたボリューム`EstimatedTimeToCompleteInSeconds`のみが含まれます。

例えば、次のコマンドでは、ボリューム初期化の Amazon EBS プロビジョンドレートで作成されたボリューム vol-1111111111111111 の初期化ステータスをチェックします。

```
aws ec2 describe-volume-status --volume-ids vol-1111111111111111
```

以下は出力の例です。

```
{
    "VolumeStatuses": [
        {
            "Actions": [],
```

```
            "AvailabilityZone": "us-east-1a",
            "Events": [],
            "VolumeId": "vol-1111111111111111",
            "VolumeStatus": {
                "Details": [
                    {
                        "Name": "io-enabled",
                        "Status": "passed"
                    },
                    {
                        "Name": "io-performance",
                        "Status": "not-applicable"
                    },
                    {
                        "Name": "initialization-state",
                        "Status": "completed"
                    }
                ],
                "Status": "ok"
            },
            "InitializationStatusDetails": {
                "InitializationType": "provisioned-rate",
                "Progress": 75,
                "EstimatedTimeToCompleteInSeconds": 850
            }
        }
    ]
}
```

PowerShell

ボリューム初期化のステータスをモニタリングするには

Get-EC2VolumeStatus コマンドレットを使用して、初期化ステータスを表示します。出力には、ボリューム初期化の Amazon EBS プロビジョンドレートで作成されたボリュームEstimatedTimeToCompleteInSecondsのみが含まれます。

例えば、次のコマンドでは、ボリューム初期化の Amazon EBS プロビジョンドレートで作成されたボリューム vol-1111111111111111 の初期化ステータスをチェックします。

```
(Get-EC2VolumeStatus `
    -Region us-east-1 `
    -VolumeId vol-1111111111111111).InitializationStatusDetails |
```

```
    Select-Object `
        InitializationType,
        Progress,
        EstimatedTimeToCompleteInSeconds
```

Amazon EventBridge

Amazon EventBridge イベントは、ボリューム初期化の完了後 5 分以内にアカウントに送信されます。これらのイベントに応答してプログラムによるアクションをトリガーするルールを作成できます。

> **Note**
>
> - イベントは、ベストエフォートベースで出力されます。
> - 初期化が完了する前にボリュームを削除するか、初期化が完了してから 5 分以内にボリュームを削除すると、イベントを受信しない可能性があります。

イベントの詳細については、「EBS ボリューム初期化イベント」を参照してください。

EventBridge を使用してボリューム初期化のステータスをモニタリングするには

1. Amazon EventBridge コンソールの https://console.aws.amazon.com/events/ を開いてください。
2. [ルール]、[ルールの作成] の順に選択します。
3. [ステップ 1] で、以下を行います。
   a. ルールの名前と説明を指定します。
   b. [イベントバス] で、イベントを受信するバスを選択します。カスタムイベントバスを作成していない場合は、デフォルトのままにするか、「Creating an event bus in Amazon EventBridge」を参照してください。
   c. [ルールタイプ] で、[イベントパターンを持つルール] を選択したままにします。
   d. [次へ] を選択します。
4. [ステップ 2] で、以下を行います。
   a. [イベントソース] で、[AWS イベントまたは EventBridge パートナーイベント] を選択します。

b. [作成方法] セクションで、[カスタムパターン (JSON エディタ)] を選択します。

c. [イベントパターン] で、以下を追加します。

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

イベント例については、「EBS ボリューム初期化イベント」を参照してください。

d. [次へ] を選択します。

5. [ステップ 3] で、以下を行います。

a. [ターゲットタイプ] では、[AWS サービス] を選択します。

b. [ターゲットの選択] で [SNS トピック] を選択し、[トピック] で必要なトピックを選択します。トピックを作成していない場合は、「Creating an Amazon SNS topic」を参照してください。

c. [許可] で、[実行ロールを使用 (推奨)] を選択したままにします。

d. [実行ロール] で、[この特定のリソースについて新しいロールを作成] を選択したままにし、デフォルトのロール名のままにします。

e. [次へ] を選択します。

6. [ステップ 4] で、必要に応じてルールのタグを指定し、[次へ] を選択します。

7. [ステップ 5] でルールを確認し、[ルールの作成] を選択します。

設定可能なインスタンス帯域幅の重み付け

インスタンス帯域幅設定 (IBC) は、Amazon EC2 インスタンスの Amazon EBS と VPC ネットワーク間のネットワーク帯域幅の割り当てを調整できる機能です。この機能は、特定の帯域幅要件を持つワークロードのパフォーマンスを最適化するのに役立ちます。インスタンス帯域幅設定は、一部のインスタンスでのみサポートされています。詳細については、「EC2 instance bandwidth weighting configuration」を参照してください。

EBS パフォーマンスの場合、ebs-1 帯域幅の重み付けを使用すると、ベースライン EBS 帯域幅が 25% 増加し、VPC ネットワーク帯域幅は同じ絶対量だけ減少します。これは、より高い EBS スループットを必要とする I/O 集約型ワークロードにとって有益です。

ワークロードを計画するときは、I/O のサイズとパターンを慎重に検討してください。一般に I/O サイズが小さいほど帯域幅制限の影響を受けにくくなるのに対し、I/O サイズやシーケンシャルワークロードが大きいほど、帯域幅の変化による影響が大きくなる可能性があります。選択した帯域幅の重み付けで最適なパフォーマンスを確保するために、特定のワークロードを徹底的にテストすることが重要です。

考慮事項

- 設定可能なインスタンス帯域幅は、一部のインスタンスタイプでサポートされています。詳細については、「帯域幅の重み付けでサポートされているインスタンスタイプ」を参照してください。
- ebs-1 帯域幅の重み付けを使用すると、EBS 帯域幅が最大 25% 増加するため、I/O 集約型アプリケーションのパフォーマンスを向上させることができます。ただし、VPC ネットワーク帯域幅は同じ絶対量だけ減少することに注意してください (EBS とネットワーク間の帯域幅仕様の組み合わせは変わりません)。
- 帯域幅の重み付けを変更すると、I/O パフォーマンスに大きな影響を与える可能性があります。vpc-1 帯域幅の重み付けにより、ネットワーク帯域幅が増加しますが、EBS ボリュームの IOPS が予想よりも減少する可能性があります。これは、特に I/O サイズが大きい場合、IOPS 制限の前に EBS 帯域幅制限に達する可能性があるためです。例えば、16 KiB の I/O サイズで通常 240,000 IOPS をサポートするインスタンスタイプでは、vpc-1 帯域幅の重みを使用する場合、EBS 帯域幅の減少により IOPS が減少する可能性があります。
- 常に特定のワークロードをテストして、選択した帯域幅の重み付けがパフォーマンスのニーズを満たすことを確認します。
- インスタンスの起動時に帯域幅の重み付けを設定することも、停止したインスタンスの帯域幅の重み付けを変更することもできます。詳細については、「Configure bandwidth weighting for your instance」を参照してください。
- インスタンス帯域幅の重み付けは、追加料金なしで設定できます。

Amazon EBS I/O の特性およびモニタリング

ボリューム設定が同じであっても、特定の I/O 特性により EBS ボリュームのパフォーマンス動作が向上します。

- SSD-Backed ボリューム、汎用 SSD (gp2 および gp3) とプロビジョンド IOPS SSD (io1 および io2) は、I/O 操作がランダムでもシーケンシャルでも、安定したパフォーマンスを提供します。
- HDD-Backed ボリューム、スループット最適化 HDD (st1) および Cold HDD (sc1) は、サイズが大きくシーケンシャルな I/O 操作の場合のみ、最適なパフォーマンスを提供します。

アプリケーションにおける SSD ボリュームおよび HDD ボリュームのパフォーマンスについて理解するには、ボリュームに対するデマンド、ボリュームに対して使用可能な IOPS の量、I/O 操作が完了するまでにかかる時間、およびボリュームのスループット制限の間のつながりについて知ることが重要です。

トピック

- IOPS
- ボリューム のキュー長とレイテンシー
- I/O サイズとボリュームのスループット制限
- CloudWatch を使用して I/O 特性を監視する
- リアルタイムの I/O パフォーマンス統計をモニタリングする
- 関連リソース

IOPS

IOPS とは、1 秒あたりの入出力操作数を表す測定単位です。操作は KiB 単位で計測され、基礎となるドライブテクノロジーが1 つの I/O としてカウントするデータの最大量を決定します。I/O サイズは、SSD ボリュームで 256 KiB、HDD ボリュームで 1,024 KiB に制限されます。これは、小さい I/O やランダム I/O の扱いにおいて、SSD ボリュームは HDD ボリュームに比べて はるかに効率的であるためです。

小さな I/O 操作が物理的に連続している場合、Amazon EBS ではできる限りこれらを最大の I/O サイズになるまで、単一の I/O 操作にマージして処理します。同様に、I/O 操作が最大 I/O サイズより大きい場合、Amazon EBS ではより小さな I/O 操作に分割して処理しようとします。例をいくつか、次の表に示します。

ボリュームタイプ	最大 I/O サイズ	アプリケーションからの I/O 操作	IOPS 数	コメント
SSD	256 KiB	1 x 1024 KiB I/O 操作	4 (1,024 ÷ 256 = 4)	Amazon EBS は、1,024 KiB の I/O 操作を 4 つのより小さな 256 KiB の操作に分割します。
		8 x シーケンシャル 32 KiB I/O 操作	1 (8 x 32 = 256)	Amazon EBS は、8 つの連続した 32 KiB の I/O 操作を、1 つの 256 KiB の操作にマージします。
		8 つのランダムな 32 KiB の I/O 操作	8	Amazon EBS は、ランダムな I/O 操作を個別にカウントします。
HDD	1,024 KIB	1 x 1024 KiB I/O 操作	1	I/O 操作は、すでに最大 I/O サイズと等しくなっています。マージまたは分割されません。
		8 x シーケンシャル 128 KiB I/O 操作	1 (8 x 128 = 1,024)	Amazon EBS は、連続した 8 つの 128 KiB I/O 操作を、1 つの 1,024 KiB の I/O 操作にマージします。

ボリュームタイプ	最大 I/O サイズ	アプリケーションからの I/O 操作	IOPS 数	コメント
		8 つのランダムな 32 KiB の I/O 操作	8	Amazon EBS は、ランダムな I/O 操作を個別にカウントします。

このため、3,000 IOPS をサポートする SSD-Backed ボリュームを (io1 または io2 ボリュームを 3,000 IOPS でプロビジョニングするか、gp2 ボリュームを 1,000 GiB にサイズ設定するか、gp3 ボリュームを使用することによって) 作成し、十分な帯域幅を提供できる EBS 最適化インスタンスにアタッチした場合、1 秒あたり最大 3,000 件の I/O 操作分のデータを転送できます (スループットは I/O サイズで決まります)。

ボリューム のキュー長とレイテンシー

ボリュームのキュー長とは、デバイスに対する保留中の I/O リクエストの数です。レイテンシーとは、実際に I/O 操作にかかるエンドツーエンドのクライアント時間です。つまり、I/O を EBS に送信してから、読み取りまたは書き込みの I/O が完了したという確認を EBS から受信するまでの時間ということになります。ゲストオペレーティングシステムまたは EBS へのネットワークリンクでのボトルネックを回避するには、I/O サイズとレイテンシーに合わせて正しくキュー長を調整する必要があります。

最適なキュー長は、アプリケーションがどの程度 IOPS およびレイテンシーの影響を受けるかによってワークロードごとに異なります。EBS ボリュームで利用可能なパフォーマンスをフル活用するための十分な I/O リクエストがワークロードから提供されないと、プロビジョニングどおりの IOPS またはスループットをボリュームで実現できないことがあります。

トランザクション量の多いアプリケーションは、I/O レイテンシーの上昇の影響を受けるため、SSD-Backed ボリュームが適しています。キュー長を小さく抑え、ボリュームで利用可能な限り高い IOPS を維持することにより、低いレイテンシーと高い IOPS を実現できます。ボリュームで利用可能な IOPS を超える IOPS を継続的に強制すると、I/O レイテンシーが上昇する可能性があります。最大限の整合性を確保するには、ボリュームは 1 分間で 1,000 のプロビジョンド IOPS ごとに 1 の平均キュー深度 (最も近い整数に四捨五入) を維持する必要があります。例えば、3,000 IOPS でプロビジョニングされたボリュームの場合、平均キュー深度は 3 である必要があります。

スループットが高いアプリケーションは I/O レイテンシーの上昇による影響を受けにくいため、HDD-Backed ボリュームが適しています。HDD-Backed ボリュームに対する高いスループットを維持するには、サイズの大きなシーケンシャル I/O を実行するときにキュー長を大きくします。

I/O サイズとボリュームのスループット制限

SSD-Backed ボリュームで I/O サイズが非常に大きい場合は、ボリュームのスループット制限に達することにより、IOPS 値がプロビジョニングした値よりも小さくなることがあります。例えば、利用可能なバーストクレジットを持つ 1,000 GiB 未満の gp2 ボリュームの IOPS 制限は 3,000 で、ボリュームスループット制限は 250 MiB/秒です。256 KiB の I/O サイズを使用している場合、ボリュームは 1000 IOPS (1000 x 256 KiB = 250 MiB) でスループット制限に達します。より小さい I/O サイズ (16 KiB など) では、スループットが 250 MiB/s を大幅に下回っているため、同じボリュームで 3,000 IOPS を維持できます。(これらの例では、ボリュームの I/O がインスタンスのスループット限界に達していないと想定しています)。各 EBS ボリュームタイプのスループット制限については、Amazon EBS ボリュームの種類を参照してください。

サイズの小さな I/O 操作では、インスタンス内で測定した IOPS がプロビジョニングの値より高くなることがあります。この状況は、インスタンスのオペレーティングシステムが、小さな I/O 操作を Amazon EBS に渡す前に、大きな操作にマージした場合に生じます。

ワークロードが HDD バックアップの st1 および sc1 ボリュームでシーケンシャル I/O を使用する場合、ワークロードで使用している I/O がシーケンシャルであれば、インスタンス内で測定した IOPS が予測値より高くなることがあります。この状況は、インスタンスのオペレーティングシステムが、シーケンシャル I/O をマージし、1,024 KiB サイズ単位でカウントすることによって生じます。ワークロードで小さな I/O またはランダム I/O を使用している場合は、スループットが予測値より低くなることがあります。これは、各ランダムな非連続 I/O が合計 IOPS 数にカウントされるためです。これにより、ボリュームの IOPS 制限に予想よりも早く達する可能性があります。

EBS ボリュームタイプにかかわらず、設定で予想される IOPS またはスループットが発生していない場合は、EC2 インスタンスの帯域幅が制限要因ではないことを確認してください。最適なパフォーマンスを得るには、常に現行世代の EBS 最適化インスタンス (または、10 Gb/s のネットワーク接続を確保できるインスタンス) を使用してください。予想された IOPS が得られない別の原因として、EBS ボリュームに対して十分な I/O を提供していないことが考えられます。

CloudWatch を使用して I/O 特性を監視する

これらの I/O 特性は、各ボリュームの CloudWatch ボリュームメトリクスを使用してモニタリングできます。

停止した I/O をモニタリングする

VolumeStalledIOCheck は、EBS ボリュームのステータスをモニタリングして、ボリュームに障害が発生した時期を判断します。メトリクスは、EBS ボリュームが I/O 操作を完了できるかどうかに基づいて 0 (合格) または 1 (失敗) ステータスを返すバイナリ値です。

VolumeStalledIOCheck メトリクスが失敗した場合、 が問題を解決 AWS するのを待つか、影響を受けるボリュームの置き換えや、ボリュームがアタッチされているインスタンスの停止と再起動などのアクションを実行できます。ほとんどの場合、このメトリクスが失敗すると、EBS は数分以内にボリュームを自動的に診断して復元します。で I/O の一時停止アクションを使用すると AWS Fault Injection Service 、制御された実験を実行して、このメトリクスに基づいてアーキテクチャとモニタリングをテストし、ストレージの障害に対する耐障害性を向上させることができます。

ボリュームの I/O レイテンシーをモニタリングする

VolumeAvgReadLatency および VolumeAvgWriteLatency メトリクスをそれぞれ使用して、Amazon EBS ボリュームの読み取りおよび書き込み操作の平均レイテンシーをモニタリングできます。の Latency Injection アクションを使用すると AWS Fault Injection Service 、制御された実験を実行して、このメトリクスに基づいてアーキテクチャとモニタリングをテストし、ストレージパフォーマンスの低下に対する耐障害性を向上させることができます。

I/O レイテンシーが必要な値よりも高い場合、ボリュームに対してプロビジョニングした IOPS またはスループット以上の処理をアプリケーションが実行しようとしていないことを確認します。VolumeAvgIOPS および VolumeAvgThroughput メトリクスを使用して、1 分間にボリュームに駆動される平均 IOPS とスループットをモニタリングし、それをボリュームのプロビジョンド IOPS とスループットと比較できます。ボリュームが 1 分間に操作を駆動しない場合、メトリクスでは 0 の値が報告されます。高 IOPS またはスループットのバーストが 1 分よりも短い間隔で発生した場合、ボリュームにマイクロバーストが発生しますが、平均 IOPS およびスループットメトリクスでは、ボリュームのプロビジョンド IOPS またはスループット制限よりも低いパフォーマンスが駆動されていると報告される可能性があります。ボリュームで特定の 1 分間にパフォーマンスのバーストが発生しているかどうかを確認するには、VolumeIOPSExceededCheck および VolumeThroughputExceededCheck メトリクスを使用します。これらのメトリクスをモニタリングして、ワークロードが一貫して特定の 1 分間にボリュームのプロビジョニングされたパフォーマンスよりも大きい IOPS またはスループットを駆動しようとしたかを判断できます。1 分間の任意の秒において駆動された IOPS が一貫してボリュームのプロビジョンド IOPS のパフォーマンスを超える場合、VolumeIOPSExceededCheck メトリクスは 1 を返します。1 分以内の任意の秒において駆動されたスループットが一貫してボリュームのプロビジョニングされたスループットパフォーマンスを超える場合、VolumeThroughputExceededCheck メトリクスは 1 を返します。駆動された IOPS とスループットがボリュームのプロビジョニングされたパフォーマンス内に留まっている場合、メトリクスは 0 を返します。

ボリュームが提供できるよりも多くの IOPS がアプリケーションに必要な場合は、次のいずれかの使用を検討する必要があります。

- 必要なレイテンシーを実現するのに十分な IOPS がプロビジョニングされている、gp3、io2、または io1 ボリューム
- 十分なベースライン IOPS パフォーマンスを実現するより大容量の gp2 ボリューム

st1 および sc1 の HDD-Back ボリュームは、1,024 KiB の最大 I/O サイズを活用するワークロードに最適な設定になっています。ボリュームの平均 I/O サイズを求めるには、VolumeWriteBytes を VolumeWriteOps で除算します。読み取り操作にも同じ計算を適用できます。平均 I/O サイズが 64 KiB を下回る場合は、st1 または sc1 のボリュームに送る I/O 操作のサイズを大きくすると、パフォーマンスが向上します。

gp2、st1、および sc1 ボリュームのバーストバケットバランスをモニタリングする

BurstBalance は gp2、st1、および sc1 ボリュームのバーストバケットバランスを残りのバランスの割合として表示します。バーストバケットが減ると、ボリューム I/O (gp2 ボリューム用) またはボリュームスループット (st1 および sc1 ボリューム用) はベースラインにスロットリングされます。この理由でボリュームに制限が適用されているかどうかを確認するには、BurstBalance の値を調べてください。利用可能な Amazon EBS メトリクスの完全なリストについては、Amazon EBS の Amazon CloudWatch メトリクス および「Nitro ベースのインスタンスの Amazon EBS メトリクス」を参照してください。

リアルタイムの I/O パフォーマンス統計をモニタリングする

Nitro ベースの Amazon EC2 インスタンスにアタッチされている Amazon EBS ボリュームの詳細なパフォーマンス統計にリアルタイムでアクセスできます。

これらの統計を組み合わせて、平均レイテンシーと IOPS を導き出したり、I/O 操作が完了中かどうかを確認したりすることができます。また、アプリケーションが EBS ボリュームまたはアタッチされたインスタンスのプロビジョンド IOPS またはスループット制限を超えた合計時間を表示することもできます。これらの統計の増加を経時的に追跡することで、アプリケーションのパフォーマンスを最適化するためにプロビジョンド IOPS またはスループット制限を増やす必要があるかどうかを判断できます。詳細なパフォーマンス統計には、読み取りおよび書き込み I/O 操作のヒストグラムも含まれています。これにより、レイテンシーバンド内で完了した I/O 操作の合計数を追跡することで、I/O レイテンシーの分布を確認できます。

詳細については、「Amazon EBS の詳細なパフォーマンス統計」を参照してください。

関連リソース

Amazon EBS の I/O 特性の詳細については、re:Invent プレゼンテーションAmazon EBS: パフォーマンスを考慮した設計を参照してください。

Amazon EBS および RAID の構成

Amazon EBS では、従来のベアメタルサーバーで使用できる標準的な RAID 設定はすべて使用できます。ただしその RAID 設定が、お使いのインスタンスのオペレーティングシステムでサポートされている必要があります。これは、RAID がすべてソフトウェアレベルで実現されるためです。

Amazon EBS ボリュームのデータは、同じアベイラビリティーゾーン内の複数のサーバーにレプリケートされます。これは、コンポーネントの 1 つに障害が発生したことが原因でデータが失われるのを防ぐためです。このレプリケーションにより、一般的なコモディティディスクドライブに比べて Amazon EBS ボリュームの信頼性が 10 倍に高まります。詳細については、「Amazon EBS の特徴」を参照してください。

内容

- RAID 設定オプション
- RAID 0 アレイの作成
- RAID アレイでのボリュームのスナップショットの作成

RAID 設定オプション

RAID 0 アレイを作成すると、単一の Amazon EBS ボリュームでプロビジョニングする場合よりも、ファイルシステムで高レベルのパフォーマンスが実現されます。I/O パフォーマンスが最も重要視される場合には、RAID 0 を使用します。RAID 0 では、I/O がストライプ内のボリューム全体に分散されます。ボリュームを追加すると、スループットと IOPS を追加したことになります。ただし、ストライプのパフォーマンスは、セット内で最もパフォーマンスの低いボリュームにより制限されることに留意してください。セット内のボリュームが 1 つ失われた場合でも、結果としてアレイのデータが完全に失われます。

RAID 0 アレイの最終的なサイズは、アレイ内のボリュームサイズの合計です。帯域幅は、アレイ内のボリュームで利用可能な帯域幅の合計です。例えば、4,000 のプロビジョンド IOPS が設定された 500 GiB の `io1` ボリュームが 2 つある場合、そのそれぞれが、使用可能な帯域幅が 8,000 IOPS でスループットが 1,000 MiB/秒の、1,000 GiB の RAID 0 アレイを構築します。

> **Important**
>
> RAID 5 と RAID 6 ではボリュームに使用できる IOPS の一部がパリティ書き込み操作によって消費されるため、Amazon EBS にはこれらの RAID モードをお勧めしません。RAID アレイの構成によっては、これらの RAID モードで使用できる IOPS が RAID 0 構成と比較して 20 ~ 30% 少なくなる場合があります。これらの RAID モードにはコストの増加も伴います。ボリュームサイズとスピードが同じ 2 ボリュームの RAID 0 アレイの方が、コストが 2 倍の 4 ボリュームの RAID 6 アレイよりも優れたパフォーマンスが得られる場合があります。
>
> RAID 1 も、Amazon EBS での使用が推奨されません。RAID 1 ではデータが同時に複数のボリュームに書き込まれるため、非 RAID 構成と比較して、Amazon EC2 と Amazon EBS の間により大きな帯域幅が必要となります。さらに、RAID 1 は書き込みパフォーマンスの向上をもたらしません。

RAID 0 アレイの作成

次の手順に従って RAID 0 アレイを作成します。

考慮事項

- この手順を実行する前に、RAID 0 アレイのサイズおよびプロビジョニングする IOPS 数を決定してください。
- アレイに作成するボリュームのサイズと IOPS パフォーマンス値は同一にしてください。EC2 インスタンスで利用可能な帯域幅を超えるアレイを作成しないよう注意してください。
- RAID ボリュームからの起動は避ける必要があります。デバイスの 1 つが失敗した場合、オペレーティングシステムを起動できなくなる場合があります。

Linux インスタンス

Linux で RAID 0 アレイを作成するには

1. アレイに Amazon EBS ボリュームを作成します。詳細については、「Amazon EBS ボリュームの作成」を参照してください。
2. アレイをホストするインスタンスに Amazon EBS ボリュームをアタッチします。詳細については、Amazon EBS ボリュームを Amazon EC2 インスタンスにアタッチを参照してください。

3. mdadm コマンドを使用して、新しくアタッチした Amazon EBS ボリュームから論理 RAID デバイスを作成します。[*number_of_volumes*] に、構成するアレイ内のボリュームの数を入れ、*device_name* に、アレイ内の各ボリュームのデバイス名 (/dev/xvdf など) を入れます。*MY_RAID* を、配列の一意の名前で置き換えることもできます。

   > **Note**
   >
   > インスタンスのデバイス名を見つけるには、lsblk コマンドを使用してデバイスのリストを表示します。

   RAID 0 アレイを作成するには、次のコマンドを実行します (アレイをストライプ化するには `--level=0` オプションをメモしておきます)。

   ```
   [ec2-user ~]$ sudo mdadm --create --verbose /dev/md0 --level=0 --name=MY_RAID --raid-devices=number_of_volumes device_name1 device_name2
   ```

   > **Tip**
   >
   > `mdadm: command not found` エラーが発生した場合は、`sudo yum install mdadm` コマンドを使用して mdadm をインストールします。

4. RAID アレイでの初期化と同期に許可された時間です。これらのオペレーションの進行状況は、次のコマンドを使用して追跡できます。

   ```
   [ec2-user ~]$ sudo cat /proc/mdstat
   ```

   出力例を次に示します。

   ```
   Personalities : [raid0]
   md0 : active raid0 xvdc[1] xvdb[0]
         41910272 blocks super 1.2 512k chunks

   unused devices: <none>
   ```

   一般的に、次のコマンドで RAID アレイに関する詳細情報を表示できます。

```
[ec2-user ~]$ sudo mdadm --detail /dev/md0
```

出力例を次に示します。

```
/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. RAID アレイにファイルシステムを作成し、それを後でマウントするときに、使用するラベルをそのファイルシステムに提供します。例えば、ext4 ファイルシステムのラベル *MY_RAID* で作成するには、次のコマンドを実行します。

```
[ec2-user ~]$ sudo mkfs.ext4 -L MY_RAID /dev/md0
```

アプリケーションの要件またはオペレーティングシステムの制限によって、ext3 や XFS などの異なるファイルシステムタイプを使用できます (対応するファイルシステム作成コマンドについては、ファイルシステムの資料を参照してください)。

6. 起動時に RAID アレイが自動的に再アセンブルされるようにするには、RAID 情報を含む設定ファイルを作成します。

```
[ec2-user ~]$ sudo mdadm --detail --scan | sudo tee -a /etc/mdadm.conf
```

> **Note**
>
> Amazon Linux 以外の Linux ディストリビューションを使用している場合は、このコマンドを変更する必要が生じることがあります。例えば、ファイルを別の場所に配置することや、`--examine` パラメータを追加することが必要な場合があります。詳細については、Linux インスタンスで man mdadm.conf を実行します。

7. 新しい RAID 設定のブロックデバイスモジュールを適切に事前ロードする新しいラムディスクイメージを作成する:

```
[ec2-user ~]$ sudo dracut -H -f /boot/initramfs-$(uname -r).img $(uname -r)
```

8. RAID アレイのマウントポイントを作成します。

```
[ec2-user ~]$ sudo mkdir -p /mnt/raid
```

9. 最後に、作成したマウントポイントに RAID デバイスをマウントします。

```
[ec2-user ~]$ sudo mount LABEL=MY_RAID /mnt/raid
```

これで RAID デバイスを使用する準備ができました。

10. (オプション) システムブート時に常に、この Amazon EBS ボリュームをマウントするには、`/etc/fstab` ファイルにデバイス用のエントリを追加します。

    a. `/etc/fstab` ファイルのバックアップコピーを作成すると、編集中に誤って破壊/削除してしまった場合にこのコピーを使用できます。

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

b. お好みのテキストエディタ (/etc/fstab や nano など) を使用して、vim ファイルを開きます。

c. 「UUID=」で始まる行にコメントして、ファイルの最後に次の形式で RAID ボリュームの新しい行を追加します。

```
device_label mount_point file_system_type fs_mntops fs_freq fs_passno
```

この行の最後の 3 つのフィールドは、ファイルシステムのマウントオプション、ファイルシステムのダンプ頻度、ブート時に実行されるファイルシステムチェックの順番です。これらの値がわからない場合は、次の例の値を使用してください。(defaults,nofail 0 2) /etc/fstab エントリの詳細については、fstab のマニュアルページを参照してください。(コマンドラインで man fstab を入力します。)) 例えば、マウントポイント /mnt/raid にラベル MY_RAID を持つデバイスに ext4 ファイルシステムをマウントするには、/etc/fstab に次のエントリを追加します。

> **Note**
>
> このボリュームをアタッチしないでインスタンスを起動することを目的としている場合 (例えば、このボリュームが異なるインスタンス間で移動される可能性がある場合)、nofail マウントオプションを追加し、ボリュームのマウントでエラーが発生してもインスタンスが起動できるようにしてください。Debian から派生した OS (Ubuntu など) では、nobootwait マウントオプションも追加する必要があります。

```
LABEL=MY_RAID       /mnt/raid   ext4    defaults,nofail        0       2
```

d. 新しいエントリを /etc/fstab に追加した後、エントリが正しく動作するかを確認する必要があります。sudo mount -a コマンドを使用して、すべてのファイルシステムを /etc/fstab にマウントします。

```
[ec2-user ~]$ sudo mount -a
```

前のコマンドを実行してもエラーが発生しない場合、/etc/fstab ファイルに問題はありません。次回ブート時にファイルシステムは自動的にマウントされます。このコマンドを実行してエラーが発生した場合、エラーを調べて、/etc/fstab を修正してください。

> **Warning**
>
> /etc/fstab ファイルにエラーがあると、システムがブート不能になる可能性があります。/etc/fstab ファイルにエラーがあるシステムをシャットダウンしないでください。

e. (オプション) /etc/fstab のエラーの修正方法が不明な場合、次のコマンドを使って、いつでもバックアップの /etc/fstab ファイルを復元することができます。

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

Windows インスタンス

Windows で RAID 0 アレイを作成するには

1. アレイに Amazon EBS ボリュームを作成します。詳細については、「Amazon EBS ボリュームの作成」を参照してください。
2. アレイをホストするインスタンスに Amazon EBS ボリュームをアタッチします。詳細については、Amazon EBS ボリュームを Amazon EC2 インスタンスにアタッチを参照してください。
3. Windows インスタンスに接続します。詳細については、「Windows インスタンスに接続する」を参照してください。
4. コマンドプロンプトを開いて、diskpart コマンドを入力します。

```
diskpart

Microsoft DiskPart version 6.1.7601
Copyright (C) 1999-2008 Microsoft Corporation.
On computer: WIN-BM6QPPL51CO
```

5. DISKPART プロンプトで、次のコマンドを使用して、利用できるディスクの一覧を表示できます。

```
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
```

アレイで使用するディスクを確認し、ディスクの番号を書き留めます。

6. アレイで使用する各ディスクは、既存のボリュームを含まないオンラインの動的なディスクである必要があります。ベーシックディスクを動的なディスクに変換する手順と既存のボリュームを削除する手順は以下のとおりです。

   a. 次のコマンドを使用して、アレイで利用するディスクを選択します。*n* を利用するディスクの番号に置き換えます。

   ```
   DISKPART> select disk n

   Disk n is now the selected disk.
   ```

   b. 選択したディスクが Offline と表示されている場合は、online disk コマンドを実行してオンラインにします。

   c. 選択したディスクにおいて、前述のlist disk コマンドの出力の Dyn 列にアスタリスクが付いていない場合、動的なディスクに変換する必要があります。

   ```
   DISKPART> convert dynamic
   ```

   > **Note**
   >
   > ディスクが書き込み禁止であることを示すエラーが表示された場合は、ATTRIBUTE DISK CLEAR READONLY コマンドを使って読み取り専用フラグをクリアしてから、動的ディスク変換をもう一度試してください。

   d. detail disk コマンドを使用して、選択したディスクの既存のボリュームを確認します。

   ```
   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
```

ディスクのボリュームの番号を書き留めます。この例では、ボリュームの番号は 2 です。ボリュームがない場合、次の手順は省略できます。

e. (前の手順でボリュームの存在が確認された場合のみ) 前の手順で確認したディスクで既存のボリュームを選択して削除します。

> **⚠ Warning**
>
> この手順により、ボリュームの既存データがすべて失われます。

i. ボリュームを選択します。*n* をボリュームの番号と置き換えてください。

```
DISKPART> select volume n
Volume n is the selected volume.
```

ii. ボリュームを削除します。

```
DISKPART> delete volume

DiskPart successfully deleted the volume.
```

iii. 選択したディスクで削除する必要があるボリュームごとに以下の手順を繰り返します。

f. アレイで使用するディスクごとにStep 6を繰り返します。

7. 使用するディスクが動的なディスクであるかどうかを確認します。この例では、RAID ボリュームにディスク 1 と 2 を使用します。

```
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. RAID アレイを作成します。Windows では、RAID 0 ボリュームはストライプ化されたボリュームとして参照されます。

   次のコマンドにより、ディスク 1 とディスク 2 上にストライプ化されたボリュームアレイを作成します (アレイをストライプ化するには `stripe` オプションをメモしてください)。

```
DISKPART> create volume stripe disk=1,2
DiskPart successfully created the volume.
```

9. 新しいボリュームを確認します。

```
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
```

   Type 列には、Volume 1 が `stripe` ボリュームであることが示されていることに注意してください。

10. ボリュームを選択してフォーマットし、ボリュームの使用を開始できるようにします。

    a. フォーマットするボリュームを選択します。*n* をボリュームの番号に置き換えます。

```
DISKPART> select volume n

Volume n is the selected volume.
```

    b. ボリュームをフォーマットします。

> **Note**
>
> 完全フォーマットを実行するには、quick オプションを省略します。

```
DISKPART> format quick recommended label="My new volume"

  100 percent completed

DiskPart successfully formatted the volume.
```

c. ボリュームに使用可能な任意のドライブ文字を割り当てます。

```
DISKPART> assign letter f

DiskPart successfully assigned the drive letter or mount point.
```

新しいボリュームを使用する準備ができました。

RAID アレイでのボリュームのスナップショットの作成

スナップショットを使用して RAID アレイの EBS ボリュームのデータをバックアップする場合は、スナップショットが一貫していることを確認する必要があります。これは、ボリュームのスナップショットが個別に作成されるためです。同期されていないスナップショットから RAID 配列の EBS ボリュームを復元すると、配列の整合性は低下します。

RAID 配列の一貫性のあるスナップショットを作成するには、EBS マルチボリュームスナップショットを使用します。マルチボリュームスナップショットを使用すると、EC2 インスタンスにアタッチされている複数の EBS ボリュームにわたって、ポイントインタイムで、データ調整済みの Crash-consistent スナップショットを取得できます。スナップショットは複数の EBS ボリュームにわたって自動的に作成されるため、整合性を確保するためにボリューム間で調整するためにインスタンスを停止する必要はありません。詳細については、「Create Amazon EBS snapshots」のマルチボリュームスナップショットを作成するステップを参照してください。

Amazon EBS ボリュームのベンチマーク

I/O ワークロードをシミュレートすることで、Amazon EBS ボリュームのパフォーマンスをテストできます。手順は次のとおりです。

1. EBS 最適化インスタンスを作成する。
2. 新しい EBS ボリュームを作成します。
3. EBS 最適化インスタンスにボリュームをアタッチする。
4. ブロックデバイスを設定およびマウントします。
5. ツールをインストールし、I/O パフォーマンスを評価する。
6. ボリュームの I/O パフォーマンスを評価する。
7. ボリュームを削除し、料金が発生しないようにインスタンスを終了する。

> **Important**
>
> 手順の一部を実行すると、ベンチマークを実行する EBS ボリューム上の既存のデータが破壊されます。ベンチマーキングの手順は、本番ボリュームではなく、テスト目的で特別に作成されたボリュームで使用するために用意されています。

インスタンスのセットアップ

EBS ボリュームで最適なパフォーマンスを実現するには、EBS 最適化インスタンスを使用することをお勧めします。EBS 最適化インスタンスは、Amazon EC2 と Amazon EBS の間で所定の帯域幅を実現するものであり、インスタンスタイプに応じて仕様で選択できます。

EBS 最適化インスタンスを作成するには、Amazon EC2 コンソールを使用してインスタンスを起動するときに [EBS 最適化インスタンスとして起動する] を選択するか、コマンドラインを使用するときに --ebs-optimized を指定します。このオプションをサポートするインスタンスタイプを必ず選択してください。

Provisioned IOPS SSD または 汎用 SSD ボリュームの設定

Amazon EC2 コンソールを使用して、プロビジョンド IOPS SSD (io1 および io2) または汎用 SSD (gp2 および gp3) ボリュームを作成するには、[ボリュームタイプ] で、[プロビジョンド IOPS SSD (io1)]、[プロビジョンド IOPS SSD (io2)]、[汎用 SSD (gp2)]、または [汎用 SSD (gp3)] を選択し

ます。コマンドラインの --volume-type パラメータには、io1、io2、gp2、または gp3 を指定します。io1、io2、および gp3 ボリュームの場合は、--iops パラメータに、1 秒あたりの I/O オペレーション数 (IOPS) を指定します。詳細については、「Amazon EBS ボリュームの種類」および「Amazon EBS ボリュームの作成」を参照してください。

(Linux インスタンスのみ) テストの例には、6 ボリュームを備えた RAID 0 アレイを作成することをお勧めします。これにより、高いレベルのパフォーマンスを実現できます。料金は、ボリューム数ではなく、プロビジョニングされたギガバイト (および io1、io2、gp3 ボリュームに対してプロビジョニングされた IOPS 数) に対して発生します。したがって、ストライプセットを作成するために、複数の小さなボリュームを作成しても追加コストは発生しません。Oracle Orion を使用してボリュームを評価する場合は、Oracle ASM と同じ方法でストライピングをシミュレートできます。したがって、Orion でストライピングを行えるようにすることをお勧めします。別のベンチマーキングツールを使用する場合は、ボリュームのストライピングを自身で行う必要があります。

RAID 0 アレイの作り方の詳細については、「RAID 0 アレイの作成」を参照してください。

スループット最適化 HDD (**st1**) または Cold HDD (**sc1**) ボリュームをセットアップする

st1 ボリュームを作成するには、Amazon EC2 コンソールを使用してボリュームを作成するときに [スループット最適化 HDD] を選択するか、コマンドラインを使用して --type **st1** を指定します。sc1 ボリュームを作成するには、Amazon EC2 コンソールを使用してボリュームを作成するときに [Cold HDD] を選択するか、コマンドラインを使用して --type **sc1** を指定します。EBS ボリュームの作成の詳細については、Amazon EBS ボリュームの作成を参照してください。インスタンスへのこれらのボリュームのアタッチについては、Amazon EBS ボリュームを Amazon EC2 インスタンスにアタッチを参照してください。

(Linux インスタンスのみ) は、この設定手順を簡素化 CloudFormation する で使用する JSON テンプレート AWS を提供します。テンプレートにアクセスして JSON ファイルとして保存します。 では、独自の SSH キーを設定 CloudFormation でき、st1ボリュームを評価するパフォーマンステスト環境を簡単にセットアップできます。テンプレートを使用すると、現行世代のインスタンスと 2 TiB の st1 ボリュームが作成され、このボリュームが /dev/xvdf のインスタンスにアタッチされます。

(Linux インスタンスのみ) テンプレートを使用して HDD ボリュームを作成する方法

1. https://console.aws.amazon.com/cloudformation で CloudFormation コンソールを開きます。
2. [Create Stack] を選択します。

3. [Upload a Template to Amazon S3] を選択し、さきほど入手した JSON テンプレートを選択します。
4. スタックに、「ebs-perf-testing「 のような名前を付け、インスタンスタイプ (デフォルトは r3.8xlarge) および SSH キーを選択します。
5. [Next] を 2 回選択し、[Create Stack] を選択します。
6. 新しいスタックのステータスが [CREATE_IN_PROGRESS] から [COMPLETE] に移行したら、[出力] を選択して新しいインスタンスのパブリック DNS エントリを取得します。このインスタンスには、2 TiB の `st1` ボリュームがアタッチされます。
7. ユーザー **`ec2-user`** として、前のステップで DNS エントリから取得したホスト名を使用し、新しいスタックに SSH を使用して接続します。
8. ベンチマークツールのインストール に進みます。

ベンチマークツールのインストール

次の表に、EBS ボリュームのパフォーマンスをベンチマークするために使用できるツールのいくつかを示します。

Linux インスタンス

ツール	説明

fio	I/O ベンチマーキングを評価します。(fio は `libaio-devel` に依存することに注意してください。)
	
	fio を Amazon Linux にインストールするには、次のコマンドを実行します。
	
	`$ sudo yum install -y fio`
	
	Ubuntu に fio インストールするには、次のコマンドを実行します。
	
	`sudo apt-get install -y fio`

Oracle Orion Calibration ツール	Oracle データベースで使用するストレージシステムの I/O パフォーマンスを調整します。

Windows インスタンス

ツール	説明
DiskSpd	DiskSpd は、Microsoft の Windows、Windows Server、クラウドサーバーインフラストラクチャエンジニアリングチームのストレージパフォーマンスツールです。これは、次からダウンロードできます。https://github.com/Microsoft/diskspd/releases `diskspd.exe` 実行可能ファイルをダウンロードした後、管理者権限でコマンドプロンプトを開き (「管理者として実行を選択」)、`diskspd.exe` ファイルをコピーしたディレクトリに移動します。 適切な実行可能フォルダ (`amd64fre`、`armfre` または `x86fre`)) から目的の `diskspd.exe` 実行可能ファイルを `C:\DiskSpd` などの短い、単純なパスにコピーします。ほとんどの場合、`amd64fre` フォルダから 64 ビットバージョンの DiskSpd を使用します。 DiskSpd のソースコードは、https://github.com/Microsoft/diskspd の GitHub でホストされています。
CrystalDiskMark	CrystalDiskMark は、シンプルなディスクベンチマークソフトウェアです。https://crystalmark.info/en/software/crystaldiskmark/ でダウンロードできます。

これらのベンチマーキングツールは、さまざまなテストパラメータをサポートしています。使用するのは、ボリュームがサポートするワークロードを見積もるためのコマンドです。評価に必要な基本的なコマンドの例を以下に示します。

ボリュームキュー長の選択

ワークロードとボリュームタイプに基づいて最適なボリュームキュー長を選択します。

SSD-Backed ボリュームのキュー長

SSD-Backed ボリュームでワークロードに最適なキュー長を決定するには、使用可能な 1000 IOPS ごとにキュー長 1 を指定するようにお勧めします (汎用 SSD ボリュームのベースライン、Provisioned IOPS SSD ボリュームにプロビジョニングする値)。その後、アプリケーションのパフォーマンスを監視して、アプリケーション要件に応じて値を調整することができます。

プロビジョニングした IOPS、スループット、または最適なシステムキュー長 (現在は 32 に設定) に達するまでは、キュー長を大きくする方が有益です。例えば、IOPS として 3,000 がプロビジョニングされたボリュームでは、キュー長 3 を設定します。アプリケーションに最適な値を確認するには、これらの値を増減して調整してください。

HDD-Backed ボリュームのキュー長

HDD-Backed ボリュームのワークロードに対する最適なキュー長を決定するには、1 MiB のシーケンシャル I/O の実行時に 4 以上のキュー長を設定しておくようお勧めします。その後、アプリケーションのパフォーマンスを監視して、アプリケーション要件に応じて値を調整することができます。例えば、2 TiB の `st1` ボリュームで、バーストスループットが 500 MiB/秒、IOPS が 500 の場合は、1,024 KiB、512 KiB、または 256 KiB のシーケンシャル I/O を実行する際に、キュー長をそれぞれ 4、8、または 16 に設定します。アプリケーションに最適な値を確認するには、これらの値を増減して調整してください。

C ステートの無効化

ベンチマーキングを実行する前に、プロセッサの C ステートを無効にする必要があります。サポートされている CPU の一時的にアイドリング状態のコアは、電力を節約するために C ステートに入ることができます。コアが処理を再開するために呼び出されると、コアが再び完全に動作するまで一定の時間が経過します。このレイテンシーは、プロセッサのベンチマーキングルーチンを妨げる可能性があります。C ステートとその EC2 インスタンスタイプでサポートされるインスタンスの詳細については、EC2 インスタンスタイプのプロセッサのステート制御を参照してください。

Linux インスタンス

Amazon Linux、RHEL、および CentOS で C ステートを無効にするには、次のようにします。

1. C ステートの数を取得します。

```
$ cpupower idle-info | grep "Number of idle states:"
```

2. C ステート c1 から cN にして無効にします。理想的には、コアは c0 ステートにある必要があります。

```
$ for i in `seq 1 $((N-1))`; do cpupower idle-set -d $i; done
```

Windows インスタンス

次のようにして、Windows システムで C ステートを無効にできます。

1. PowerShell で、現在のアクティブな電力スキームを取得します。

```
$current_scheme = powercfg /getactivescheme
```

2. 電力スキームの GUID を取得します。

```
(Get-WmiObject -class Win32_PowerPlan -Namespace "root\cimv2\power" -Filter
 "ElementName='High performance'").InstanceID
```

3. 電力設定 GUID を取得します。

```
(Get-WmiObject -class Win32_PowerSetting -Namespace "root\cimv2\power" -Filter
 "ElementName='Processor idle disable'").InstanceID
```

4. 電力設定サブグループの GUID を取得します。

```
(Get-WmiObject -class Win32_PowerSettingSubgroup -Namespace "root\cimv2\power" -
Filter "ElementName='Processor power management'").InstanceID
```

5. インデックスの値を 1 に設定して、C ステートを無効にします。値 0 は、C ステートが無効であることを示します。

```
powercfg /
setacvalueindex <power_scheme_guid> <power_setting_subgroup_guid> <power_setting_guid>
 1
```

6. アクティブなスキームを設定して、設定が保存されていることを確認します。

```
powercfg /setactive <power_scheme_guid>
```

ベンチマーキングを実行する

次の手順では、さまざまな EBS ボリュームタイプに対するベンチマークコマンドについて説明します。

EBS ボリュームがアタッチされている EBS 最適化インスタンスで、次のコマンドを実行します。EBS ボリュームをスナップショットから作成した場合は、ベンチマーキングを実行する前に、

必ず初期化してください。詳細については、「作成後にボリュームを手動で初期化する」を参照してください。

> **Tip**
>
> EBS の詳細なパフォーマンス統計によって提供される I/O レイテンシーヒストグラムを使用して、ベンチマークテストの I/O パフォーマンスの分布を比較できます。詳細については、「Amazon EBS の詳細なパフォーマンス統計」を参照してください。

ボリュームのテストが完了したら、クリーンアップに関する次のトピックの Amazon EBS ボリュームの削除 および「インスタンスの終了」を参照してください。

Provisioned IOPS SSD ボリュームと 汎用 SSD ボリュームをベンチマークする

Linux インスタンス

作成した RAID 0 アレイで fio を実行します。

次のコマンドは、16 KB のランダム書き込みオペレーションを実行します。

```
$ 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
```

次のコマンドは、16 KB のランダム読み取りオペレーションを実行します。

```
$ 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
```

結果の読み方については、チュートリアル「fio のディスク IO パフォーマンスの確認」を参照してください。

Windows インスタンス

作成したボリュームで DiskSpd を実行します。

次のコマンドは、C: ドライブ上にある 20 GB のテストファイルを使用して、30 秒のランダム I/O テストを実行します。書き込み率 25%、読み取り率 75%、ブロックサイズは 8 K です。これは、そ

れぞれ 4 つの未処理の I/O を持ち、1 GB の書き込みエントロピー値シードを持つ 8 つのワーカースレッドを使用します。テストの結果は、DiskSpeedResults.txt というテキストファイルに保存されます。これらのパラメータは、SQL Server OLTP ワークロードをシミュレートします。

```
diskspd -b8K -d30 -o4 -t8 -h -r -w25 -L -Z1G -c20G C:\iotest.dat > DiskSpeedResults.txt
```

結果の読み方については、チュートリアルDiskSPd のディスク IO パフォーマンスの確認を参照してください。

st1 および sc1 ボリュームのベンチマーク (Linux インスタンス)

st1 ボリュームまたは sc1 ボリュームで fio を実行します。

> **Note**
>
> これらのテストを実行する前に、st1 および sc1 (Linux インスタンスインスタンスのみ) で高いスループットの読み取りが多いワークロードに先読みを増やすの説明に従って、バッファ付き I/O をインスタンスに設定してください。

次のコマンドでは、アタッチされた st1 ブロックデバイス (例: /dev/xvdf) に対して、1 MiB のシーケンシャル読み取り操作を実行します。

```
$ 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
```

次のコマンドでは、アタッチされた st1 ブロックデバイスに対して、1 MiB のシーケンシャル書き込み操作を実行します。

```
$ 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
```

ワークロードによっては、ブロックデバイスの異なる部分に対してシーケンシャル読み取りとシーケンシャル書き込みの組み合わせを実行するケースがあります。このようなワークロードを評価する場合は、読み取りと書き込みに対して別々の fio ジョブを同時に実行し、fio offset_increment オプションを使用して、ブロックデバイスの別々の場所を各ジョブに割り当てることをお勧めします。

このワークロードの実行は、シーケンシャル書き込みまたはシーケンシャル読み取りのワークロードの場合より、少し複雑になります。テキストエディターを使用して、次の内容を含む fio ジョブファイル (この例では `fio_rw_mix.cfg`) を作成します。

```
[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
```

次に、以下のコマンドを実行します。

```
$ sudo fio fio_rw_mix.cfg
```

結果の読み方については、チュートリアルfio のディスク I/O パフォーマンスの確認を参照してください。

シーケンシャルの読み取りまたは書き込みの操作を使用しても、ダイレクト I/O の fio ジョブを複数実行した場合は、`st1` および `sc1` ボリュームで予測を下回るスループットになります。単一のダイ

レクト I/O ジョブを使用し、`iodepth` パラメータを指定して、I/O 操作の同時実行数を制御することをお勧めします。

Amazon Data Lifecycle Manager でバックアップを自動化

Amazon Data Lifecycle Manager を使用して、EBS スナップショットと EBS-backed AMI の作成、保持、削除を自動化できます。スナップショットと AMI 管理を自動化すると、次のことができるようになります。

- 定期的なバックアップスケジュールを実施して貴重なデータを保護する。
- 定期的に更新できる標準化された AMI を作成する。
- 監査担当者または社内のコンプライアンスが必要とするバックアップを保持する。
- 古いバックアップを削除してストレージコストを削減する。
- 分離されたリージョンまたはアカウントにデータをバックアップするディザスタリカバリ用バックアップポリシーを作成します。

Amazon EventBridge と AWS CloudTrail モニタリングと組み合わせることで、Amazon Data Lifecycle Manager は Amazon EC2 インスタンスと EBS ボリュームの完全なバックアップソリューションを追加料金なしで提供します。

> **Important**
>
> - Amazon Data Lifecycle Manager では、他の方法で作成されたスナップショットまたは AMI を管理することはできません。
> - Amazon Data Lifecycle Manager では、instance store-backed AMI の作成、保持、および削除を自動化することはできません。

Amazon Data Lifecycle Manager は、Amazon Elastic Block Store (Amazon EBS) のサービス機能として評価されます。Amazon EBS がリストされたコンプライアンスプログラムによるAWS 対象範囲内のサービス (FedRAMP、HIPAA BAA、SOC など) はすべて、Amazon Data Lifecycle Manager にも適用されます。

クォータ

AWS アカウントには、Amazon Data Lifecycle Manager に関連する次のクォータがあります。

説明	クォータ
リージョンごとのカスタムライフサイクルポリシー	100
リージョンごとの EBS スナップショットのデフォルトポリシー	1
リージョンごとの EBS-backed AMI のデフォルトポリシー	1
リソースあたりのタグ	45

Amazon Data Lifecycle Manager の仕組み

以下はAmazon Data Lifecycle Managerの主要な要素です。

要素

- ポリシー
- ポリシースケジュール (カスタムポリシーのみ)
- ターゲットリソースタグ (カスタムポリシーのみ)
- スナップショット
- EBS-backed AMI
- Amazon Data Lifecycle Manager のタグ

ポリシー

Amazon Data Lifecycle Manager で、バックアップの作成と保持の要件を定義するポリシーを作成します。これらのポリシーでは通常、以下を指定できます。

- ポリシータイプ – ポリシーで管理するバックアップリソースのタイプ (スナップショットまたは EBS-backed AMI) を定義します。
- ターゲットリソース – ポリシーのターゲットとなるリソースのタイプ (インスタンスまたは EBS ボリューム) を定義します。

- 作成頻度 – ポリシーを実行し、スナップショットまたは AMI を作成する頻度を定義します。
- 保持しきい値 – スナップショットまたは AMI の作成後、ポリシーで保持する期間を定義します。
- 追加アクション – クロスリージョンでのコピー、アーカイブ、リソースのタグ付けなど、ポリシーで実行する必要がある追加アクションを定義します。

Amazon Data Lifecycle Manager には、デフォルトポリシーとカスタムポリシーが用意されています。

デフォルトポリシー

デフォルトポリシーでは、最近のバックアップがないリージョン内のすべてのボリュームとインスタンスをバックアップします。必要に応じて、除外パラメータを指定してボリュームとインスタンスを除外できます。

Amazon Data Lifecycle Manager は、次のポリシーをサポートします。

- EBS スナップショットのデフォルトポリシー – ボリュームをターゲットとし、スナップショットの作成、保持、および削除を自動化します。
- EBS-backed AMI のデフォルトポリシー – インスタンスをターゲットし、EBS-backed AMI の作成、保持、および登録解除を自動化します。

デフォルトポリシーは、各アカウントおよび AWS リージョンのリソースタイプごとに 1 つしか設定できません。

カスタムポリシー

カスタムポリシーは、割り当てられたタグに基づく特定のリソースをターゲットとしており、高速スナップショット復元、スナップショットアーカイブ、クロスアカウントコピー、事前スクリプトと事後スクリプトなどの高度な機能をサポートします。カスタムポリシーには最大 4 つのスケジュールを含めることができ、各スケジュールには独自の作成頻度、保持しきい値、および高度な機能設定を設定できます。

Amazon Data Lifecycle Manager は、次のカスタムポリシーをサポートします。

- EBS スナップショットポリシー – ボリュームまたはインスタンスをターゲットとし、EBS スナップショットの作成、保持、および削除を自動化します。
- EBS-backed AMI ポリシー – インスタンスをターゲットし、EBS-backed AMI の作成、保持、および登録解除を自動化します。

- クロスアカウントコピーのイベントポリシー – 共有されているスナップショットのクロスリージョンコピーアクションを自動化します。

詳細については、「Amazon Data Lifecycle Manager のデフォルトポリシーとカスタムポリシー」を参照してください。

ポリシースケジュール (カスタムポリシーのみ)

ポリシースケジュールは、ポリシーによってスナップショットまたは AMI が作成されるタイミングを定義します。ポリシーは、最大 4 つのスケジュール — (1 つの必須スケジュールと、最大 3 つのオプションのスケジュール) を持つことができます。

1 つのポリシーに複数のスケジュールを追加すると、同じポリシーを使用して異なる頻度でスナップショットまたは AMI を作成できます。例えば、毎日、毎週、毎月、および毎年のスナップショットを作成する単一のポリシーを作成できます。これにより、複数のポリシーを管理する必要がなくなります。

スケジュールごとに、頻度、高速スナップショット復元設定 (スナップショットライフサイクルポリシーのみ)、クロスリージョンのコピールール、およびタグを定義できます。スケジュールに割り当てられているタグは、そのスケジュールが初期化された時点で作成される、スナップショットまたは AMI に自動的に割り当てられます。さらに、Amazon Data Lifecycle Manager は、スケジュールの頻度に基づいて、システム生成タグを各スナップショットまたは AMI に自動的に割り当てます。

各スケジュールは、その頻度に基づいて個別に初期化されます。複数のスケジュールが同時に初期化された場合、Amazon Data Lifecycle Manager はスナップショットまたは AMI を 1 つだけ作成し、保持期間が最も長いスケジュールのスナップショット保持設定を適用します。初期化されたすべてのスケジュールのタグがスナップショットまたは AMI に適用されます。

- (スナップショットライフサイクルポリシーのみ) 初期化された 1 つ以上のスケジュールで高速スナップショット復元が有効になっている場合、初期化されたすべてのスケジュールで指定されている、すべてのアベイラビリティーゾーンで、スナップショットの高速スナップショット復元が有効化されます。初期化されたスケジュールの最も長い保持設定が、各アベイラビリティーゾーンに対して使用されます。
- 初期化された複数のスケジュールでクロスリージョンコピーが有効化されている場合は、それらのスケジュール全体で指定されているすべてのリージョンに対し、スナップショットもしくは AMI がコピーされます。初期化されたスケジュールの最も長い保存期間が適用されます。

ターゲットリソースタグ (カスタムポリシーのみ)

Amazon Data Lifecycle Manager カスタムポリシーでは、バックアップするリソースを識別するためのリソースタグが使用されます。スナップショットまたは EBS-backed AMI ポリシーの作成時に、複数のターゲットリソースタグを指定することができます。指定されたタイプのリソース (インスタンスまたはボリューム) のうち、指定されたターゲットリソースタグの少なくとも 1 つを持つすべてのリソースがポリシーのターゲットになります。例えば、ボリュームをターゲットとするスナップショットポリシーを作成し、`purpose=prod`、`costcenter=prod`、`environment=live` をターゲットリソースタグとして指定した場合、ポリシーは、これらのタグとキー値のペアのいずれかを持つすべてのボリュームをターゲットとします。

リソースで複数のポリシーを実行する場合は、ターゲットリソースに複数のタグを割り当ててから、それぞれが特定のリソースタグをターゲットとする個別のポリシーを作成できます。

タグキーに \ や = の文字を使用することはできません。ターゲットリソースタグでは大文字と小文字が区別されます。詳細については、「リソースのタグ付け」を参照してください。

スナップショット

スナップショットは、EBS ボリュームからデータをバックアップするための主な手段です。ストレージコストを節約するために、連続するスナップショットは増分で、以前のスナップショット以降に変更されたボリュームデータのみが含まれています。ボリュームの一連のスナップショットでスナップショットを 1 つ削除すると、そのスナップショットに固有のデータだけが削除されます。キャプチャされたボリュームの残りの部分は保存されます。詳細については、「Amazon EBS スナップショット」を参照してください。

EBS-backed AMI

Amazon マシンイメージ (AMI) には、インスタンスの起動に必要な情報が用意されています。同じ設定で複数のインスタンスが必要な場合は、1 つの AMI から複数のインスタンスを起動できます。Amazon Data Lifecycle Manager は、EBS-backed AMI のみをサポートします。EBS-backed AMI には、ソースインスタンスにアタッチされた各 EBS ボリュームのスナップショットが含まれます。詳細については、「Amazon マシンイメージ (AMI)」を参照してください。

Amazon Data Lifecycle Manager のタグ

Amazon Data Lifecycle Manager は、ポリシーによって作成されたすべてのスナップショットと AMI に対し以下のようなタグを適用することで、他の方法で作成されたスナップショットや AMI と区別します。

- aws:dlm:lifecycle-policy-id
- aws:dlm:lifecycle-schedule-name
- aws:dlm:expirationTime – 期間ベースのスケジュールにより作成されたスナップショット用。スナップショットを標準階層から削除する時期を表します。
- dlm:managed
- aws:dlm:archived – スケジュールによりアーカイブされたスナップショット用。
- aws:dlm:pre-script – 事前スクリプトにより作成されたスナップショット用。
- aws:dlm:post-script – 事後スクリプトにより作成されたスナップショット用。

作成時に、スナップショットと AMI に適用するカスタムタグを指定することもできます。タグキーに \ や = の文字を使用することはできません。

Amazon Data Lifecycle Manager がボリュームをスナップショットポリシーに関連付けるために使用するターゲットタグは、オプションで、ポリシーによって作成されたスナップショットに適用できます。同様に、インスタンスを AMI ポリシーに関連付けるために使用するターゲットタグは、ポリシーによって作成された AMI にオプションで適用できます。

Amazon Data Lifecycle Manager のデフォルトポリシーとカスタムポリシー

このセクションでは、デフォルトポリシーとカスタムポリシーを比較し、それぞれの類似点と相違点に焦点を当てます。

トピック

- EBS スナップショットポリシーの比較
- EBS-backed AMI ポリシーの比較

EBS スナップショットポリシーの比較

次の表は、EBS スナップショットのデフォルトポリシーとカスタム EBS スナップショットポリシーの違いを示しています。

機能	EBS スナップショットのデフォルトポリシー	カスタム EBS スナップショットポリシー
マネージドバックアップリソース	EBS スナップショット	EBS スナップショット
ターゲットリソースタイプ	ボリューム	ボリュームまたはインスタンス
リソースターゲット設定	最新のスナップショットがないリージョン内のすべてのボリュームをターゲットとします。除外パラメータを指定して特定のボリュームを除外できます。	特定のタグを持つボリュームまたはインスタンスのみをターゲットとします。
除外パラメータ	はい。ブートボリューム、特定のボリュームタイプ、特定のタグが付いたボリュームは除外できます。	はい。インスタンスをターゲットにするときに、ブートボリュームと特定のタグが付いたボリュームを除外できます。
サポート AWS Outposts	いいえ	はい
複数のスケジュールのサポート	いいえ	はい。ポリシーごとに最大 4 つのスケジュールまで
サポートされている保持タイプ	経過期間ベースの保持のみ	経過期間ベースの保持とカウントベースの保持
スナップショットの作成頻度	1 ~ 7 日おき。	毎日、毎週、毎月、毎年、または cron 式を使用したカスタム頻度。
スナップショット保持期限	2 ~ 14 日。	最大 1,000 スナップショット (カウントベース) または最大 100 年 (経過期間ベース)。

機能	EBS スナップショットのデフォルトポリシー	カスタム EBS スナップショットポリシー
アプリケーション整合性のあるスナップショットのサポート	いいえ	はい。事前スクリプトと事後スクリプトを使用する
スナップショットアーカイブのサポート	いいえ	はい
高速スナップショット復元のサポート	いいえ	はい
クロスリージョンでのコピーのサポート	はい。デフォルト設定を使用 [1]	はい。カスタム設定を使用
クロスアカウント共有のサポート	いいえ	はい
拡張削除のサポート [2]	はい	いいえ

[1] デフォルトポリシーの場合:

- タグをクロスリージョンコピーにコピーすることはできません。
- コピーには、ソーススナップショットと同じ保持期間が使用されます。
- コピーは、ソーススナップショットと同じ暗号化状態を取得します。送信先リージョンで暗号化がデフォルトで有効になっている場合、ソーススナップショットが暗号化されていなくても、コピーは常に暗号化されます。コピーは、送信先リージョンのデフォルト KMS キーで常に暗号化されます。

[2] デフォルトポリシーとカスタムポリシーの場合:

- ターゲットインスタンスまたはボリュームが削除された場合、Amazon Data Lifecycle Manager は、保持期間に基づいて、最後の 1 つ前のスナップショットまで削除し続けます。デフォルトポリシーでは、削除を拡張して、最後のスナップショットを含めるようにできます。
- ポリシーが削除されるか、エラーまたは無効状態になると、Amazon Data Lifecycle Manager はスナップショットの削除を停止します。デフォルトポリシーでは、削除を拡張して、最後の 1 つを含むスナップショットの削除を続行できます。

EBS-backed AMI ポリシーの比較

次の表は、EBS-backed AMI のデフォルトポリシーとカスタム EBS-backed AMI ポリシーの違いを示しています。

機能	EBS-backed AMI のデフォルトポリシー	カスタム EBS-backed AMI ポリシー
マネージドバックアップリソース	EBS-backed AMI	EBS-backed AMI
ターゲットリソースタイプ	インスタンス	インスタンス
リソースターゲット設定	最新の AMI がないリージョン内のすべてのインスタンスをターゲットとします。除外パラメータを指定して特定のインスタンスを除外できます。	特定のタグを持つインスタンスのみをターゲットとします。
AMI 作成前のインスタンスの再起動	いいえ	はい
除外パラメータ	はい。特定のタグが付いたインスタンスを除外できます。	いいえ

機能	EBS-backed AMI のデフォルトポリシー	カスタム EBS-backed AMI ポリシー
複数のスケジュールのサポート	いいえ	はい。ポリシーごとに最大 4 つのスケジュールまで。
AMI の作成頻度	1 ~ 7 日おき。	毎日、毎週、毎月、毎年、または cron 式を使用したカスタム頻度。
サポートされている保持タイプ	経過期間ベースの保持のみ。	経過期間ベースの保持とカウントベースの保持。
AMI 保持	2 ~ 14 日。	最大 1000 個 の AMI (カウントベース) または最大 100 年 (経過期間ベース)。
AMI の非推奨サポート	いいえ	はい
クロスリージョンでのコピーのサポート	はい。デフォルト設定を使用 [1]	はい。カスタム設定を使用
拡張削除のサポート [2]	はい	いいえ

[1] デフォルトポリシーの場合:

- タグをクロスリージョンコピーにコピーすることはできません。
- コピーには、ソース AMI と同じ保持期間が使用されます。
- コピーには、ソース AMI と同じ暗号化状態を適用します。送信先リージョンで暗号化がデフォルトで有効になっている場合、ソース AMI が暗号化されていない場合でも、コピーは常に暗号化されます。コピーは、送信先リージョンのデフォルト KMS キーで常に暗号化されます。

[2] デフォルトポリシーとカスタムポリシーの場合:

- ターゲットインスタンスが終了した場合、Amazon Data Lifecycle Manager は、保持期間に基づいて、最後の 1 つ前の AMI まで登録解除し続けます。デフォルトポリシーでは、登録解除を拡張して、最後の AMI を含めるようにできます。
- ポリシーが削除されるか、エラーまたは無効状態になると、Amazon Data Lifecycle Manager は AMI の登録解除を停止します。デフォルトポリシーでは、削除を拡張して、最後の 1 つを含む AMI の登録解除を続行できます。

Amazon Data Lifecycle Manager のデフォルトポリシーを作成する

インスタンスから EBS-backed AMI を定期的に作成する場合、EBS-backed AMI のデフォルトポリシーを使用します。アタッチ状態に関係なくすべてのボリュームのスナップショットを作成する場合、または特定のボリュームを除外する場合は、EBS スナップショットのデフォルトポリシーを使用します。

このセクションでは、デフォルトポリシーを作成する方法について説明します。

トピック

- デフォルトポリシーに関する考慮事項
- Amazon EBS スナップショットのデフォルトポリシーを作成する
- EBS-backed AMI のデフォルトポリシーを作成する
- アカウントとリージョン間で Data Lifecycle Manager のデフォルトポリシーを有効にする

デフォルトポリシーに関する考慮事項

デフォルトポリシーを使用する際には、次の点に注意してください。

- デフォルトポリシーでは、最近のバックアップ (スナップショットまたは AMI) があるターゲットリソース (インスタンスまたはボリューム) をバックアップしません。作成頻度によって、どのリソースがバックアップされるかが決まります。ボリュームまたはインスタンスは、最新のスナップショットまたは AMI がポリシーの作成頻度よりも古い場合にのみバックアップされます。例えば、作成頻度を 3 日に指定した場合、EBS スナップショットのデフォルトポリシーでは、最新のスナップショットが 3 日より前のボリュームのスナップショットのみが作成されます。
- デフォルトでは、除外パラメータが指定されていない限り、デフォルトポリシーはそのリージョン内のすべてのインスタンスまたはボリュームをターゲットとします。
- デフォルトポリシーでは、一意のスナップショットの最小セットを作成します。例えば、EBS-backed AMI ポリシーと EBS スナップショットポリシーを有効にした場合、スナップショット

ポリシーでは、EBS-backed AMI ポリシーで既にバックアップされているボリュームのスナップショットを複製しません。

- デフォルトポリシーでは、24 時間以上経過したリソースのみをターゲットにし始めます。
- ボリュームを削除するか、デフォルトポリシーのターゲットとなるインスタンスを終了すると、Amazon Data Lifecycle Manager は、保持期間に従って、それまで作成されたバックアップ (スナップショットまたは AMI) を最後の 1 つ前のバックアップまで削除し続けます。必要がない場合は、このバックアップを手動で削除する必要があります。

  Amazon Data Lifecycle Manager で最後のバックアップを削除する場合は、[削除を延長] を有効にします。
- デフォルトポリシーが削除されるか、エラーまたは無効状態になると、Amazon Data Lifecycle Manager は以前に作成されたバックアップ (スナップショットまたは AMI) の削除を停止します。Amazon Data Lifecycle Manager で、最後の 1 つを含むバックアップの削除を続行する場合は、ポリシーを削除する前、またはポリシーの状態が「無効」または「削除済み」に変わる前に、[削除を延長] を有効にする必要があります。
- デフォルトポリシーを作成して有効にすると、Amazon Data Lifecycle Manager はターゲットリソースを 4 時間の時間ウィンドウにランダムに割り当てます。ターゲットリソースは、割り当てられたウィンドウ中に、指定した作成頻度でバックアップされます。例えば、ポリシーの作成頻度が 3 日で、ターゲットリソースが 12 時 ~ 16 時のウィンドウに割り当てられている場合、そのリソースは 3 日ごとに 12 時 ~ 16 時の間にバックアップされます。

Amazon EBS スナップショットのデフォルトポリシーを作成する

次の手順では、EBS スナップショットのデフォルトポリシーを作成する方法を示します。

Console

EBS スナップショットのデフォルトポリシーを作成するには

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. ナビゲーションパネルで、[ライフサイクルマネージャー]、[ライフサイクルポリシーの作成] の順に選択します。
3. [ポリシータイプ] で [デフォルトポリシー] を選択し、次に [EBS スナップショットポリシー] を選択します。
4. [説明] にポリシーの簡単な説明を入力します。
5. [IAM ロール] で、スナップショットを管理する許可を持つ IAM ロールを選択します。

Amazon Data Lifecycle Manager が提供するデフォルトの IAM ロールを使用する場合は、[デフォルト] を選択することをお勧めします。ただし、以前に作成したカスタム IAM ロールを使用することもできます。

6. [作成頻度] で、ポリシーを実行してボリュームのスナップショットを作成する頻度を指定します。

   指定する頻度によって、どのボリュームをバックアップするかも決まります。このポリシーにより、指定した頻度内に他の手段でバックアップされていないボリュームのみがバックアップされます。例えば、作成頻度を 3 日に指定した場合、このポリシーでは、過去 3 日以内にバックアップされていないボリュームのスナップショットのみが作成されます。

7. [保持期間] には、作成されたスナップショットをポリシーで保持する期間を指定します。スナップショットが保持しきい値に達すると、自動的に削除されます。保持期間は、作成頻度と同じか長くする必要があります。

8. (オプション) [除外パラメータ] を設定すると、スケジュールされたバックアップから特定のボリュームを除外できます。除外されたボリュームは、ポリシー実行時にバックアップされません。

   a. ブートボリュームを除外するには、[ブート・ボリュームを除外] を選択します。ブートボリュームを除外すると、データ (非ブート) ボリュームのみがポリシーでバックアップされます。つまり、インスタンスにブートボリュームとしてアタッチされているボリュームのスナップショットは作成されません。

   b. 特定のボリュームタイプを除外するには、[特定のボリュームタイプを除外] を選択し、除外するボリュームタイプを選択します。残りのタイプのボリュームのみがポリシーでバックアップされます。

   c. 特定のタグを持つボリュームを除外するには、[タグを追加] を選択し、タグのキーと値を指定します。このポリシーでは、指定したタグのいずれかを持つボリュームのスナップショットは作成されません。

9. (オプション) [詳細設定] で、ポリシーで実行する必要があるその他のアクションを指定します。

   a. 割り当てられたタグをソースボリュームからスナップショットにコピーするには、[ボリュームからタグをコピー] を選択します。

   b. [拡張削除] を無効にした場合:

      - ソースボリュームが削除された場合、Amazon Data Lifecycle Manager は、保持期間に基づいて、それまで作成された、最後の 1 つ前のスナップショットまで削除し続

けます。Amazon Data Lifecycle Manager で、最後の 1 つを含むすべてのスナップショットを削除する場合は、[拡張削除] を選択します。

- ポリシーが削除されるか、`error` または `disabled` 状態になると、Amazon Data Lifecycle Manager はスナップショットの削除を停止します。Amazon Data Lifecycle Manager で、最後の 1 つを含むスナップショットの削除を続行する場合は、[拡張削除] を選択します。

> **Note**
>
> [拡張削除] を有効にすると、上記の両方の動作が同時にオーバーライドされます。

c. ポリシーによって作成されたスナップショットを他のリージョンにコピーするには、[クロスリージョンコピーを作成] を選択し、送信先リージョンを最大 3 つ選択します。

- ソーススナップショットが暗号化されている場合、または送信先リージョンで暗号化がデフォルトで有効になっている場合には、コピーされたスナップショットは、送信先リージョンの EBS 暗号化用のデフォルト KMS キーを使用して暗号化されます。
- ソーススナップショットが暗号化されておらず、送信先リージョンで暗号化がデフォルトで無効になっている場合、コピーされたスナップショットは暗号化されません。

10. (オプション) ポリシーにタグを追加するには、[タグを追加] を選択し、タグのキーと値のペアを指定します。
11. [デフォルトポリシーの作成] を選択します。

> **Note**
>
> `Role with name AWSDataLifecycleManagerDefaultRole already exists` エラーが発生した場合、詳細については「Amazon Data Lifecycle Manager の問題のトラブルシューティング」を参照してください。

AWS CLI

EBS スナップショットのデフォルトポリシーを作成するには

create-lifecycle-policy コマンドを使用します。ユースケースやプリファレンスに応じて、次の 2 つの方式のいずれかでリクエストパラメータを指定できます。

- 方式 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"
```

例えば、リージョン内のすべてのボリュームをターゲットとし、デフォルトの IAM ロールを使用し、毎日実行し (デフォルト)、スナップショットを 7 日間保持する (デフォルト) という設定の EBS スナップショットのデフォルトポリシーを作成する場合は、次のパラメータを指定する必要があります。

```
$ 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
```

- 方式 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
```

ここで policyDetails.json には以下が含まれます。

```
{
    "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"
        }]
    }
}
```

EBS-backed AMI のデフォルトポリシーを作成する

次の手順では、EBS-backed AMI のデフォルトポリシーを作成する方法を示します。

Console

EBS-backed AMI のデフォルトポリシーを作成するには

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. ナビゲーションパネルで、[ライフサイクルマネージャー]、[ライフサイクルポリシーの作成] の順に選択します。
3. [ポリシータイプ] に [デフォルトポリシー] を選択し、次に [EBS-backed AMI ポリシー] を選択します。
4. [説明] にポリシーの簡単な説明を入力します。
5. [IAM ロール] で、AMI を管理する許可を持つ IAM ロールを選択します。

   Amazon Data Lifecycle Manager が提供するデフォルトの IAM ロールを使用する場合は、[デフォルト] を選択することをお勧めします。ただし、以前に作成したカスタム IAM ロールを使用することもできます。
6. [作成頻度] では、ポリシーを実行してインスタンスから AMI を作成する頻度を指定します。

指定する頻度によって、どのインスタンスがバックアップされるかも決まります。このポリシーにより、指定した頻度内に他の手段でバックアップされていないインスタンスのみがバックアップされます。例えば、作成頻度を 3 日に指定した場合、このポリシーでは、過去 3 日以内にバックアップされていないインスタンスからのみ AMI が作成されます。

7. [保持期間] には、作成された AMI をポリシーで保持する期間を指定します。AMI が保持しきい値に達すると、その AMI は自動的に登録解除され、関連付けられているスナップショットが削除されます。保持期間は、作成頻度と同じか長くする必要があります。
8. (オプション) [除外パラメータ] を設定すると、スケジュールされたバックアップから特定のインスタンスを除外できます。除外されたインスタンスは、ポリシー実行時にバックアップされません。

   - 特定のタグを持つインスタンスを除外するには、[タグを追加] を選択し、タグのキーと値を指定します。このポリシーでは、指定したタグのいずれかを持つインスタンスからは AMI が作成されません。
9. (オプション) [詳細設定] で、ポリシーで実行する必要があるその他のアクションを指定します。

   a. 割り当てられたタグをソースインスタンスからその AMI にコピーするには、[インスタンスからタグをコピー] を選択します。

   b. [拡張削除] を無効にした場合:

      - ソースインスタンスが終了した場合、Amazon Data Lifecycle Manager は、保持期間に基づいて、それまで作成された、最後の 1 つ前の AMI まで登録解除し続けます。Amazon Data Lifecycle Manager で、最後の 1 つを含むすべての AMI の登録を解除する場合は、[拡張削除] を選択します。
      - ポリシーが削除されるか、`error` または `disabled` 状態になると、Amazon Data Lifecycle Manager は AMI の登録解除を停止します。Amazon Data Lifecycle Manager で、最後の 1 つを含む AMI の登録解除を続行する場合は、[拡張削除] を選択します。

      > **Note**
      >
      > 拡張削除を有効にすると、上記の両方の動作が同時にオーバーライドされます。

   c. ポリシーによって作成された AMI を他のリージョンにコピーするには、[クロスリージョンコピーを作成] を選択し、送信先リージョンを最大 3 つ選択します。

- ソース AMI が暗号化されている場合、または送信先リージョンで暗号化がデフォルトで有効になっている場合には、コピーされた AMI は、送信先リージョンの EBS 暗号化用のデフォルト KMS キーを使用して暗号化されます。
- ソース AMI が暗号化されておらず、送信先リージョンで暗号化がデフォルトで無効になっている場合、コピーされた AMI は暗号化されません。

10. (オプション) ポリシーにタグを追加するには、[タグを追加] を選択し、タグのキーと値のペアを指定します。
11. [デフォルトポリシーの作成] を選択します。

> **Note**
>
> `Role with name AWSDataLifecycleManagerDefaultRoleForAMIManagement already exists` エラーが発生した場合、詳細については「Amazon Data Lifecycle Manager の問題のトラブルシューティング」を参照してください。

AWS CLI

EBS-backed AMI のデフォルトポリシーを作成するには

create-lifecycle-policy コマンドを使用します。ユースケースやプリファレンスに応じて、次の 2 つの方式のいずれかでリクエストパラメータを指定できます。

- 方式 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}]
```

例えば、リージョン内のすべてのインスタンスをターゲットとし、デフォルトの IAM ロールを使用し、毎日実行し (デフォルト)、AMI を 7 日間保持する (デフォルト) という設定の EBS-backed AMI のデフォルトポリシーを作成する場合は、次のパラメータを指定する必要があります。

```
$ 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
```

- 方式 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
```

ここで policyDetails.json には以下が含まれます。

```
{
    "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"
        }]
    }
}
```

アカウントとリージョン間で Data Lifecycle Manager のデフォルトポリシーを有効にする

CloudFormation StackSets を使用すると、1 回のオペレーションで複数のアカウントと AWS リージョンで Amazon Data Lifecycle Manager のデフォルトポリシーを有効にできます。

StackSets を使用して、次のいずれかの方法でデフォルトポリシーを有効にすることができます。

- AWS 組織全体 — 組織全体または AWS 組織内の特定の組織単位にわたって、デフォルトのポリシーが一貫して有効になり、設定されていることを確認します。これは、サービス管理のアクセス許可を使用して行われます。 CloudFormation StackSets は、ユーザーに代わって必要な IAM ロールを作成します。
- 特定の AWS アカウント全体 — 特定のターゲットアカウント間でデフォルトポリシーが一貫して有効になり、設定されていることを確認します。これには、セルフマネージドアクセス許可が必要です。StackSet 管理者アカウントとターゲットアカウント間の信頼関係を確立するために必要な IAM ロールを作成します。

詳細については、「AWS CloudFormation ユーザーガイド」の「スタックセット用のアクセス許可モデル」を参照してください。

次の手順を使用して、 AWS 組織全体、特定の OUs、または特定のターゲットアカウントで Amazon Data Lifecycle Manager のデフォルトポリシーを有効にします。

前提条件

デフォルトポリシーを有効にする方法に応じて、以下のいずれかを実行します。

- (AWS 組織全体) 組織内のすべての機能を有効にし、 で信頼されたアクセスを有効にする AWS Organizations必要があります。組織の管理アカウントまたは委任管理者アカウントにサインインする必要があります。
- (特定のターゲットアカウント全体) StackSet 管理者アカウントとターゲットアカウントの間に信頼関係を確立するために必要なロールを作成して、セルフマネージドアクセス許可を付与する必要があります。

Console

AWS 組織全体または特定のターゲットアカウントでデフォルトポリシーを有効にするには

1. https://console.aws.amazon.com/cloudformation で CloudFormation コンソールを開きます。
2. ナビゲーションペインで [StackSets]、[StackSet を作成] の順に選択します。
3. [アクセス許可] で、デフォルトのポリシーを有効にする方法に応じて、次のいずれかを実行します。
   - (AWS 組織全体) サービス管理のアクセス許可を選択します。
   - (特定のターゲットアカウント全体) [セルフサービスアクセス許可] を選択します。次に、[IAM 管理ロール ARN] で管理者アカウント用に作成した IAM サービスロールを選択し、[IAM 実行ロール名] にターゲットアカウントで作成した IAM サービスロールの名前を入力します。
4. [テンプレートの準備] で [サンプルテンプレートを使用] を選択します。
5. [サンプルテンプレート] で次のいずれかを実行します。
   - (EBS スナップショットのデフォルトポリシー) [EBS スナップショットの Amazon Data Lifecycle Manager デフォルトポリシーの作成] を選択します。
   - (EBS-backed AMI のデフォルトポリシー) [EBS-backed AMI の Amazon Data Lifecycle Manager デフォルトポリシーの作成] を選択します。
6. [次へ] を選択します。
7. [StackSet 名] と [StackSet の説明] に、わかりやすい名前と簡単な説明を入力します。
8. [パラメータ] セクションで、必要に応じてデフォルトのポリシー設定を設定します。

   > **Note**
   > 重要なワークロードの場合、CreateInterval = 1 日、RetainInterval = 7 日をお勧めします。

9. [次へ] を選択します。
10. (オプション) [タグ] では、StackSet とスタックリソースを識別するのに役立つタグを指定します。
11. [マネージド型の実行] の場合は、[アクティブ] を選択します。
12. [次へ] を選択します。

13. [Add stacks to stack set] (スタックセットにスタックを追加) で、[Deploy new stacks] (新しいスタックのデプロイ) を選択します。
14. デフォルトポリシーを有効にする方法に応じて、以下のいずれかを実行します。
    - (AWS 組織全体) デプロイターゲットでは、次のいずれかのオプションを選択します。
      - AWS 組織全体にデプロイするには、組織へのデプロイを選択します。
      - 特定の組織単位 (OU) にデプロイするには、[組織単位 (OU) へのデプロイ] を選択し、[OU ID] に OU ID を入力します。OU を追加するには、[別の OU を追加] を選択します。
    - (特定のターゲットアカウント全体) [アカウント] の場合は、次のいずれかを実行します。
      - 特定のターゲットアカウントにデプロイするには、[スタックをアカウントにデプロイ] を選択し、[アカウント番号] にターゲットアカウントの ID を入力します。
      - 特定の OU 内のすべてのアカウントにデプロイするには、[スタックを組織単位内のすべてのアカウントにデプロイ] を選択し、[組織番号] にターゲット OU の ID を入力します。
15. [自動デプロイ] で、[アクティブ化済み] を選択します。
16. [アカウント削除の動作] で、[スタックを保持] を選択します。
17. [リージョンの指定] では、デフォルトポリシーを有効にする特定のリージョンを選択するか、[すべてのリージョンを追加] を選択してすべてのリージョンでデフォルトポリシーを有効にします。
18. [次へ] を選択します。
19. スタックセット設定を確認し、 が IAM リソースを作成する CloudFormation 可能性があることを確認してから、送信を選択します。

AWS CLI

AWS 組織全体でデフォルトポリシーを有効にするには

1. スタックセットを作成します。create-stack-set コマンドを使用します。

   `--permission-model` の場合、`SERVICE_MANAGED` を指定します。

   `--template-url` で、次のいずれかのテンプレート URL を指定します。

- (EBS-backed AMI のデフォルトポリシー) https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/DataLifecycleManagerAMIDefaultPolicy.yaml
- (EBS スナップショットのデフォルトポリシー) https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/DataLifecycleManagerEBSSnapshotDefaultPolicy.yaml

--parameters で、デフォルトポリシーの設定を指定します。サポートされているパラメータ、パラメータの説明、および有効な値については、URL を使用してテンプレートをダウンロードし、テキストエディタを使用してテンプレートを表示します。

--auto-deployment の場合、Enabled=true, RetainStacksOnAccountRemoval=true を指定します。

```
$ 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. スタックセットをデプロイします。create-stack-instances コマンドを使用します。

--stack-set-name で、前のステップで作成したスタックセットの名前を指定します。

--deployment-targets OrganizationalUnitIds で、組織全体にデプロイするルート OU の ID、または組織内の特定の OU にデプロイする OU ID を指定します。

で--regions、デフォルトポリシーを有効にする AWS リージョンを指定します。

```
$ 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"]'
```

特定のターゲットアカウント間でデフォルトポリシーを有効にするには

1. スタックセットを作成します。create-stack-set コマンドを使用します。

   --template-url で、次のいずれかのテンプレート URL を指定します。

   - (EBS-backed AMI のデフォルトポリシー) https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/DataLifecycleManagerAMIDefaultPolicy.yaml
   - (EBS スナップショットのデフォルトポリシー) https://s3.amazonaws.com/cloudformation-stackset-sample-templates-us-east-1/DataLifecycleManagerEBSSnapshotDefaultPolicy.yaml

   --administration-role-arn で、スタックセット管理者用に以前に作成した IAM サービスロールの ARN を指定します。

   --execution-role-name で、ターゲットアカウントで作成した IAM サービスロールの名前を指定します。

   --parameters で、デフォルトポリシーの設定を指定します。サポートされているパラメータ、パラメータの説明、および有効な値については、URL を使用してテンプレートをダウンロードし、テキストエディタを使用してテンプレートを表示します。

   --auto-deployment の場合、Enabled=true, RetainStacksOnAccountRemoval=true を指定します。

   ```
   $ 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. スタックセットをデプロイします。create-stack-instances コマンドを使用します。

   --stack-set-name で、前のステップで作成したスタックセットの名前を指定します。

   には--accounts、ターゲット AWS アカウントの IDs を指定します。

で--regions、デフォルトポリシーを有効にする AWS リージョンを指定します。

```
$ aws cloudformation create-stack-instances \
--stack-set-name stackset_name \
--accounts '["account_ID_1","account_ID_2"]' \
--regions '["region_1", "region_2"]'
```

EBS スナップショット用の Amazon Data Lifecycle Manager カスタムポリシーを作成

以下の手順では、Amazon Data Lifecycle Manager を使用して Amazon EBS スナップショットのライフサイクルを自動化する方法を示します。

トピック

- スナップショットライフサイクルポリシーを作成する
- スナップショットライフサイクルポリシーに関する考慮事項
- その他のリソース
- Data Lifecycle Manager を使用してアプリケーション整合性のあるスナップショットを自動化
- Data Lifecycle Manager の事前スクリプトと事後スクリプトのその他のユースケース
- Amazon Data Lifecycle Manager の事前スクリプトと事後スクリプトの仕組み
- Data Lifecycle Manager の事前スクリプトと事後スクリプトで作成されたスナップショットを特定
- Amazon Data Lifecycle Manager の事前スクリプトと事後スクリプトをモニタリング

スナップショットライフサイクルポリシーを作成する

スナップショットのライフサイクルポリシーを作成するには、次のいずれかの手順を使用します。

Console

スナップショットのポリシーを作成するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインで、[Elastic Block Store]、[ライフサイクルマネージャー]、[ライフサイクルポリシーの作成] の順に選択します。

3. リポジトリの [ポリシータイプの選択] 画面で、[EBS スナップショットポリシー] を選択し、[次へ] をクリックします。
4. [Target resources] (ターゲットリソース) セクションで、以下の操作を行います。

   a. [Target resource types] (ターゲットリソースタイプ) で、バックアップするリソースの種類を選択します。`Volume` を選択して個々のボリュームのスナップショットを作成するか、`Instance` を選択してインスタンスにアタッチされたボリュームからマルチボリュームスナップショットを作成します。

   b. (Outpost およびローカルゾーンのお客様のみ) ターゲットリソースが存在する場所を指定します。

      [ターゲットリソースの場所] で、ターゲットリソースが存在する場所を指定します。

      - リージョン内のリソースをターゲットにするには、[AWS リージョン] を選択します。Amazon Data Lifecycle Manager は、現在のリージョン内で、一致するターゲットタグを持つ、指定されたタイプのすべてのリソースをバックアップします。スナップショットは同じリージョンに作成されます。
      - ローカルゾーン内のリソースをターゲットにするには、[AWS ローカルゾーン] を選択します。Amazon Data Lifecycle Manager は、現在のリージョン内のすべてのローカルゾーンで、一致するターゲットタグを持つ、指定されたタイプのすべてのリソースをバックアップします。スナップショットは、ソースリソースと同じローカルゾーン、またはその親リージョンに作成できます。
      - Outpost のリソースをターゲットにするには、[AWS Outpost] を選択します。Amazon Data Lifecycle Manager は、アカウントのすべての Outposts で、一致するターゲットタグを持つ、指定されたタイプのすべてのリソースをバックアップします。スナップショットは、ソースリソースと同じ Outpost、またはその親リージョンに作成できます。

   c. [Target resource tags] (ターゲットリソースタグ) で、バックアップするボリュームもしくはインスタンスを識別する、リソースタグを選択します。ポリシーでは、指定されたタグキーと値のペアを持つリソースのみがバックアップされます。

5. [説明] にポリシーの簡単な説明を入力します。
6. [IAM ロール] では、スナップショットの管理、ならびにボリュームとインスタンスの記述に必要な、アクセス許可を持つ IAM ロールを選択します。Amazon Data Lifecycle Manager が提供するデフォルトのロールを使用するには、[デフォルトのロール] を選択します。以前に作成したカスタム IAM ロールを使用する場合には、[別のロールを選択] をクリックした上で、使用するロールを選択します。

7. [ポリシータグ] に、ライフサイクルポリシーに適用されるタグを追加します。これらのタグは、ポリシーを識別および分類するために使用することができます。
8. [Policy status] (ポリシーステータス) では、[Enable] (有効化) を選択すると、次のスケジュールした時刻にポリシーが実行されます。ポリシーが実行されないようにするには、[Disable policy] (ポリシーの無効化) を選択します。ここでポリシーを有効にしない場合、作成後に手動で有効にするまで、スナップショットの作成は開始されません。
9. (インスタンスのみをターゲットとするポリシー) ボリュームをマルチボリュームスナップショットセットから除外します。

   デフォルトでは、Amazon Data Lifecycle Manager は、ターゲットインスタンスにアタッチされたすべてのボリュームのスナップショットを作成します。ただし、アタッチされたボリュームのサブセットのスナップショットを作成することもできます。[Parameters] (パラメータ) セクションで、次を実行します。

   - ターゲットインスタンスにアタッチされたルートボリュームのスナップショットを作成しない場合は、[Exclude root volume] (ルートボリュームを除外) を選択します。このオプションを選択すると、ターゲットインスタンスにアタッチされているデータ (非ルート) ボリュームのみがマルチボリュームスナップショットセットに含まれます。
   - ターゲットインスタンスにアタッチされたデータ (非ルート) ボリュームのサブセットのスナップショットを作成する場合は、[Exclude specific data volumes] (特定のデータボリュームを除外) を選択し、スナップショットを作成しないデータボリュームを識別するために使用するタグを指定します。Amazon Data Lifecycle Manager は、指定されたタグのいずれかを含むデータボリュームのスナップショットを作成しません。Amazon Data Lifecycle Manager は、指定されたタグを持たないデータボリュームのスナップショットのみを作成します。
10. [次へ] を選択します。
11. [スケジュールの設定] 画面で、ポリシースケジュールを設定します。ポリシーには、最大 4 つのスケジュールを含めることができます。スケジュール 1 は必須です。スケジュール 2、3、および 4 はオプションです。追加したポリシースケジュールごとに、以下の操作を行います。

    a. [スケジュールの詳細] セクションで、次の操作を行います。

       i. [スケジュール名] で、スケジュールの分かりやすい名前を指定します。

       ii. [頻度]とそれに関連するフィールドで、ポリシーの実行間隔を設定します。

ポリシーの実行は、日次、週次、月次、年次のいずれかのスケジュールで設定できます。または、[カスタム cron 式] をクリックし、最長 1 年の間隔を指定します。詳細については、「Amazon EventBridge ユーザーガイド」の「Setting a schedule pattern for scheduled rules (legacy) in Amazon EventBridge」を参照してください。

> **Note**
>
> スケジュールに対しスナップショットのアーカイブを有効にする必要がある場合は、その頻度に、[monthly] (毎月) または [yearly] (毎年) のいずれかを選択する必要があります。または、作成頻度が 28 日以上の CRON 式を指定します。
> 頻度を毎月とし、特定の週の特定の日 (例えば、その月の第 2 木曜日) にスナップショットを作成する場合、カウントベースのスケジュールでは、アーカイブ階層の保存回数は 4 以上にする必要があります。

iii. [開始時刻] では、ポリシー実行の開始予定時刻を指定します。初回のポリシー実行は、予定時刻から 1 時間以内に開始されます。時刻は、hh:mm UTC 形式で入力する必要があります。

iv. [保持タイプ] では、スケジュールによって作成されるスナップショットの保持ポリシーを指定します。

スナップショットは、総数または期間に基づいて保持できます。

- カウントベースの保持
  - スナップショットアーカイブを無効にすると、範囲は 1 ~ 1000 になります。保持期間がしきい値に達すると、最も古いスナップショットは完全に削除されます。
  - スナップショットアーカイブを有効にすると、範囲は 0 (作成直後にアーカイブ) ~ 1000 になります。保持期間がしきい値に達すると、最も古いスナップショットは完全なスナップショットに変換され、アーカイブ階層に移動します。
- 年齢に基づく保持
  - スナップショットアーカイブを無効にすると、範囲は 1 日 ~ 100 年になります。保持期間がしきい値に達すると、最も古いスナップショットは完全に削除されます。

- スナップショットアーカイブを有効にすると、範囲は 0 日 (作成直後にアーカイブ) ~ 100 年になります。保持期間がしきい値に達すると、最も古いスナップショットは完全なスナップショットに変換され、アーカイブ階層に移動します。

> **Note**
>
> - すべてのスケジュールは、同じ保持タイプ (期間ベースまたはカウントベース) にする必要があります。保持タイプを指定できるのは、スケジュール 1 のみです。スケジュール 2、3、4 は、スケジュール 1 から保持タイプを継承します。各スケジュールには、独自の保持回数または期間を設定できます。
> - 高速スナップショット復元、クロスリージョンコピー、またはスナップショットの共有を有効にする場合は、保持回数で 1　以上を、または保持期間で 1 日以上を指定する必要があります。

v. (AWS Outposts および Local Zone のお客様のみ) スナップショットの送信先を指定します。

[スナップショットの送信先] で、ポリシーによって作成されるスナップショットの送信先を指定します。

- ポリシーがリージョン内のリソースをターゲットにしている場合、スナップショットは同じリージョンに作成する必要があります。 AWS リージョンは自動的に選択されます。
- ポリシーでローカルゾーン内のリソースがターゲットになっている場合は、ソースリソースと同じローカルゾーン、またはその親リージョンにスナップショットを作成する必要があります。
- ポリシーで Outpost 上のリソースがターゲットになっている場合は、ソースリソースと同じ Outpost、またはその親リージョンにスナップショットを作成する必要があります。

b. スナップショットのタグ付けを設定します。

[タグ付け] セクションで、以下を実行します。

i. ソースボリュームのすべてのユーザー定義タグを、スケジュールにより作成されたスナップショットにコピーするには、[ソースからタグをコピー] を選択します。

ii. 他のタグを指定し、このスケジュールによって作成されたスナップショットに割り当てるには、[タグを追加] をクリックします。

c. アプリケーション整合性のあるスナップショット用の事前スクリプトと事後スクリプトを設定します。

詳細については、「Data Lifecycle Manager を使用してアプリケーション整合性のあるスナップショットを自動化」を参照してください。

d. (ボリュームのみをターゲットとするポリシー) スナップショットアーカイブを設定します。

[スナップショットのアーカイブ] セクションで、次の操作を行います。

> **Note**
>
> スナップショットのアーカイブを有効にできるのは、ポリシー内で 1 つのスケジュールのみです。

i. スケジュールのスナップショットアーカイブを有効にするには、[Archive snapshots created by this schedule] (このスケジュールで作成されたスナップショットをアーカイブする) を選択します。

> **Note**
>
> スナップショットのアーカイブを有効にできるのは、スナップショットの作成頻度に毎月または毎年を設定した場合、または作成頻度が 28 日以上の cron 式を指定した場合のみです。

ii. アーカイブ層のスナップショットのための保存ルールを指定します。

- [count-based schedules] (カウントベースのスケジュール) で、アーカイブ層に保持するスナップショットの数を指定します。保持数のしきい値に達した場合は、最も古いスナップショットが完全にアーカイブ階層から削除されます。例えば、3 を指定した場合、スケジュールは最大 3 つのスナップショットをアーカイブ階層に保持します。4 番目のスナップショットをアーカイブする際には、アー

カイブ階層に既存の 3 つのスナップショットのうち、最も古いものが削除されます。

- [age-based schedules] (期間ベースのスケジュール) で、アーカイブ階層でのスナップショットの保持期間を指定します。保持数のしきい値に達した場合は、最も古いスナップショットが完全にアーカイブ階層から削除されます。例えば、120 日間を指定した場合、その期間に達したスナップショットは、スケジュールによりアーカイブ階層から自動的に削除されます。

> **Important**
>
> アーカイブされたスナップショットの最小保持期間は 90 日です。スナップショットを少なくとも 90 日間保持する保存ルールを指定する必要があります。

e. 高速スナップショット復元を有効にします。

スケジュールによって作成されたスナップショットで高速スナップショット復元を有効化するには、[高速スナップショット復元] セクションで、[スナップショットの高速復元を有効化する] を選択します。高速スナップショット復元を有効にする場合は、特定のアベイラビリティーゾーンを選択する必要があります。保存期間に基づく保持スケジュールを使用する場合は、各スナップショットに対して高速スナップショット復元を有効にする期間を指定する必要があります。スケジュールでカウントベースの保持を使用する場合は、高速スナップショット復元を有効にするスナップショットの最大数を指定する必要があります。

スケジュールで Outpost にスナップショットを作成している場合、高速スナップショット復元を有効にすることはできません。高速スナップショット復元は、Outpost に保存されているローカルスナップショットではサポートされません。

> **Note**
>
> 特定のアベイラビリティーゾーンでスナップショットの高速スナップショット復元を有効にしている時間中は、請求が発生します。料金は 1 時間を最小として時間単位で計算されます。

f. クロスリージョンコピーを設定します。

スケジュールによって作成されたスナップショットを、Outpost または別のリージョンにコピーするには、[クロスリージョンコピー] セクションで、[クロスリージョンコピーを有効化する] 選択します。

スケジュールによってリージョンにスナップショットが作成されている場合は、アカウント内で最大 3 つの異なるリージョンあるいは Outposts に対し、そのスナップショットをコピーできます。送信先となるリージョンまたは Outpost ごとに、個別のクロスリージョンコピーのためのルールを指定する必要があります。

リージョンまたは Outpost ごとに、異なる保持ポリシーを選択できます。また、すべてのタグをコピーするか、いずれのタグもコピーしないかを選択できます。ソーススナップショットが暗号化されている場合、またはデフォルトの暗号化が有効化されている場合には、コピーされたスナップショットも暗号化されます。ソーススナップショットが暗号化されていない場合には、暗号化できます。KMS キー を指定しない場合、スナップショットは、各送信先リージョンにおける EBS 暗号化用のデフォルト KMS キー を使用して暗号化されます。送信先リージョンで KMS キー を指定する場合、選択した IAM ロールには KMS キー へのアクセス権が必要です。

> **Note**
>
> リージョンあたりの同時スナップショットコピー数を超えないようにする必要があります。

ポリシーが Outpost にスナップショットを作成している場合、リージョンまたは別の Outpost に対し、そのスナップショットをコピーすることはできません。また、クロスリージョンコピー設定は使用できません。

g. クロスアカウントの共有を設定します。

クロスアカウント共有で、スケジュールによって作成されたスナップショットを他の AWS アカウントと自動的に共有するようにポリシーを設定します。以下の操作を実行します。

i. 他の AWS アカウントとの共有を有効にするには、クロスアカウント共有を有効にするを選択します。

ii. スナップショットを共有するアカウントを追加するには、[アカウントを追加] をクリックし、12 桁の AWS アカウント ID を入力した後、[追加] をクリックします。

iii. スナップショットの共有を特定の期間後に自動的に解除するには、[Unshare automatically] (共有を自動解除する) を選択します。スナップショットの共有の自動解除を選択した場合、自動的な共有解除が実行されるまでの期間は、ポリシーがスナップショットを保持する期間より長くすることはできません。例えば、ポリシーの保存設定でスナップショットが 5 日間保持される場合、スナップショットの共有を自動的に解除するまでの期間としてポリシーに設定できるのは、最大 4 日間です。これは、期間ベースおよび数値ベースのスナップショット保持設定を持つポリシーに適用されます。

自動的な共有解除を有効にしない場合、スナップショットは削除されるまで共有されます。

> **Note**
>
> 共有できるのは、暗号化されていないスナップショットまたはカスタマーマネージド型キーを使用して暗号化されたスナップショットだけです。デフォルトの EBS 暗号化 KMS キー で暗号化されたスナップショットを共有することはできません。暗号化されたスナップショットを共有する場合は、ソースボリュームの暗号化に使用された KMS キー も、ターゲットアカウントと共有する必要があります。詳細については、「AWS Key Management Service デベロッパーガイド」の「他のアカウントのユーザーに KMS キーの使用を許可する」を参照してください。

h. 新たにスケジュールを追加するには、画面の上部にある [他のスケジュールを追加する] をクリックします。追加スケジュールごとに、このトピックの説明にならってフィールドを設定します。

i. 必要なスケジュールを追加したら、[ポリシーをレビュー] をクリックします。

12. ポリシーの概要を確認した後、[ポリシーを作成] をクリックします。

> **Note**
>
> `Role with name AWSDataLifecycleManagerDefaultRole already exists` エラーが発生した場合、詳細については「Amazon Data Lifecycle Manager の問題のトラブルシューティング」を参照してください。

Command line

スナップショットのライフサイクルポリシーを作成するには、create-lifecycle-policy コマンドを使用します。PolicyType で、EBS_SNAPSHOT_MANAGEMENT を指定する。

> **Note**
>
> 構文を簡略化するために、次の例では、ポリシーの詳細を含む JSON ファイル、policyDetails.json を使用しています。

例 1 – 2 つのスケジュールを持つスナップショットライフサイクルポリシー

この例では、値が 115 で costcenter のタグキーを持つすべてのボリュームのスナップショットを作成するスナップショットライフサイクルポリシーを作成します。ポリシーには、2 つのスケジュールが含まれます。最初のスケジュールでは、毎日 03:00 UTC にスナップショットが作成されます。2 番目のスケジュールでは、毎週金曜日の 17:00 (UTC) に週次のスナップショットが作成されます。

```
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
```

次は、policyDetails.json ファイルの例です。

```
{
    "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
    }
]}
```

リクエストが成功すると、コマンドは新しく作成されたポリシーの ID を返します。以下は出力の例です。

```
{
   "PolicyId": "policy-0123456789abcdef0"
}
```

例 2 - インスタンスをターゲットとし、データ (非ルート) ボリュームのサブセットのスナップショットを作成するスナップショットライフサイクルポリシー

この例では、code=production でタグ付けされたインスタンスからマルチボリュームスナップショットセットを作成するスナップショットライフサイクルポリシーを作成します。ポリシー

には 1 つのスケジュールのみが含まれます。スケジュールでは、code=temp でタグ付けされたデータボリュームのスナップショットは作成されません。

```
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
```

次は、policyDetails.json ファイルの例です。

```
{
    "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
    }
]}
```

リクエストが成功すると、コマンドは新しく作成されたポリシーの ID を返します。以下は出力の例です。

```
{
   "PolicyId": "policy-0123456789abcdef0"
}
```

例 3 — Outpost リソースでのローカルスナップショットを自動化する、スナップショットライフサイクルポリシー

この例では、すべての Outposts について、team=dev によりタグ付けされたボリュームのスナップショットを作成する、スナップショットライフサイクルポリシーを作成します。このポリシーにより、ソースボリュームと同じ Outposts にスナップショットが作成されます。このポリシーによるスナップショットの作成は、00:00 UTCから開始され、その後 12 時間ごとに実行されます。

```
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
```

次は、policyDetails.json ファイルの例です。

```
{
    "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
    }
]}
```

例 4 — リージョンにスナップショットを作成し、それを Outpost にコピーするスナップショットライフサイクルポリシー

次のポリシー例では、team=dev によりタグ付けされたボリュームのスナップショットを作成します。スナップショットは、ソースボリュームと同じリージョンに作成されます。スナップショットの作成は、00:00 UTC から開始され、その後 12 時間ごとに実行されます。スナップショットは最大 1 個まで保持されます。また、このポリシーでは、スナップショットを Outpost arn:aws:outposts:us-east-1:123456789012:outpost/op-1234567890abcdef0 にコピーし、デフォルトの暗号化 KMS キー を使用しながら、コピーされたスナップショットの暗号化も行います。コピーの保持期間は 1 か月間です。

```
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
```

次は、policyDetails.json ファイルの例です。

```
{
    "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"
            }
        }]
    }
]}
```

例 5 - アーカイブが有効な期間ベースのスケジュールを持つスナップショットライフサイクルポリシー

この例では、Name=Prod によりタグ付けされたボリュームを対象とする、スナップショットライフサイクルポリシーを作成します。このポリシーには、毎月の初日の午前 9 時にスナップショットを作成する、期間ベースのスケジュールが 1 つあります。スケジュールは、各スナップショットをアーカイブ階層に移動した後も、そのスナップショットを標準階層に 1 日間保持します。スナップショットは、90 日間アーカイブ階層に保持された後に削除されます。

```
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
```

次は、policyDetails.json ファイルの例です。

```
{
    "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"
      }
    ]
}
```

例 6 - アーカイブが有効なカウントベースのスケジュールを持つスナップショットライフサイクルポリシー

この例では、Purpose=Test によりタグ付けされたボリュームを対象とする、スナップショットライフサイクルポリシーを作成します。このポリシーには、毎月の初日の午前 9 時にスナップショットを作成する、カウントベースのスケジュールが 1 つあります。スケジュールは、作成直後にスナップショットをアーカイブし、最大 3 つのスナップショットをアーカイブ階層に保持します。

```
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
```

次は、policyDetails.json ファイルの例です。

```
{
    "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"
      }
    ]
}
```

スナップショットライフサイクルポリシーに関する考慮事項

スナップショットのライフサイクルポリシーには、次の一般的な考慮事項が適用されます。

- スナップショットライフサイクルポリシーは、ポリシーと同じリージョンにあるインスタンスまたはボリュームのみを対象としています。
- 最初のスナップショット作成オペレーションは、指定された開始時刻から 1 時間以内に開始されます。その後に続くスナップショット作成オペレーションは、スケジュールされた時刻の 1 時間以内に開始されます。
- ボリュームまたはインスタンスをバックアップするために複数のポリシーを作成できます。例えば、ボリュームに 2 つのタグがあり、タグ A が 12 時間ごとにスナップショットを作成するポリシー A のターゲットであり、タグ B が 24 時間ごとにスナップショットを作成するポリシー B のターゲットである場合、Amazon Data Lifecycle Manager は両方のポリシーのスケジュールに従ってスナップショットを作成します。または、複数のスケジュールを持つ単一のポリシーを作成することで、同じ結果を得ることができます。例えば、タグ A のみをターゲットとするポリシーを 1 つ作成し、スケジュールを 2 つ指定できます (1 つは 12 時間ごと、1 つは 24 時間ごと)。
- ターゲットリソースタグでは大文字と小文字が区別されます。
- ポリシーによってターゲットにされたリソースからターゲットタグを削除した場合、以降、Amazon Data Lifecycle Manager では、標準階層とアーカイブ層に既に存在するスナップショットの管理は行いません。不要になった場合は手動で削除する必要があります。
- インスタンスをターゲットとするポリシーを作成し、ポリシーの作成後に新しいボリュームがターゲットインスタンスにアタッチされた場合、新しく追加されたボリュームは、次回のポリシー実行時にバックアップに含まれます。ポリシー実行時にインスタンスにアタッチされたすべてのボリュームが含まれます。
- スナップショットを 1 つだけ作成するように設定されているカスタム cron ベースのスケジュールを持つポリシーを作成した場合、そのポリシーでは、保持のしきい値に達しても、そのスナップショットは自動的に削除されません。スナップショットが不要になった場合は、手動で削除する必要があります。
- 保持期間が作成頻度よりも短い経過日ベースのポリシーを作成した場合、Amazon Data Lifecycle Manager は次のスナップショットが作成されるまで常に最新のスナップショットを保持します。

例えば、経過日ベースのポリシーで保存期間が 7 日間のスナップショットが毎月 1 つ作成される場合、Amazon Data Lifecycle Manager は、保持期間が 7 日間であっても、各スナップショットを 1 か月間保持します。

スナップショットの共有には、次の考慮事項が適用されます。

- スナップショットのアーカイブは、ボリュームをターゲットとするスナップショットポリシーに対してのみ有効にできます。
- アーカイブルールは、ポリシーごとに 1 つのスケジュールでのみ指定が可能す。
- コンソールを使用している場合、スケジュールに設定されている作成頻度が、毎月または毎年、または 28 日以上の cron 式の場合にのみ、スナップショットアーカイブを有効にできます。

  AWS CLI、 AWS API、または AWS SDK を使用している場合、スケジュールに作成頻度が 28 日以上の cron 式がある場合にのみ、スナップショットのアーカイブを有効にできます。
- アーカイブ階層における最小保持期間は 90 日です。
- アーカイブされるスナップショットは、アーカイブ階層に移動される際に完全なスナップショットに変換されます。これにより、スナップショットのストレージコストが高くなる可能性があります。詳細については、「Amazon EBS スナップショットをアーカイブするための料金と請求」を参照してください。
- スナップショットのアーカイブを使用する場合、スナップショットの高速復元とスナップショット共有は無効になります。
- うるう年のために、保持ルールによるアーカイブの保持期間が 90 日到達できない場合、Amazon Data Lifecycle Manager が、スナップショットの保持期間が最低 90 日間になるように調整します。
- Amazon Data Lifecycle Manager によって作成されたスナップショットを手動でアーカイブし、スケジュールによる保持期間のしきい値を超えても依然としてアーカイブされている場合には、そのスナップショットに対して Amazon Data Lifecycle Manager による管理は行われません。ただし、スケジュールによる保持期間のしきい値に達する前にスナップショットを標準階層に復元すれば、そのスナップショットに対し、引き続きスケジュールの保存ルールに従った管理が行われます。
- 標準階層に永続的または一時的に復元された (Amazon Data Lifecycle Manager がアーカイブを行った) スナップショットが、スケジュールでの保持期間のしきい値に達した後も標準階層に残っている場合、そのスナップショットに対する Amazon Data Lifecycle Manager による管理は行われなくなります。ただし、保持期間のしきい値に達する前にスナップショットを再アーカイブすると、この保持期間が終了した時点で、スナップショットはスケジュールにより削除されます。

- Amazon Data Lifecycle Manager によってアーカイブされたスナップショットは、`Archived snapshots per volume` および `In-progress snapshot archives per account` のクォータにカウントされます。
- 24 時間再試行した後も、スケジュールがスナップショットをアーカイブできない場合、スナップショットは標準階層に残されます。その後、アーカイブ層から削除される予定時刻に基づいて削除されるようにスケジュールされます。例えば、スナップショットを 120 日間アーカイブするスケジュールでは、アーカイブが失敗してもスナップショットが 120 日間は標準階層に残され、その後完全に削除されます。カウントベースのスケジュールの場合、スナップショットはスケジュールによるの保持回数にカウントされません。
- スナップショットは、それが作成されたリージョンと同じリージョンにアーカイブされる必要があります。クロスリージョンコピーとスナップショットアーカイブを有効にしている場合、このスナップショットのコピーは Amazon Data Lifecycle Manager によりアーカイブされません。
- Amazon Data Lifecycle Manager によってアーカイブされたスナップショットには、`aws:dlm:archived=true` システムタグが付けられます。さらに、アーカイブが有効な期間ベースのスケジュールで作成されたスナップショットには、`aws:dlm:expirationTime` システムタグ (スナップショットがアーカイブされる予定の日付と時刻を示します) が付けられます。

ルートボリュームとデータ (非ルート) ボリュームの除外には、次の考慮事項が適用されます。

- ブートボリュームを除外することを選択し、結果としてインスタンスにアタッチされたすべての追加データボリュームを除外するタグを指定した場合、Amazon Data Lifecycle Manager は、影響を受けるインスタンスのスナップショットを作成せず、`SnapshotsCreateFailed` CloudWatch メトリクスを発行します。詳細については、「CloudWatch を使用して、ポリシーをモニタリング」を参照してください。

スナップショットライフサイクルポリシーによってターゲットにされたボリュームを削除したり、インスタンスを終了したりする場合の考慮事項は次のとおりです。

- 数値ベースの保持期間が設定されたポリシーでターゲットされているボリュームの削除やインスタンスの終了を行った場合、以後 Amazon Data Lifecycle Manager では、これらのボリュームまたはインスタンスで作成されたスナップショットの管理は行いません。前に作成されたこれらのスナップショットが不要になった場合、手動で削除する必要があります。
- 期間ベースの保持スケジュールが設定されたポリシーによってターゲットされているボリュームの削除やインスタンスの終了を行っても、ポリシーは、削除されたボリュームまたは終了されたインスタンスから以前に作成されたスナップショットを、 定義されたスケジュールに基づいて順番に

すべて (ただし、最新のものは除き) 削除し続けます。最後のスナップショットが不要になった場合は、手動で削除する必要があります。

スナップショットライフサイクルポリシーや高速スナップショット復元に関する考慮事項は次のとおりです。

- Amazon Data Lifecycle Manager は、サイズが 16 TiB 以下のスナップショットに対してのみ高速スナップショット復元を有効にできます。詳細については、「Amazon EBS 高速スナップショット復元」を参照してください。
- 高速スナップショット復元が有効化されているスナップショットについては、対応するポリシーを削除もしくは無効化した場合、対応するポリシーの高速スナップショット復元を無効化した場合、または対応するアベイラビリティーゾーンの高速スナップショット復元を無効化した場合であっても、当該復元は有効に保たれます。このようなスナップショットについては、手動で高速スナップショット復元を無効化する必要があります。
- ポリシーの高速スナップショット復元の有効化中に、有効化できるスナップショットの最大数を超えると、Amazon Data Lifecycle Manager はスケジュールに沿ったスナップショット作成は行うものの、作成したスナップショットの高速スナップショット復元は有効化しません。高速スナップショット復元が有効化されているスナップショットが削除されると、その次にAmazon Data Lifecycle Managerが作成するスナップショットの高速スナップショット復元が有効化されます。
- あるスナップショットの高速スナップショット復元を有効化すると、当該スナップショットが最適化されるまでに、1 TiB あたり 60 分の時間がかかります。Amazon Data Lifecycle Manager が次のスナップショットを作成する前に各スナップショットが完全に最適化されるようなスケジュールの設定をお勧めします。
- インスタンスを対象とするポリシーの高速スナップショット復元を有効にすると、Amazon Data Lifecycle Manager は、マルチボリュームスナップショットセット内の各スナップショットの高速スナップショット復元を個別に有効にします。Amazon Data Lifecycle Manager が、マルチボリュームスナップショットセット内のスナップショットの 1 つに対して高速スナップショット復元を有効にできない場合でも、スナップショットセット内の残りのスナップショットに対して高速スナップショット復元を有効にしようとします。
- 特定のアベイラビリティーゾーンでスナップショットの高速スナップショット復元を有効にしている時間中は、請求が発生します。料金は 1 時間を最小として時間単位で計算されます。詳細については、「料金と請求」を参照してください。

> **Note**
>
> ライフサイクルポリシーの設定によっては、複数のスナップショットに対して同時に複数のアベイラビリティーゾーンで高速スナップショット復元を有効にすることができます。

スナップショットライフサイクルポリシーおよびマルチアタッチが有効なボリュームに関する考慮事項は次のとおりです。

- マルチアタッチが有効な同じボリュームを持つインスタンスをターゲットとするライフサイクルポリシーを作成する場合、Amazon Data Lifecycle Manager はアタッチされたインスタンスごとにボリュームのスナップショットを開始します。timestamp タグを使用して、アタッチされたインスタンスから作成された時間整合性のあるスナップショットのセットを識別します。

アカウント間でスナップショットを共有する場合の考慮事項は次のとおりです。

- 共有できるのは、暗号化されていないスナップショットまたは カスタマーマネージド型キー を使用して暗号化されたスナップショットだけです。
- デフォルトの EBS 暗号化 KMS キーで暗号化されたスナップショットを共有することはできません。
- 暗号化されたスナップショットを共有する場合は、ソースボリュームの暗号化に使用された KMS キーも、ターゲットアカウントと共有する必要があります。詳細については、AWS Key Management Service デベロッパーガイドの「他のアカウントのユーザーに KMS キーの使用を許可する」を参照してください。

スナップショットのポリシーやスナップショットのアーカイブに関する考慮事項は次のとおりです。

- ポリシーによって作成されたスナップショットを手動でアーカイブし、ポリシーの保持しきい値に達したときにそのスナップショットがアーカイブ階層にある場合、Amazon Data Lifecycle Manager はスナップショットを削除しません。Amazon Data Lifecycle Manager は、スナップショットがアーカイブ階層に保存されている間は、スナップショットを管理しません。アーカイブ階層に保存されているスナップショットが不要になった場合は、手動で削除する必要があります。

次の考慮事項は、スナップショットポリシーおよびごみ箱に適用されます。

- Amazon Data Lifecycle Manager がポリシーの保持しきい値に達したときにスナップショットを削除してごみ箱に移動し、そのスナップショットをごみ箱から手動で復元した場合は、スナップショットが不要になったら手動で削除する必要があります。Amazon Data Lifecycle Manager は、スナップショットを管理しなくなります。
- ポリシーによって作成されたスナップショットを手動で削除し、ポリシーの保持しきい値に達したときにそのスナップショットがごみ箱にある場合、Amazon Data Lifecycle Manager はスナップショットを削除しません。Amazon Data Lifecycle Manager は、スナップショットがごみ箱に保存されている間は、スナップショットを管理しません。

  ポリシーの保持しきい値に達する前にスナップショットがごみ箱から復元された場合、Amazon Data Lifecycle Manager は、ポリシーの保持しきい値に達したときにスナップショットを削除します。

  ポリシーの保持しきい値に達した後にスナップショットがごみ箱から復元された場合、Amazon Data Lifecycle Manager はそのスナップショットを削除しません。スナップショットが不要になった場合は、手動で削除する必要があります。

以下は、エラー状態にあるスナップショットライフサイクルポリシーに関する考慮事項です。

- 期間ベースの保持スケジュールを持つポリシーの場合、ポリシーが `error` 状態の間に有効期限を迎えるスナップショットは無期限に保持されます。これらのスナップショットは手動で削除する必要があります。ポリシーを再度有効にすると、Amazon Data Lifecycle Manager は保持期間が終了した時にスナップショットの削除を再開します。
- カウントベースの保存スケジュールが設定されているポリシーの場合、ポリシーが `error` 状態の間はスナップショットの作成と削除が停止されます。ポリシーを再度有効にすると、Amazon Data Lifecycle Manager はスナップショットの作成を再開し、保持しきい値に達した時にスナップショットの削除を再開します。

スナップショットポリシーとスナップショットロックに関する考慮事項は次のとおりです。

- Amazon Data Lifecycle Manager によって作成されたスナップショットを手動でロックし、その後保持しきい値を超えても依然としてロックされている場合、そのスナップショットについては Amazon Data Lifecycle Manager による管理は行われません。スナップショットが不要になった場合は、手動で削除する必要があります。
- Amazon Data Lifecycle Manager によって作成され、高速スナップショット復元が有効化されているスナップショットを手動でロックし、その後保持しきい値を超えても依然としてロックされている場合、そのスナップショットについては Amazon Data Lifecycle Manager による高速スナップ

ショット復元の無効化または削除は行われません。スナップショットが不要になった場合は、手動で高速スナップショット復元を無効にして削除する必要があります。

- Amazon Data Lifecycle Manager によって作成されたスナップショットを AMI に手動で登録してからロックし、その後保持しきい値を超えても依然としてロックされた状態で AMI に関連付けられている場合、そのスナップショットについては Amazon Data Lifecycle Manager による削除の試行が続行されます。AMI の登録が解除され、スナップショットのロックが解除されると、Amazon Data Lifecycle Manager はスナップショットを自動的に削除します。

その他のリソース

詳細については、「Amazon Data Lifecycle Manager ストレージを使用した Amazon EBS スナップショットと AMI 管理の自動化 AWS 」ブログを参照してください。

Data Lifecycle Manager を使用してアプリケーション整合性のあるスナップショットを自動化

Amazon Data Lifecycle Manager では、インスタンスをターゲットとするスナップショットライフサイクルポリシーで事前スクリプトと事後スクリプトを有効にすることで、アプリケーション整合性のあるスナップショットを自動化できます。

Amazon Data Lifecycle Manager は AWS Systems Manager (Systems Manager) と統合して、アプリケーション整合性のあるスナップショットをサポートします。Amazon Data Lifecycle Manager は、事前スクリプトと事後スクリプトを含む Systems Manager (SSM) コマンドドキュメントを使用して、アプリケーション整合性のあるスナップショットを完成させるために必要なアクションを自動化します。Amazon Data Lifecycle Manager は、スナップショットの作成を開始する前に、事前スクリプト内のコマンドを実行して I/O をフリーズおよびフラッシュします。Amazon Data Lifecycle Manager はスナップショットの作成を開始した後に、事後スクリプト内のコマンドを実行して I/O を解凍します。

Amazon Data Lifecycle Manager を使用すると、以下のアプリケーション整合性のあるスナップショットを自動化できます。

- Volume Shadow Copy Service (VSS) を使用した Windows アプリケーション
- AWS マネージド SSDM ドキュメントを使用する SAP HANA。詳細については、「SAP HANA 用 Amazon EBS スナップショット」を参照してください。
- SSM ドキュメントテンプレートを使用した MySQL、PostgreSQL、InterSystems IRIS などのセルフマネージドデータベース

トピック

- 事前スクリプトと事後スクリプトを使用するための要件
- アプリケーション整合性のあるスナップショットの使用開始
- Amazon Data Lifecycle Manager での VSS バックアップに関する考慮事項
- アプリケーション整合性のあるスナップショットの責任共有

事前スクリプトと事後スクリプトを使用するための要件

次の表は、Amazon Data Lifecycle Manager で事前スクリプトと事後スクリプトを使用する際の要件の概要を示しています。

	アプリケーションコンシステントなスナップショット		
要件	VSS バックアップ	カスタム SSM ドキュメント	その他のユースケース
SSM Agent installed and running on target instances	✓	✓	✓
VSS system requirements met on target instances	✓		
VSS enabled instance profile associated with target instances	✓		
VSS components installed on target instances	✓		
Prepare SSM document with pre and post script commands		✓	✓

	アプリケーションコンシステントなスナップショット		
Prepare Amazon Data Lifecycle Manager IAM role run pre and post scripts	✓	✓	✓
Create snapshot policy that targets instances and is configured for pre and post scripts	✓	✓	✓

アプリケーション整合性のあるスナップショットの使用開始

このセクションでは、Amazon Data Lifecycle Manager を使用してアプリケーション整合性のあるスナップショットを自動化するために必要な手順について説明します。

ステップ 1: ターゲットインスタンスを準備する

Amazon Data Lifecycle Manager を使用して、ターゲットインスタンスをアプリケーション整合性のあるスナップショット用に準備する必要があります。ユースケースに応じて、以下のいずれかを実行します。

Prepare for VSS Backups

ターゲットインスタンスを VSS バックアップ用に準備するには

1. SSM Agent がまだインストールされていない場合は、ターゲットインスタンスにインストールします。SSM Agent がターゲットインスタンスに既にインストールされている場合は、このステップをスキップしてください。

   詳細については、「Windows Server 用 EC2 インスタンスで SSM Agent を使用する」を参照してください。
2. SSM エージェントが実行されていることを確認します。詳細については、「SSM Agent ステータスの確認とエージェントの起動」を参照してください。

3. Systems Manager の Amazon EC2 インスタンスをセットアップします。詳細については、「AWS Systems Manager ユーザーガイド」の「Systems Manager を利用した EC2 インスタンスの管理」を参照してください。
4. VSS バックアップのシステム要件が満たされていることを確認します。
5. VSS 対応インスタンスプロファイルをターゲットインスタンスにアタッチします。
6. VSS コンポーネントをインストールします。

Prepare for SAP HANA backups

ターゲットインスタンスを SAP HANA バックアップ用に準備するには

1. ターゲットインスタンス上に SAP HANA 環境を準備します。
   a. SAP HANA と一緒にインスタンスをセットアップします。既存の SAP HANA 環境がない場合は、「AWSでの SAP HANA 環境設定」を参照してください。
   b. SystemDB に適切な管理者ユーザーとしてログインします。
   c. Amazon Data Lifecycle Manager で使用するデータベースバックアップユーザーを作成します。

```
CREATE USER username PASSWORD password NO FORCE_FIRST_PASSWORD_CHANGE;
```

   例えば、次のコマンドは、名前が dlm_user でパスワードが password のユーザーを作成します。

```
CREATE USER dlm_user PASSWORD password NO FORCE_FIRST_PASSWORD_CHANGE;
```

   d. 前のステップで作成したデータベースバックアップユーザーに BACKUP OPERATOR ロールを割り当てます。

```
GRANT BACKUP OPERATOR TO username
```

   例えば、次のコマンドは、dlm_user という名前のユーザーにロールを割り当てます。

```
GRANT BACKUP OPERATOR TO dlm_user
```

   e. オペレーティングシステムに管理者 (例: *sid*adm) としてログインします。

f. 接続情報を保存する hdbuserstore エントリを作成して、ユーザーが情報を入力しなくても SAP HANA SSM ドキュメントが SAP HANA に接続できるようにします。

```
hdbuserstore set DLM_HANADB_SNAPSHOT_USER
 localhost:3hana_instance_number13 username password
```

例えば、次のようになります。

```
hdbuserstore set DLM_HANADB_SNAPSHOT_USER localhost:30013 dlm_user password
```

g. 接続をテストします。

```
hdbsql -U DLM_HANADB_SNAPSHOT_USER "select * from dummy"
```

2. SSM Agent がまだインストールされていない場合は、ターゲットインスタンスにインストールします。SSM Agent がターゲットインスタンスに既にインストールされている場合は、このステップをスキップしてください。

   詳細については、「Linux 用 EC2 インスタンスに SSM Agent を手動でインストールおよびアンインストールする」を参照してください。

3. SSM エージェントが実行されていることを確認します。詳細については、「SSM Agent ステータスの確認とエージェントの起動」を参照してください。

4. Systems Manager の Amazon EC2 インスタンスをセットアップします。詳細については、「AWS Systems Manager ユーザーガイド」の「Systems Manager を利用した EC2 インスタンスの管理」を参照してください。

Prepare for custom SSM documents

ターゲットインスタンスのカスタム SSM ドキュメントを準備するには

1. SSM Agent がまだインストールされていない場合は、ターゲットインスタンスにインストールします。SSM Agent がターゲットインスタンスに既にインストールされている場合は、このステップをスキップしてください。

   - (Linux インスタンス) Linux 用 EC2 インスタンスに SSM Agent を手動でインストールおよびアンインストールする
   - (Windows インスタンス) Windows Server 用 EC2 インスタンスで SSM Agent を使用する

2. SSM エージェントが実行されていることを確認します。詳細については、「SSM Agent ステータスの確認とエージェントの起動」を参照してください。
3. Systems Manager の Amazon EC2 インスタンスをセットアップします。詳細については、「AWS Systems Manager ユーザーガイド」の「Systems Manager を利用した EC2 インスタンスの管理」を参照してください。

ステップ 2: SSM ドキュメントを準備する

> **Note**
>
> このステップはカスタム SSM ドキュメントの場合にのみ必要です。VSS バックアップや SAP HANA には必要ありません。VSS Backups と SAP HANA の場合、Amazon Data Lifecycle Manager は AWS マネージド SSM ドキュメントを使用します。

MySQL や PostgreSQL、または InterSystems IRIS などのセルフマネージドデータベースのアプリケーション整合性のあるスナップショットを自動化している場合は、スナップショットの作成を開始する前に I/O をフリーズおよびフラッシュするための事前スクリプトと、スナップショットの作成を開始した後に I/O を解凍するための事後スクリプトを含む SSM コマンドドキュメントを作成する必要があります。

MySQL や PostgreSQL、または InterSystems IRIS データベースが標準設定を使用している場合は、以下のサンプル SSM ドキュメントコンテンツを使用して SSM コマンドドキュメントを作成できます。MySQL や PostgreSQL、または InterSystems IRIS データベースが非標準設定を使用している場合は、以下のサンプルコンテンツを SSM コマンドドキュメントの開始点として使用し、要件に合わせてカスタマイズできます。あるいは、新しい SSM ドキュメントを最初から作成する場合は、以下の空の SSM ドキュメントテンプレートを使用して、該当するドキュメントセクションに事前コマンドと事後コマンドを追加できます。

> **⚠ 次の点に注意してください。**
>
> - SSM ドキュメントがデータベース設定に対して正しく必要なアクションを実行することを確認するのはお客様の責任です。
> - SSM ドキュメント内の事前スクリプトと事後スクリプトが I/O を正常にフリーズ、フラッシュ、および解凍できた場合にのみ、スナップショットのアプリケーション整合性が保証されます。

- SSM ドキュメントには、pre-script、post-script、dry-run などの allowedValues の必須フィールドが含まれている必要があります。Amazon Data Lifecycle Manager は、これらのセクションのコンテンツに基づいてインスタンス上でコマンドを実行します。SSM ドキュメントにこれらのセクションがない場合、Amazon Data Lifecycle Manager は、そのドキュメントを実行に失敗したものとして扱います。

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
```

詳細については、GitHub リポジトリを参照してください。

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."
```

SSM ドキュメントコンテンツを取得したら、以下のいずれかの手順でカスタム SSM ドキュメントを作成します。

Console

SSM コマンドドキュメント を作成するには

1. https://console.aws.amazon.com//systems-manager/ で AWS Systems Manager コンソールを開きます。
2. ナビゲーションペインで [ドキュメント] を選択し、[ドキュメントの作成]、[コマンドまたはセッション] の順に選択します。
3. [名前] に、ドキュメントのわかりやすい名前を入力します。
4. [ターゲットタイプ] に、[/AWS::EC2::Instance] を選択します。
5. [ドキュメントタイプ] に、[コマンド] を選択します。
6. [コンテンツ] フィールドで [YAML] を選択し、ドキュメントコンテンツを貼り付けます。
7. [ドキュメントタグ] セクションで、タグキーが `DLMScriptsAccess` でタグ値が `true` のタグを追加します。

> **Important**
>
> `DLMScriptsAccess:true` タグは、ステップ 3: Amazon Data Lifecycle Manager IAM ロールを準備するで使用される AWSDataLifecycleManagerSSMFullAccess AWS 管理ポリシーで必要です。このポリシーでは `aws:ResourceTag` 条件キーが使用されており、このタグを持つ SSM ドキュメントへのアクセスを制限します。

8. [ドキュメントの作成] を選択します。

AWS CLI

SSM コマンドドキュメント を作成するには

create-document コマンドを使用します。`--name` には、ドキュメントのわかりやすい名前を指定します。`--document-type` の場合、`Command` を指定します。`--content` には、SSM ドキュメントコンテンツを含む .yaml ファイルへのパスを指定します。`--tags` の場合、`"Key=DLMScriptsAccess,Value=true"` を指定します。

```
$ 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"
```

ステップ 3: Amazon Data Lifecycle Manager の IAM ロールを準備する

> **Note**
>
> このステップは次の場合に必要です。
>
> - カスタム IAM ロールを使用する、事前/事後スクリプト対応のスナップショットポリシーを作成または更新します。
> - コマンドラインを使用して、デフォルトを使用する、事前/事後スクリプト対応のスナップショットポリシーを作成または更新します。
>
> コンソールを使用して、スナップショットを管理するためのデフォルトロール (AWSDataLifecycleManagerDefaultRole) を使用する、事前/事後スクリプト対応のスナップショットポリシーを作成または更新する場合は、このステップをスキップしてください。この場合、AWSDataLifecycleManagerSSMFullAccess ポリシーはそのロールに自動的にアタッチされます。

ポリシーに使用する IAM ロールが、ポリシーの対象となるインスタンスで事前スクリプトと事後スクリプトを実行するために必要な SSM アクションを実行するアクセス許可を Amazon Data Lifecycle Manager に付与していることを確認する必要があります。

Amazon Data Lifecycle Manager には、必要なアクセス許可を含むマネージドポリシー (AWSDataLifecycleManagerSSMFullAccess) が用意されています。このポリシーを IAM ロールにアタッチしてスナップショットを管理し、アクセス許可が含まれていることを確認できます。

> **Important**
>
> AWSDataLifecycleManagerSSMFullAccess マネージドポリシーでは、事前スクリプトと事後スクリプトを使用するときに、`aws:ResourceTag` 条件キーを使って特定の SSM ドキュメントへのアクセスを制限します。Amazon Data Lifecycle Manager が SSM ドキュメントにアクセスできるようにするには、SSM ドキュメントに がタグ付けされていることを確認する必要がありますDLMScriptsAccess:true。

あるいは、カスタムポリシーを手動で作成するか、使用する IAM ロールに必要なアクセス許可を直接割り当てることもできます。AWSDataLifecycleManagerSSMFullAccess マネージドポリシーで定義されているのと同じアクセス許可を使用できますが、`aws:ResourceTag` 条件キーはオプションです。その条件キーを含めない場合は、SSM ドキュメントに `DLMScriptsAccess:true` のタグを付ける必要はありません。

以下のいずれかの方法を使用して、AWSDataLifecycleManagerSSMFullAccess ポリシーを IAM ロールに追加します。

Console

マネージドポリシーをカスタムロールにアタッチするには

1. https://console.aws.amazon.com/iam/ で IAM コンソールを開きます。
2. ナビゲーションパネルで [Roles (ロール)] を選択します。
3. スナップショットを管理するためのカスタムロールを検索し、選択します。
4. [アクセス許可] タブで、[アクセス許可の追加]、[ポリシーをアタッチ] の順に選択します。
5. [AWSDataLifecycleManagerSSMFullAccess] マネージドポリシーを検索して選択し、[アクセス許可の追加] を選択します。

AWS CLI

マネージドポリシーをカスタムロールにアタッチするには

attach-role-policy コマンドを使用します。`---role-name` には、カスタムロールの名前を指定します。`--policy-arn` の場合、`arn:aws:iam::aws:policy/AWSDataLifecycleManagerSSMFullAccess` を指定します。

```
$ aws iam attach-role-policy \
--policy-arn arn:aws:iam::aws:policy/AWSDataLifecycleManagerSSMFullAccess \
--role-name your_role_name
```

ステップ 4: スナップショットライフサイクルポリシーを作成する

アプリケーション整合性のあるスナップショットを自動化するには、インスタンスをターゲットとするスナップショットライフサイクルポリシーを作成し、そのポリシーの事前スクリプトと事後スクリプトを設定する必要があります。

Console

スナップショットライフサイクルポリシーを作成するには

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. ナビゲーションペインで、[Elastic Block Store]、[ライフサイクルマネージャー]、[ライフサイクルポリシーの作成] の順に選択します。
3. リポジトリの [ポリシータイプの選択] 画面で、[EBS スナップショットポリシー] を選択し、[次へ] をクリックします。
4. [Target resources] (ターゲットリソース) セクションで、以下の操作を行います。
   a. [ターゲットリソースタイプ] で、[Instance] を選択します。
   b. [ターゲットリソースタグ] で、バックアップするインスタンスを識別するリソースタグを指定します。指定したタグを持つリソースのみがバックアップされます。
5. [IAM ロール] には、AWSDataLifecycleManagerDefaultRole (スナップショットを管理するためのデフォルトロール) を選択するか、事前スクリプトおよび事後スクリプト用に作成して準備したカスタムロールを選択します。
6. 必要に応じて、スケジュールと追加のオプションを設定します。メンテナンスウィンドウ中など、ワークロードに合った期間にスナップショット作成時刻をスケジュールすることをお勧めします。

   SAP HANA では、高速スナップショット復元を有効にすることをお勧めします。

   > **Note**
   >
   > VSS バックアップのスケジュールを有効にすると、[特定のデータボリュームを除外] または [ソースからタグをコピー] を有効にすることはできません。

7. [事前スクリプトと事後スクリプト] セクションで [事前スクリプトと事後スクリプトを有効にする] を選択し、ワークロードに応じて次の操作を行います。

   - Windows アプリケーションのアプリケーション整合性のあるスナップショットを作成するには、[VSS バックアップ] を選択します。
   - SAP HANA ワークロードのアプリケーション整合性のあるスナップショットを作成するには、[SAP HANA] を選択します。
   - カスタム SSM ドキュメントを使用して、セルフマネージドの MySQL、PostgreSQL、または InterSystems IRIS データベースを含む、他のすべてのデータベースとワークロード

のアプリケーション整合性のあるスナップショットを作成するには、[カスタム SSM ドキュメント] を選択します。

1. [自動化オプション] に、[事前スクリプトと事後スクリプト] を選択します。
2. [SSM ドキュメント] に、準備した SSM ドキュメントを選択します。

8. 選択したオプションに応じて、以下の追加オプションを設定します。

- [スクリプトタイムアウト] — (カスタム SSM ドキュメントのみ) Amazon Data Lifecycle Manager がスクリプトの実行試行を完了していない場合に、その試行が失敗するまでのタイムアウト期間。スクリプトがタイムアウト期間内に完了しない場合、Amazon Data Lifecycle Manager はその試行に失敗します。タイムアウト期間は、事前スクリプトと事後スクリプトに個別に適用されます。デフォルトのタイムアウト期間は 10 秒です。また、最大タイムアウト期間は 120 秒です。
- [失敗したスクリプトの再試行] — タイムアウト期間内に完了しなかったスクリプトを再試行する場合は、このオプションを選択します。事前スクリプトが失敗した場合、Amazon Data Lifecycle Manager は、事前スクリプトと事後スクリプトの実行を含め、スナップショット作成プロセス全体を再試行します。事後スクリプトが失敗した場合、Amazon Data Lifecycle Manager は、事後スクリプトのみを再試行します。この場合、事前スクリプトは完了し、スナップショットが作成された可能性があります。
- Crash-consistent スナップショットをデフォルトで作成 – このオプションを選択すると、事前スクリプトの実行に失敗した場合に、Crash-consistent スナップショットがデフォルトで作成されます。これは、事前スクリプトと事後スクリプトが有効になっていない場合の、Amazon Data Lifecycle Manager のデフォルトのスナップショット作成動作です。再試行を有効にした場合、Amazon Data Lifecycle Manager は、再試行回数をすべて使い切った後にのみ、Crash-consistent スナップショットをデフォルトで作成します。事前スクリプトが失敗し、Crash-consistent スナップショットがデフォルトで作成されなかった場合、Amazon Data Lifecycle Manager は、そのスケジュールの実行中にインスタンスのスナップショットを作成しません。

> **Note**
>
> SAP HANA のスナップショットを作成する場合は、このオプションを無効にすることをお勧めします。SAP HANA ワークロードのCrash-consistent スナップショットは、同じ方法では復元できません。

9. [デフォルトポリシーの作成] を選択します。

> **Note**
>
> Role with name AWSDataLifecycleManagerDefaultRole already exists エラーが発生した場合、詳細については「Amazon Data Lifecycle Manager の問題のトラブルシューティング」を参照してください。

AWS CLI

スナップショットライフサイクルポリシーを作成するには

create-lifecycle-policy コマンドを使用して、CreateRule に Scripts パラメータを含めます。パラメータの詳細については、「Amazon Data Lifecycle Manager API Reference」を参照してください。

```
$ aws dlm create-lifecycle-policy \
--description "policy_description" \
--state ENABLED \
--execution-role-arn iam_role_arn \
--policy-details file://policyDetails.json
```

ユースケースに応じて、ここで policyDetails.json には以下のいずれかが含まれます。

- VSS バックアップ

```
{
    "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
        }
    }]
}
```

- SAP HANA バックアップ

```
{
    "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
        }
    }]
}
```

- カスタム SSM ドキュメント

```
{
```

```
    "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
        }
    }]
}
```

Amazon Data Lifecycle Manager での VSS バックアップに関する考慮事項

Amazon Data Lifecycle Manager では、Amazon EC2 インスタンスで実行されている VSS (Volume Shadow Copy Service) 対応の Windows アプリケーションをバックアップおよび復元できます。アプリケーションに Windows VSS に登録された VSS ライターがある場合、Amazon Data Lifecycle Manager は、そのアプリケーションに対してアプリケーションと整合性のあるスナップショットを作成します。

> **Note**
>
> Amazon Data Lifecycle Manager は現在、Amazon EC2 で実行されているリソースのアプリケーション整合性のあるスナップショットのみをサポートしています。特に、既存のインスタンスをバックアップから作成された新しいインスタンスに置き換えることでアプリケーションデータを復元できるバックアップシナリオを対象としています。VSS バックアップで

は、すべてのインスタンスタイプまたはアプリケーションがサポートされているわけではありません。詳細については、「Amazon EC2 ユーザーガイド」の「アプリケーションと整合性のある、Windows VSS ベースの Amazon EBS スナップショット」を参照してください。

サポートされていない インスタンスタイプ

以下の Amazon EC2 インスタンスタイプは、VSS バックアップではサポートされていません。ポリシーがこれらのインスタンスタイプのいずれかをターゲットにしている場合、Amazon Data Lifecycle Manager は引き続き VSS バックアップを作成できますが、スナップショットには必要なシステムタグがタグ付けされていない可能性があります。これらのタグがないと、スナップショットは作成後に Amazon Data Lifecycle Manager によって管理されません。これらのスナップショットは手動で削除する必要がある場合があります。

- T3: `t3.nano` | `t3.micro`
- T3a: `t3a.nano` | `t3a.micro`
- T2: `t2.nano` | `t2.micro`

アプリケーション整合性のあるスナップショットの責任共有

以下を確認する必要があります。

- SSM Agent がインストールされ、最新の状態で、ターゲットインスタンスで実行されている
- Systems Manager には、ターゲットインスタンスで必要なアクションを実行する権限があります。
- Amazon Data Lifecycle Manager には、ターゲットインスタンスでの事前スクリプトと事後スクリプトの実行に必要な Systems Manager アクションを実行する権限があります。
- カスタムワークロード (MySQL や PostgreSQL、または InterSystems IRIS などのセルフマネージド型データベース) の場合、使用する SSM ドキュメントには、データベース設定の I/O をフリーズ、フラッシュ、および解凍するための適切で必要なアクションが含まれています。
- スナップショット作成時刻はワークロードのスケジュールに合わせて調整されます。例えば、スケジュールされたメンテナンスウィンドウ中にスナップショットの作成をスケジュールしようとします。

Amazon Data Lifecycle Manager では以下の動作が保証されます。

- スナップショットの作成は、スケジュールされたスナップショット作成時刻から 60 分以内に開始されます。
- 事前スクリプトは、スナップショットの作成が開始される前に実行されます。
- 事後スクリプトは、事前スクリプトが成功し、スナップショットの作成が開始された後に実行されます。Amazon Data Lifecycle Manager は、事前スクリプトが成功した場合にのみ事後スクリプトを実行します。事前スクリプトが失敗した場合、Amazon Data Lifecycle Manager は事後スクリプトを実行しません。
- スナップショットは作成時に適切なタグでタグ付けされます。
- CloudWatch メトリクスおよびイベントは、スクリプトが開始されたとき、およびスクリプトが失敗または成功したときに発生します。

Data Lifecycle Manager の事前スクリプトと事後スクリプトのその他のユースケース

事前スクリプトと事後スクリプトを使用してアプリケーション整合性のあるスナップショットを自動化するだけでなく、事前スクリプトと事後スクリプトを一緒に、または個別に使用して、スナップショット作成前または作成後の他の管理タスクを自動化できます。例えば、次のようになります。

- スナップショットを作成する前に、事前スクリプトを使用してパッチを適用します。これにより、毎週または毎月の定期的なソフトウェアアップデートを適用した後にスナップショットを作成できます。

  > **Note**
  >
  > 事前スクリプトのみを実行することを選択した場合は、[Crash-consistent スナップショットをデフォルトで作成] がデフォルトで有効になります。

- スナップショットを作成した後に、事後スクリプトを使用してパッチを適用します。これにより、毎週または毎月の定期的なソフトウェアアップデートを適用する前にスナップショットを作成できます。

他のユースケースで使用を開始する

このセクションでは、アプリケーション整合性のあるスナップショット以外のユースケースで事前スクリプトまたは事後スクリプトを使用する場合に実行する必要がある手順について説明します。

ステップ 1: ターゲットインスタンスを準備する

ターゲットインスタンスを事前スクリプトまたは事後スクリプト用に準備するには

1. SSM Agent がまだインストールされていない場合は、ターゲットインスタンスにインストールします。SSM Agent がターゲットインスタンスに既にインストールされている場合は、このステップをスキップしてください。
   - (Linux インスタンス) Linux 用 EC2 インスタンスに SSM Agent を手動でインストールおよびアンインストールする
   - (Windows インスタンス) Windows Server 用 EC2 インスタンスで SSM Agent を使用する
2. SSM エージェントが実行されていることを確認します。詳細については、「SSM Agent ステータスの確認とエージェントの起動」を参照してください。
3. Systems Manager の Amazon EC2 インスタンスをセットアップします。詳細については、「AWS Systems Manager ユーザーガイド」の「Systems Manager を利用した EC2 インスタンスの管理」を参照してください。

ステップ 2: SSM ドキュメントを準備する

実行するコマンドを使った事前スクリプトまたは事後スクリプトを含む SSM コマンドドキュメントを作成する必要があります。

以下の空の SSM ドキュメントテンプレートを使用して SSM ドキュメントを作成し、事前スクリプトコマンドと事後スクリプトコマンドを該当するドキュメントセクションに追加できます。

⚠ 次の点に注意してください。

- SSM ドキュメントがワークロードに対して正しく必要なアクションを実行することを確認するのはお客様の責任です。
- SSM ドキュメントには、`pre-script`、`post-script`、`dry-run` などの `allowedValues` の必須フィールドが含まれている必要があります。Amazon Data Lifecycle Manager は、これらのセクションのコンテンツに基づいてインスタンス上でコマ

ンドを実行します。SSM ドキュメントにこれらのセクションがない場合、Amazon Data Lifecycle Manager は、そのドキュメントを実行に失敗したものとして扱います。

```
###======================================================================================###
# 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."
```

ステップ 3: Amazon Data Lifecycle Manager の IAM ロールを準備する

> **Note**
>
> このステップは次の場合に必要です。

- カスタム IAM ロールを使用する、事前/事後スクリプト対応のスナップショットポリシーを作成または更新します。
- コマンドラインを使用して、デフォルトを使用する、事前/事後スクリプト対応のスナップショットポリシーを作成または更新します。

コンソールを使用して、スナップショットを管理するためのデフォルトロール (AWSDataLifecycleManagerDefaultRole) を使用する、事前/事後スクリプト対応のスナップショットポリシーを作成または更新する場合は、このステップをスキップしてください。この場合、AWSDataLifecycleManagerSSMFullAccess ポリシーはそのロールに自動的にアタッチされます。

ポリシーに使用する IAM ロールが、ポリシーの対象となるインスタンスで事前スクリプトと事後スクリプトを実行するために必要な SSM アクションを実行するアクセス許可を Amazon Data Lifecycle Manager に付与していることを確認する必要があります。

Amazon Data Lifecycle Manager には、必要なアクセス許可を含むマネージドポリシー (AWSDataLifecycleManagerSSMFullAccess) が用意されています。このポリシーを IAM ロールにアタッチしてスナップショットを管理し、アクセス許可が含まれていることを確認できます。

Important

AWSDataLifecycleManagerSSMFullAccess マネージドポリシーでは、事前スクリプトと事後スクリプトを使用するときに、`aws:ResourceTag` 条件キーを使って特定の SSM ドキュメントへのアクセスを制限します。Amazon Data Lifecycle Manager が SSM ドキュメントにアクセスできるようにするには、SSM ドキュメントに がタグ付けされていることを確認する必要がありますDLMScriptsAccess:true。

あるいは、カスタムポリシーを手動で作成するか、使用する IAM ロールに必要なアクセス許可を直接割り当てることもできます。AWSDataLifecycleManagerSSMFullAccess マネージドポリシーで定義されているのと同じアクセス許可を使用できますが、`aws:ResourceTag` 条件キーはオプションです。この条件キーを使用しない場合は、SSM ドキュメントに `DLMScriptsAccess:true` のタグを付ける必要はありません。

以下のいずれかの方法を使用して、AWSDataLifecycleManagerSSMFullAccess ポリシーを IAM ロールに追加します。

Console

マネージドポリシーをカスタムロールにアタッチするには

1. https://console.aws.amazon.com/iam/ で IAM コンソールを開きます。
2. ナビゲーションパネルで [Roles (ロール)] を選択します。
3. スナップショットを管理するためのカスタムロールを検索し、選択します。
4. [アクセス許可] タブで、[アクセス許可の追加]、[ポリシーをアタッチ] の順に選択します。
5. [AWSDataLifecycleManagerSSMFullAccess] マネージドポリシーを検索して選択し、[アクセス許可の追加] を選択します。

AWS CLI

マネージドポリシーをカスタムロールにアタッチするには

attach-role-policy コマンドを使用します。`---role-name` には、カスタムロールの名前を指定します。`--policy-arn` の場合、`arn:aws:iam::aws:policy/AWSDataLifecycleManagerSSMFullAccess` を指定します。

```
$ aws iam attach-role-policy \
--policy-arn arn:aws:iam::aws:policy/AWSDataLifecycleManagerSSMFullAccess \
--role-name your_role_name
```

スナップショットライフサイクルポリシーを作成する

Console

スナップショットライフサイクルポリシーを作成するには

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. ナビゲーションペインで、[Elastic Block Store]、[ライフサイクルマネージャー]、[ライフサイクルポリシーの作成] の順に選択します。
3. リポジトリの [ポリシータイプの選択] 画面で、[EBS スナップショットポリシー] を選択し、[次へ] をクリックします。
4. [Target resources] (ターゲットリソース) セクションで、以下の操作を行います。

   a. [ターゲットリソースタイプ] で、[Instance] を選択します。

b. [ターゲットリソースタグ] で、バックアップするインスタンスを識別するリソースタグを指定します。指定したタグを持つリソースのみがバックアップされます。

5. [IAM ロール] には、AWSDataLifecycleManagerDefaultRole (スナップショットを管理するためのデフォルトロール) を選択するか、事前スクリプトおよび事後スクリプト用に作成して準備したカスタムロールを選択します。
6. 必要に応じて、スケジュールと追加のオプションを設定します。メンテナンスウィンドウ中など、ワークロードに合った期間にスナップショット作成時刻をスケジュールすることをお勧めします。
7. [事前スクリプトと事後スクリプト] セクションで、[事前スクリプトと事後スクリプトを有効にする] を選択し、次の操作を行います。

   a. [カスタム SSM ドキュメント] を選択します。

   b. [自動化] オプションで、実行するスクリプトに一致するオプションを選択します。

   c. [SSM ドキュメント] に、準備した SSM ドキュメントを選択します。

8. 必要に応じて、次の追加オプションを設定します。

   - [スクリプトタイムアウト] — Amazon Data Lifecycle Manager がスクリプトの実行試行を完了していない場合に、その試行が失敗するまでのタイムアウト期間。スクリプトがタイムアウト期間内に完了しない場合、Amazon Data Lifecycle Manager はその試行に失敗します。タイムアウト期間は、事前スクリプトと事後スクリプトに個別に適用されます。デフォルトのタイムアウト期間は 10 秒です。また、最大タイムアウト期間は 120 秒です。
   - [失敗したスクリプトの再試行] — タイムアウト期間内に完了しなかったスクリプトを再試行する場合は、このオプションを選択します。事前スクリプトが失敗した場合、Amazon Data Lifecycle Manager は、事前スクリプトと事後スクリプトの実行を含め、スナップショット作成プロセス全体を再試行します。事後スクリプトが失敗した場合、Amazon Data Lifecycle Manager は、事後スクリプトのみを再試行します。この場合、事前スクリプトは完了し、スナップショットが作成された可能性があります。
   - Crash-consistent スナップショットをデフォルトで作成 – このオプションを選択すると、事前スクリプトの実行に失敗した場合に、Crash-consistent スナップショットがデフォルトで作成されます。これは、事前スクリプトと事後スクリプトが有効になっていない場合の、Amazon Data Lifecycle Manager のデフォルトのスナップショット作成動作です。再試行を有効にした場合、Amazon Data Lifecycle Manager は、再試行回数をすべて使い切った後にのみ、Crash-consistent スナップショットをデフォルトで作成します。事前スクリプトが失敗し、Crash-consistent スナップショットがデフォルトで作成されなかった

場合、Amazon Data Lifecycle Manager は、そのスケジュールの実行中にインスタンスのスナップショットを作成しません。

9. [デフォルトポリシーの作成] を選択します。

> **Note**
>
> `Role with name AWSDataLifecycleManagerDefaultRole already exists` エラーが発生した場合、詳細については「Amazon Data Lifecycle Manager の問題のトラブルシューティング」を参照してください。

AWS CLI

スナップショットライフサイクルポリシーを作成するには

create-lifecycle-policy コマンドを使用して、`CreateRule` に `Scripts` パラメータを含めます。パラメータの詳細については、「Amazon Data Lifecycle Manager API Reference」を参照してください。

```
$ aws dlm create-lifecycle-policy \
--description "policy_description" \
--state ENABLED \
--execution-role-arn iam_role_arn \
--policy-details file://policyDetails.json
```

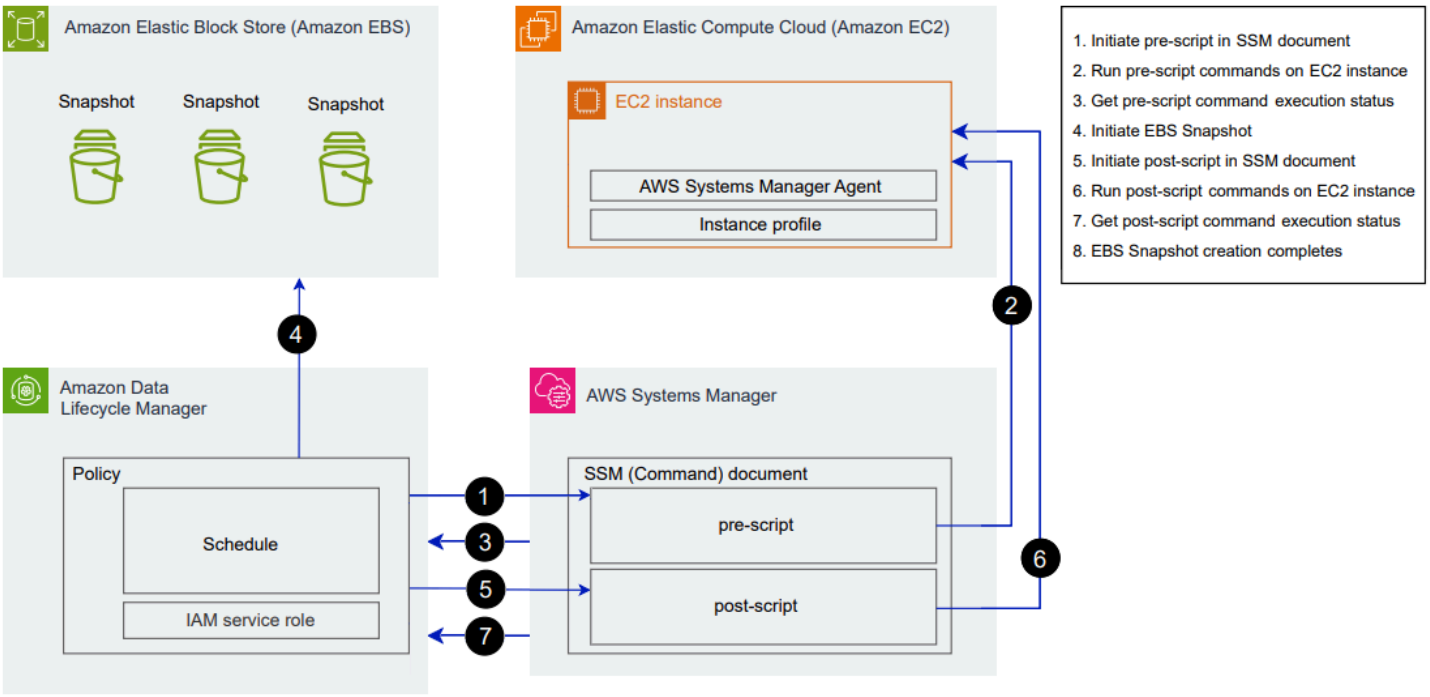
ここで `policyDetails.json` には以下が含まれます。

```
{
    "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
        }
    }]
}
```

Amazon Data Lifecycle Manager の事前スクリプトと事後スクリプトの仕組み

次の図は、カスタム SSM ドキュメントを使用する場合の事前スクリプトと事後スクリプトのプロセスフローを示しています。これは、VSS バックアップには適用されません。



スケジュールされたスナップショット作成時刻に、以下のアクションおよびクロスサービスインタラクションが発生します。

1. Amazon Data Lifecycle Manager は、SSM ドキュメントを呼び出して `pre-script` パラメータを渡すことにより、事前スクリプトアクションを開始します。

   > **Note**
   >
   > ステップ 1～3 は、事前スクリプトを実行する場合にのみ発生します。事後スクリプトのみを実行する場合、ステップ 1～3 はスキップされます。

2. Systems Manager は、ターゲットインスタンスで実行されている SSM Agent に事前スクリプトコマンドを送信します。SSM Agent はインスタンス上でコマンドを実行し、ステータス情報を Systems Manager に送り返します。

   たとえば、SSM ドキュメントを使用してアプリケーション整合性のあるスナップショットを作成する場合、スナップショットが作成される前にバッファされたすべてのデータがボリュームに書き込まれるように、事前スクリプトがフリーズして I/O をフラッシュすることがあります。

3. Systems Manager は、事前スクリプトコマンドのステータス更新を Amazon Data Lifecycle Manager に送信します。事前スクリプトが失敗した場合、Amazon Data Lifecycle Manager は、事前スクリプトオプションと事後スクリプトオプションの設定方法に応じて、以下のいずれかのアクションを実行します。

再試行	Crash-consistent スナップショットをデフォルトで作成	アクション
残り再試行回数ありで有効	有効	成功するか、再試行が終了するまでスクリプトを再試行してください。
正常に完了することなく終了	有効	Crash-consistent スナップショットが作成されるため、事後スクリプトは実行しません。
残り再試行回数ありで有効	Disabled	成功するか、再試行が終了するまでスクリプトを再試行してください。
正常に完了することなく終了	Disabled	ターゲットインスタンスのスナップショット作成がスキップされる

再試行	Crash-consistent スナップショットをデフォルトで作成	アクション
		ため、事後スクリプトは実行しません
Disabled	有効	Crash-consistent スナップショットが作成されるため、事後スクリプトは実行しません。
Disabled	Disabled	ターゲットインスタンスのスナップショット作成がスキップされるため、事後スクリプトは実行しません

4. Amazon Data Lifecycle Manager がスナップショットの作成を開始します。
5. Amazon Data Lifecycle Manager は、SSM ドキュメントを呼び出して `post-script` パラメータを渡すことにより、事後スクリプトアクションを開始します。

> **Note**
>
> ステップ 5 ~ 7 は、事前スクリプトを実行する場合にのみ発生します。事後スクリプトのみを実行する場合、ステップ 1 ~ 3 はスキップされます。

6. Systems Manager は、ターゲットインスタンスで実行されている SSM Agent に事後スクリプトコマンドを送信します。SSM Agent はインスタンス上でコマンドを実行し、ステータス情報を Systems Manager に送り返します。

   たとえば、SSM ドキュメントがアプリケーション整合性のあるスナップショットを有効にしている場合、このポストスクリプトは I/O を解凍して、スナップショットの作成後にデータベースが通常の I/O オペレーションを再開するようにします。
7. 事後スクリプトを実行し、Systems Manager で正常に完了したことが表示されれば、処理は完了します。

   事後スクリプトが失敗した場合、Amazon Data Lifecycle Manager は、事前スクリプトオプションと事後スクリプトオプションの設定方法に応じて、以下のいずれかのアクションを実行します。

再試行	アクション
残り再試行回数ありで有効	成功するか、再試行回数を使い切るまで事後スクリプトを再試行します
成功することなく終了	ポストスクリプトをスキップする
Disabled	ポストスクリプトをスキップする

事後スクリプトが失敗しても、事前スクリプト (有効な場合) は正常に完了し、スナップショットが作成された可能性があることに注意してください。インスタンスが期待どおりに動作していることを確認するために、インスタンスに対してさらにアクションを実行する必要がある場合があります。例えば、事前スクリプトが I/O を一時停止してフラッシュしたが、事後スクリプトが I/O の解凍に失敗した場合は、I/O を自動解凍するようにデータベースを設定するか、I/O を手動で解凍する必要があります。

8. スナップショットの作成プロセスは、事後スクリプトの完了後に完了することがあります。スナップショットの完了にかかる時間は、スナップショットのサイズによって異なります。

Data Lifecycle Manager の事前スクリプトと事後スクリプトで作成されたスナップショットを特定

Amazon Data Lifecycle Manager は、事前スクリプトと事後スクリプトで作成されたスナップショットに、以下のシステムタグを自動的に割り当てます。

- キー: `aws:dlm:pre-script`、値: SUCCESS|FAILED

  タグ値 SUCCESS は、事前スクリプトが正常に実行されたことを示します。タグ値 FAILED は、事前スクリプトが正常に実行されなかったことを示します。

- キー: `aws:dlm:post-script`、値: SUCCESS|FAILED

  タグ値 SUCCESS は、事後スクリプトが正常に実行されたことを示します。タグ値 FAILED は、事後スクリプトが正常に実行されなかったことを示します。

カスタム SSM ドキュメントと SAP HANA バックアップでは、スナップショットに aws:dlm:pre-script:SUCCESS と aws:dlm:post-script:SUCCESS の両方のタグが付けられている場合、アプリケーション整合性のあるスナップショットが正常に作成されたと推測できます。

さらに、VSS バックアップを使用して作成されたアプリケーション整合性のあるスナップショットには、自動的に次のタグが付けられます。

- キー: AppConsistent tag、値: true|false

  タグ値 true は、VSS バックアップが成功し、スナップショットとアプリケーションの整合性が保たれていることを示します。タグ値 false は、VSS バックアップが成功せず、スナップショットとアプリケーションの整合性が保たれていないことを示します。

Amazon Data Lifecycle Manager の事前スクリプトと事後スクリプトをモニタリング

Amazon CloudWatch メトリクス

Amazon Data Lifecycle Manager は、事前スクリプトと事後スクリプトが失敗した場合と成功した場合、および VSS バックアップが失敗した場合と成功した場合に、次の CloudWatch メトリクスを発行します。

- PreScriptStarted
- PreScriptCompleted
- PreScriptFailed
- PostScriptStarted
- PostScriptCompleted
- PostScriptFailed
- VSSBackupStarted
- VSSBackupCompleted
- VSSBackupFailed

詳細については、「CloudWatch を使用して、Data Lifecycle Manager のポリシーをモニタリング」を参照してください。

Amazon EventBridge

Amazon Data Lifecycle Manager は、事前スクリプトまたは事後スクリプトが開始、成功、または失敗したときに、次の Amazon EventBridge イベントを発行します。

- `DLM Pre Post Script Notification`

詳細については、「EventBridge を使用した Data Lifecycle Manager ポリシーのモニタリング」を参照してください。

EBS-backed AMI 用の Amazon Data Lifecycle Manager カスタムポリシーを作成

以下の手順では、Amazon Data Lifecycle Manager を使用して EBS-backed AMI のライフサイクルを自動化する方法を示します。

トピック

- AMI ライフサイクルポリシーを作成する
- AMI ライフサイクルポリシーに関する考慮事項
- その他のリソース

AMI ライフサイクルポリシーを作成する

AMI ライフサイクルポリシーを作成するには、次のいずれかの手順に従います。

Console

AMI ポリシーを作成するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインで、[Elastic Block Store]、[ライフサイクルマネージャー]、[ライフサイクルポリシーの作成] の順に選択します。
3. [ポリシータイプの選択] 画面で、[EBS-backed AMI ポリシー] を選択した後、[次] をクリックします。
4. [Target resources] (ターゲットのリソース) セクションの [Target resource tags] (ターゲットリソースタグ) で、バックアップするボリュームまたはインスタンスを識別するリソースタグを選択します。ポリシーでは、特定のタグキーと値のペアを持つリソースのみがバックアップされます。

5. [説明] にポリシーの簡単な説明を入力します。
6. [IAM ロール] で、AMI とスナップショットを管理しインスタンスを記述するためのアクセス許可を持つ、IAM ロールを選択します。Amazon Data Lifecycle Manager から提供されるデフォルトのロールを使用するには、[デフォルトロール] を選択します。以前に作成したカスタム IAM ロールを使用するには、[別のロールを選択] をクリックした上で、使用するロールを選択します。
7. [ポリシータグ] に、ライフサイクルポリシーに適用されるタグを追加します。これらのタグは、ポリシーを識別および分類するために使用することができます。
8. [作成後のポリシーの状態] で、[ポリシーの有効化] を選択すると、次にスケジュールした時刻でポリシーが実行されます。ポリシーが実行されないようにするには、[ポリシーの無効化] を選択します。ここでポリシーを有効にしない場合、作成後に手動で有効にするまで AMI の作成は開始されません。
9. [インスタンスの再起動] セクションに、AMI の作成前にインスタンスを再起動するかどうかが表示されています。ターゲットのインスタンスが再起動されないようにするには、[いいえ] を選択します。[いいえ] を選択することで、データの整合性に問題が発生する場合があります。AMI の作成前にインスタンスを再起動するには、[はい] を選択します。これを選択すると、データの整合性が保証されます。ただし、複数のターゲットインスタンスが同時に再起動する可能性があります。
10. [次へ] を選択します。
11. [スケジュールの設定] 画面で、ポリシースケジュールを設定します。1 つのポリシーには、最大 4 つのスケジュールを含めることができます。スケジュール 1 は必須です。スケジュール 2、3、および 4 はオプションです。追加したポリシースケジュールごとに、以下の操作を行います。

    a. [スケジュールの詳細] セクションで、次の操作を行います。

        i. [スケジュール名] で、スケジュールの分かりやすい名前を指定します。
        ii. [頻度]とそれに関連するフィールドで、ポリシーの実行間隔を設定します。

            ポリシーの実行は、日次、週次、月次、年次のいずれかのスケジュールで設定できます。または、[カスタム cron 式] をクリックし、最長 1 年の間隔を指定します。詳細については、「Amazon EventBridge ユーザーガイド」の「Setting a schedule pattern for scheduled rules (legacy) in Amazon EventBridge」を参照してください。
        iii. [開始時刻] では、ポリシー実行の開始予定時刻を指定します。初回のポリシー実行は、スケジュールした時刻から 1 時間以内に開始されます。時刻は、hh:mm UTC 形式で入力する必要があります。

iv. [保持タイプ] で、スケジュールで作成された AMI の保持ポリシーを指定します。

AMI は、総数または期間に基づいて保持できます。

カウントベースの保持の場合、指定できる範囲は 1 ~ 1000 です。このカウントが最大数に達すると、新しい AMI が作成される時点で、最も古いスナップショットの登録が解除されます。

保存期間に基づく保持の場合、指定できる範囲は 1 日 ~ 100 年です。各 AMI の保存期間が終了すると、その AMI は削除されます。

> **Note**
>
> すべてのスケジュールは、同じ保持タイプである必要があります。保持タイプを指定できるのは、スケジュール 1 のみです。スケジュール 2、3、4 は、スケジュール 1 から保持タイプを継承します。各スケジュールには、独自の保持回数または期間を設定できます。

b. AMI のタグ付けを設定します。

[タグ付け] セクションで、以下を実行します。

i. ソースインスタンスからスケジュールにより作成された AMI に対し、すべてのユーザー定義タグをコピーするには、[ソースからタグをコピー] を選択します。

ii. デフォルトで、スケジュールによって作成された AMI には、ソースインスタンスの ID を使用して自動的にタグ付けが行われます。この自動タグ付けが行われないようにするには、[可変タグ] から、`instance-id:$(instance-id)` タイルを削除します

iii. 他のタグを指定し、このスケジュールによって作成された AMI に割り当てるには、[タグを追加] をクリックします。

c. AMI の非推奨を設定します。

使用しなくなったAMI を非推奨にするには、AMI 非推奨セクションで、このスケジュールの AMI 非推奨を有効にするを選択し、AMI 非推奨ルールを指定します。AMI の非推奨ルールでは、AMI を非推奨にするタイミングを指定します。

スケジュールでカウントベースの AMI 保持を使用する場合は、非推奨にする最も古い AMI の数を指定する必要があります。非推奨カウントは、スケジュールの AMI 保持カウ

ント以下である必要があり、また1000 を超えることはできません。例えば、最大 5 つの AMI を保持するようにスケジュールが設定されている場合、最も古い5 つの AMI を非推奨にするようにスケジュールを設定することができます。

スケジュールで年齢ベースの AMI 保持を使用する場合は、AMI が非推奨になるまでの期間を指定する必要があります。非推奨カウントは、スケジュールの AMI 保持期間以下である必要があり、10年 (120か月、520 週、または3650日) を超えることはできません。例えば、AMI を10 日間保持するようにスケジュールが設定されている場合、作成後 10 日経過後に AMIを非推奨にするようにスケジュールを設定できます。

d. クロスリージョンでのコピーを設定します。

スケジュールによって作成された AMI を別のリージョンにコピーするには、[クロスリージョンのコピー] セクションで、[クロスリージョンコピーを有効化する] を選択します。AMI は、アカウント内で最大 3 つの異なるリージョンにコピーできます。送信先となるリージョンごとに、個別のクロスリージョンコピーのためのルールを指定する必要があります。

送信先リージョンごとに、以下を指定できます。

- AMIコピーの保持ポリシー。保持期間が終了すると、送信先リージョンのコピーは自動的に登録解除されます。
- AMIコピーの暗号化ステータス。ソース AMI が暗号化されている場合、または暗号化がデフォルトで有効化されている場合には、コピーされた AMI も常に暗号化されます。ソース AMI が暗号化されておらず、暗号化がデフォルトで無効になっている場合は、オプションで暗号化を有効にできます。KMS キー を指定しない場合、AMI は、各送信先リージョンにおける EBS 暗号化用のデフォルト KMS キー を使用して暗号化されます。送信先リージョンで KMS キーを指定する場合、選択した IAM ロールには KMS キーへのアクセス権が必要です。
- AMIコピーの非推奨ルール。非推奨期間が終了すると、AMI コピーは自動的に非推奨になります。非推奨期間は、コピーの保存期間以下である必要があり、また10 年を超えることはできません。
- ソース AMI からすべてのタグをコピーするか、タグをコピーしないかです。

> **Note**
>
> リージョンごとの、AMI の同時コピー数を超えないようにします。

e. 新たにスケジュールを追加するには、画面の上部にある [他のスケジュールを追加する] をクリックします。追加スケジュールごとに、このトピックの説明にならってフィールドを設定します。

f. 必要なスケジュールを追加したら、[ポリシーをレビュー] をクリックします。

12. ポリシーの概要を確認した後、[ポリシーを作成] をクリックします。

> **Note**
>
> `Role with name AWSDataLifecycleManagerDefaultRoleForAMIManagement already exists` エラーが発生した場合、詳細については「Amazon Data Lifecycle Manager の問題のトラブルシューティング」を参照してください。

Command line

AMI ライフサイクルポリシーを作成するには、create-lifecycle-policy コマンドを使用します。`PolicyType` で、`IMAGE_MANAGEMENT` を指定する。

> **Note**
>
> 構文を簡略化するために、次の例では、ポリシーの詳細を含む JSON ファイル、`policyDetails.json` を使用しています。

例 1: 年齢ベースの保持と AMI の非推奨

この例では、ターゲットインスタンスを再起動せずに、値が `production` で `purpose` のタグキーを持つすべてのインスタンスの AMI を作成する AMI ライフサイクルポリシーを作成します。ポリシーには、毎日`01:00` UTC時 に AMI を作成するスケジュールが 1 つ含まれます。ポリシーは AMI を2日間保持し、そして1 日後に非推奨にします。また、作成した AMI に対し、ソースインスタンスからタグもコピーされます。

```
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
```

次は、policyDetails.json ファイルの例です。

```
{
    "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
    }
}
```

リクエストが成功すると、コマンドは新しく作成されたポリシーの ID を返します。以下は出力の例です。

```
{
   "PolicyId": "policy-9876543210abcdef0"
}
```

例 2: クロスリージョンコピーを使用した、カウントベースの保持と AMI の非推奨

この例では、ターゲットインスタンスを再起動し、production の値の purpose タグキーを持つすべてのインスタンスの AMI を作成し、 AMI ライフサイクルポリシーを作成します。このポリシーには、17:30 UTC 時に始まる 6 時間ごとにAMIを作成するスケジュールが1 つ含まれます。ポリシーは 3 AMI を保持し、2 つの最も古い AMIを自動的に非推奨にします。また、AMIを us-east-1 にコピーし、2 つの AMI コピーを保持し、最も古い AMI を自動的に非推奨にするクロスリージョンコピールールもあります。

```
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
```

次は、policyDetails.json ファイルの例です。

```
{
    "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
        }]
    }]
}
```

AMI ライフサイクルポリシーに関する考慮事項

AMI ライフサイクルポリシーを作成する場合の一般的な考慮事項は次のとおりです。

- AMI ライフサイクルポリシーは、ポリシーと同じリージョンにあるインスタンスのみを対象としています。
- 最初の AMI 作成のオペレーションは、指定された開始時刻から 1 時間以内に開始されます。後続の AMI 作成オペレーションは、それらがスケジュールされた時刻から 1 時間以内に開始されます。
- Amazon Data Lifecycle Manager が AMI を登録解除すると、バッキングスナップショットが自動的に削除されます。
- ターゲットリソースタグでは大文字と小文字が区別されます。

- ポリシーによってターゲットにされたインスタンスからターゲットタグを削除した場合、以降、Amazon Data Lifecycle Manager では、標準階層に存在する AMI の管理は行いません。不要になった場合は手動で削除する必要があります。
- インスタンスをバックアップするために複数のポリシーを作成できます。例えば、インスタンスに 2 つのタグがあり、タグ A が 12 時間ごとに AMI を作成するポリシー A のターゲットであり、タグ B が 24 時間ごとに AMI を作成するポリシー B のターゲットである場合、Amazon Data Lifecycle Manager は両方のポリシーのスケジュールに従って AMI を作成します。または、複数のスケジュールを持つ単一のポリシーを作成することで、同じ結果を得ることができます。例えば、タグ A のみをターゲットとするポリシーを 1 つ作成し、スケジュールを 2 つ指定できます (1 つは 12 時間ごと、1 つは 24 時間ごと)。
- ポリシーの作成後にターゲットインスタンスにアタッチされた新しいボリュームは、次回のポリシー実行時に自動的にバックアップに含まれます。ポリシー実行時にインスタンスにアタッチされたすべてのボリュームが含まれます。
- AMI を 1 つだけ作成するように設定されたカスタム cron ベースのスケジュールでポリシーを作成した場合、保持のしきい値に達しても、その AMI はポリシーによって自動的に登録解除されることはありません。AMI が不要になった場合は、手動で登録解除する必要があります。
- 保持期間が作成頻度よりも短い経過日ベースのポリシーを作成した場合、Amazon Data Lifecycle Manager は次の AMI が作成されるまで常に最新の AMI を保持します。例えば、経過日ベースのポリシーで保存期間が 7 日間の AMI が毎月 1 つ作成される場合、Amazon Data Lifecycle Manager は、保持期間が 7 日間であっても、各 AMI を 1 か月間保持します。
- カウントベースのポリシーの場合、Amazon Data Lifecycle Manager は常に、保持ポリシーに従って最も古い AMI の登録解除を試みる前に、作成頻度に従って AMI を作成します。
- AMI を正常に登録解除し、関連するバッキングスナップショットを削除するには、数時間かかることがあります。以前に作成した AMI が正常に登録解除される前に Amazon Data Lifecycle Manager が次の AMI を作成した場合、一時的に保持数を超える数の AMI を保持する可能性があります。

ポリシーによってターゲットにされたインスタンスを終了する場合の考慮事項は次のとおりです。

- カウントベースの保持スケジュールが設定されたポリシーによってターゲットにされたインスタンスを終了すると、以前に終了したインスタンスから作成された AMI は、ポリシーで管理されなくなります。これらの以前に作成された AMI は、不要になったら手動で登録解除する必要があります。
- 経過時間ベースの保持スケジュールが設定されたポリシーで作成されたインスタンスを終了すると、ポリシーは定義されたスケジュールに従って、すでに作成された、最後の 1 つ前の AMI ま

で AMI を登録解除し続けます。最後の AMI が不要になった場合は、手動で削除する必要があります。

AMI ポリシーと AMI の非推奨に関する考慮事項は次のとおりです。

- カウントベースの保持を行っているスケジュールで AMI 非推奨カウントを増やすと、そのスケジュールによって作成されたすべての AMI (既存および新規) に変更が適用されます。
- 年齢ベースの保持でスケジュールの AMI の非推奨期間を長くした場合、その変更は新しい AMI にのみ適用されます。既存の AMIは影響を受けません。
- スケジュールからAMIの非推奨ルールを削除しても、Amazon Data Lifecycle Manager は、そのスケジュールで以前に非推奨となっていたAMIの非推奨をキャンセルしません。
- スケジュールの AMIの非推奨カウントや期間を減らしても、Amazon Data Lifecycle Manager は、そのスケジュールによって以前に非推奨とされたAMIの非推奨をキャンセルしません。
- AMI ポリシーで作成された AMI を手動で非推奨にしても、Amazon Data Lifecycle Manager は非推奨を上書きしません。
- AMI ポリシーで以前に非推奨とされたAMI の非推奨を手動でキャンセルしても、Amazon Data Lifecycle Manager はキャンセルを上書きしません。
- AMI が複数の競合するスケジュールで作成され、1 つ以上のスケジュールに AMI の非推奨ルールがない場合、Amazon Data Lifecycle Manager はその AMI を非推奨としません。
- AMI が複数の競合するスケジュールで作成され、それらのすべてのスケジュールに AMI の非推奨ルールがある場合、Amazon Data Lifecycle Manager は、非推奨に移行する日が最も遅くなる非推奨ルールを使用します。

次の考慮事項は、AMI ポリシーおよびごみ箱に適用されます。

- Amazon Data Lifecycle Manager がポリシーの保持しきい値に達したときに AMI の登録を解除してごみ箱に移動した時にその AMI をごみ箱から手動で復元する場合、AMI が不要になった際には手動で AMI の登録を解除する必要があります。Amazon Data Lifecycle Manager は、AMI を管理しなくなります。
- ポリシーによって作成された AMI を手動で登録解除し、ポリシーの保持しきい値に達したときにその AMI がごみ箱にある場合、Amazon Data Lifecycle Manager はスナップショットを登録解除しません。Amazon Data Lifecycle Manager は、AMI がごみ箱に保存されている間は、スナップショットを管理しません。

ポリシーの保持しきい値に達する前に AMI がごみ箱から復元された場合、Amazon Data Lifecycle Manager は、ポリシーの保持しきい値に達したときに AMI を削除します。

ポリシーの保持しきい値に達した後に AMI がごみ箱から復元された場合、Amazon Data Lifecycle Manager はその AMI を削除しません。不要になった場合は、手動で削除する必要があります。

以下は、エラー状態にある AMI ポリシーに関する考慮事項です。

- 期間ベースの保持スケジュールを持つポリシーの場合、ポリシーが `error` 状態の間に有効期限を迎える AMI は無期限に保持されます。これらの AMI は手動で登録解除する必要があります。ポリシーを再度有効にすると、Amazon Data Lifecycle Manager は保持期間が終了した時に AMI の登録解除を再開します。
- カウントベースの保持スケジュールが設定されているポリシーの場合、ポリシーが `error` 状態の間は AMI の作成と登録解除が停止されます。ポリシーを再度有効にすると、Amazon Data Lifecycle Manager は AMI の作成を再開し、保持しきい値に達した時に AMI の登録解除を再開します。

AMI ポリシーと AMI の無効化に関する考慮事項は次のとおりです。

- Amazon Data Lifecycle Manager が作成した AMI を無効化し、保持しきい値に達したときにその AMI が無効になっている場合、Amazon Data Lifecycle Manager は AMIを登録解除し、関連付けられたスナップショットを削除します。
- Amazon Data Lifecycle Manager が作成した AMI を無効にし、関連付けられたスナップショットを手動でアーカイブし、保持しきい値に達したときにそれらのスナップショットがアーカイブされた場合、Amazon Data Lifecycle Manager はそれらのスナップショットを削除せず、管理もできなくなります。

AMI ポリシーと AMI 登録解除保護には、以下の考慮事項が適用されます。

- Amazon Data Lifecycle Manager に作成された AMI の登録解除保護を手動で実行し、それでも AMI 保持しきい値に達して有効化されると、Amazon Data Lifecycle Manager はその AMI を管理しなくなります。不要になった場合は、AMI の登録を手動で解除して、基となるスナップショットを削除する必要があります。

その他のリソース

詳細については、「Amazon Data Lifecycle Manager ストレージを使用した Amazon EBS スナップショットと AMI 管理の自動化 AWS 」ブログを参照してください。

Data Lifecycle Manager を使用してクロスアカウントのスナップショットコピーを自動化

クロスアカウントのスナップショットのコピーを自動化すると、Amazon EBS スナップショットを分離アカウントの特定のリージョンにコピーし、暗号化キーを使用してそれらのスナップショットを暗号化できます。これにより、アカウントが侵害された場合にデータの損失から保護することができます。

アカウント間でスナップショットのコピーを自動化するには、次の 2 つのアカウントが使用されます。

- ソースアカウント — ソースアカウントは 、スナップショットを作成してターゲットアカウントと共有するアカウントです。このアカウントでは、設定された間隔でスナップショットを作成し、他の AWS アカウントと共有する EBS スナップショットポリシーを作成する必要があります。
- ターゲットアカウント— ターゲットアカウントは 、スナップショットを共有する共有先アカウントを持つアカウントで、共有スナップショットのコピーを作成するアカウントです。このアカウントでは、指定した 1 つ以上のソースアカウントによって共有されるスナップショットを自動的にコピーするクロスアカウントコピーイベントポリシーを作成する必要があります。

> **Note**
>
> ソースアカウントの EBS スナップショットポリシーとターゲットアカウントのクロスアカウントコピーイベントポリシーの両方を同じ AWS リージョンに作成する必要があります。これにより、ターゲットアカウントは必要に応じてスナップショットを異なる送信先リージョンにコピーできるようになります。

トピック

- クロスアカウントスナップショットコピーポリシーの作成
- スナップショット説明フィルターの指定
- クロスアカウントスナップショットコピーポリシーに関する考慮事項

- その他のリソース

クロスアカウントスナップショットコピーポリシーの作成

アカウント間でスナップショットをコピーするためにソースアカウントとターゲットアカウントを準備するには、次の手順を実行します。

手順 1: EBS スナップショットポリシーを作成する (ソースアカウント)

ソースアカウントで EBS スナップショットポリシーを作成します。これにより、スナップショットを作成し、必要なターゲットアカウントと共有します。

ポリシーを作成するときは、クロスアカウント共有を有効にし、スナップショットを共有するターゲット AWS アカウントを指定する必要があります。このアカウントは、スナップショットを共有するアカウントです。暗号化されたスナップショットを共有する場合は、選択したターゲットアカウントに、ソースボリュームの暗号化に使用された KMS キー を使用するためのアクセス権限を付与する必要があります。詳細については、「ステップ 2: カスタマーマネージド型キー (ソースアカウント) を共有する」を参照してください。

> **Note**
>
> - このポリシーは、ステップ 3 でターゲットアカウントのクロスアカウントコピーイベントポリシーを作成するリージョンと同じ AWS リージョンに作成します。クロスアカウントスナップショット共有が正しく機能するには、両方のポリシーが同じリージョンにある必要があります。
> - 共有できるのは、暗号化されていないスナップショットまたはカスタマーマネージド型キーを使用して暗号化されたスナップショットだけです。デフォルトの EBS 暗号化 KMS キー で暗号化されたスナップショットを共有することはできません。暗号化されたスナップショットを共有する場合は、ソースボリュームの暗号化に使用された KMS キーも、ターゲットアカウントと共有する必要があります。詳細については、「AWS Key Management Service デベロッパーガイド」の「他のアカウントのユーザーに KMS キーの使用を許可する」を参照してください。

EBS スナップショットポリシーを作成する方法については、EBS スナップショット用の Amazon Data Lifecycle Manager カスタムポリシーを作成を参照してください。

EBS スナップショットポリシーを作成するには、次のいずれかの方法を使用します。

ステップ 2: カスタマーマネージド型キー (ソースアカウント) を共有する

暗号化されたスナップショットを共有する場合は、IAM ロールと (前のステップで選択した) ターゲットの AWS アカウントに、ソースボリュームの暗号化に使用されたカスタマーマネージド型キーを使用するためのアクセス権限を付与する必要があります。

> **Note**
>
> この手順は、暗号化されたスナップショットを共有する場合にのみ実行してください。暗号化されていないスナップショットを共有する場合は、この手順をスキップします。

Console

1. https://console.aws.amazon.com/kms で AWS KMS コンソールを開きます。
2. を変更するには AWS リージョン、ページの右上隅にあるリージョンセレクターを使用します。
3. ナビゲーションペインで、[カスタマーマネージド型キー] を選択してから、ターゲットアカウントと共有する必要がある KMS キーを選択します。

   KMS キー の ARN を記録しておきます。これは後で必要になります。
4. [キーポリシー] タブで、[キーユーザー] セクションまで下にスクロールします。[追加] を選択し、前のステップで選択した IAM ロールの名前を入力してから、[追加] をクリックします。
5. [キーポリシー] タブで、[その他の AWS アカウント] セクションまで下にスクロールします。他の AWS アカウントを追加を選択し、前のステップでスナップショットを共有するために選択したすべてのターゲット AWS アカウントを追加します。
6. [Save changes] (変更の保存) をクリックします。

Command line

KMS キー に現在アタッチされているキーポリシーを取得するには、get-key-policy コマンドを使用します。

例えば、次のコマンドは、`9d5e2b3d-e410-4a27-a958-19e220d83a1e` の ID を持つ KMS キー のキーポリシーを取得し、`snapshotKey.json` という名前のファイルに書き込みます。

```
$ aws kms get-key-policy \
```

```
    --policy-name default \
    --key-id 9d5e2b3d-e410-4a27-a958-19e220d83a1e \
    --query Policy \
    --output text > snapshotKey.json
```

任意のテキストエディタを使用してキーポリシーを開きます。スナップショットポリシーの作成時に指定した IAM ロールの ARN と、KMS キー を共有するターゲットアカウントの ARN を追加します。

たとえば、次のポリシーでは、デフォルトの IAM ロールの ARN とターゲットアカウントのルートアカウントの ARN が指定されています。 222222222222.

> **Tip**
>
> 最小権限のプリンシパルに従うには、kms:CreateGrant へのフルアクセスを許可しないでください。代わりに、次の例に示すように、 AWS サービスによってユーザーに代わって権限が作成された場合にのみ、 kms:GrantIsForAWSResource条件キーを使用して、ユーザーが KMS キーに権限を作成できるようにします。

```
{
    "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"
        }
    }
}
```

ファイルを保存して閉じます。次に、put-key-policy コマンドを使用して、更新されたキーポリシーを KMS キー にアタッチします。

```
$ aws kms put-key-policy \
    --policy-name default \
    --key-id 9d5e2b3d-e410-4a27-a958-19e220d83a1e \
    --policy file://snapshotKey.json
```

手順 3: クロスアカウントコピーイベントポリシーの作成 (ターゲットアカウント)

ターゲットアカウントで、必要なソースアカウントで共有されるスナップショットを自動的にコピーするクロスアカウントコピーイベントポリシーを作成する必要があります。

このポリシーは、指定されたソースアカウントの 1 つがスナップショットを共有する場合にのみ、ターゲットアカウントで実行されます。

> **Note**
>
> このポリシーは、ステップ 1 で作成したソースアカウントの EBS スナップショットポリシーと同じ AWS リージョンに作成します。クロスアカウントスナップショット共有が正し

く機能するには、両方のポリシーが同じリージョンにある必要があります。その後、必要に応じて異なる送信先リージョンにスナップショットをコピーするようにこのポリシーを設定できます。

クロスアカウントコピーイベントポリシーを作成するには、次のいずれかの方法を使用します。

Console

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインで、[Elastic Block Store]、[ライフサイクルマネージャー]、[ライフサイクルポリシーの作成] の順に選択します。
3. [ポリシータイプの選択] 画面で、[クロスアカウントコピーのイベントポリシー] を選択した上で、[次へ] をクリックします。
4. [ポリシーの説明] に、ポリシーの簡単な説明を入力します。
5. [ポリシータグ] に、ライフサイクルポリシーに適用されるタグを追加します。これらのタグは、ポリシーを識別および分類するために使用することができます。
6. [イベントの設定] セクションで、ポリシーを実行するスナップショット共有イベントを定義します。以下の操作を実行します。

   a. 共有アカウントでは、共有スナップショットをコピーするソース AWS アカウントを指定します。アカウントの追加 を選択し、12 桁の AWS アカウント ID を入力し、追加 を選択します。

   b. [説明でフィルタリング] に、正規表現を使用して必要なスナップショットの説明を入力します。指定したソースアカウントによって共有され、指定したフィルターに一致する説明を持つスナップショットのみが、ポリシーによってコピーされます。詳細については、を参照してくださいスナップショット説明フィルターの指定

7. [IAM ロール] で、スナップショットのコピーアクションを実行するアクセス許可を持つ IAM ロールを選択します。Amazon Data Lifecycle Manager から提供されるデフォルトのロールを使用するには、[デフォルトロール] を選択します。以前に作成したカスタム IAM ロールを使用する場合には、[別のロールを選択] をクリックした上で、使用するロールを選択します。

   暗号化されたスナップショットをコピーする場合は、ソースボリュームの暗号化に使用する暗号化 KMS キー を使用するためのアクセス権限を選択した IAM ロールに付与する必要があります。同様に、別の KMS キー を使用して送信先リージョンのスナップショットを暗号

化する場合は、送信先 KMS キー を使用するためのアクセス権限を IAM ロールに付与する必要があります。詳細については、を参照してくださいステップ 4: IAM ロールに必要な KMS キー の使用を許可する (ターゲットアカウント)

8. [コピーアクション] セクションで、アクティブ化された際にポリシーが実行する、スナップショットのコピーアクションを定義します。ポリシーは、スナップショットを最大 3 つのリージョンにコピーできます。送信先となるリージョンごとに、個別のコピールールを指定する必要があります。追加したルールごとに以下を実行します。

   a. [名前] に、コピーアクションのわかりやすい名前を入力します。

   b. [Target Region] (ターゲットリージョン) で、スナップショッのをコピー先リージョンを選択します。

   c. [有効期限] では、作成したスナップショットのコピーを、ターゲットリージョンに保持する期間を指定します。

   d. スナップショットのコピーを暗号化するには、[暗号化] で、[暗号化の有効化] を選択します。ソーススナップショットが暗号化されている場合、またはアカウントで暗号化がデフォルトで有効になっている場合は、ここで暗号化を有効しなくても、スナップショットのコピーは常に暗号化されます。ソーススナップショットが暗号化されておらず、アカウントで暗号化がデフォルトで有効になっていない場合は、暗号化を有効または無効にすることができます。暗号化を有効にし、KMS キー を指定しない場合、スナップショットは、各送信先リージョンでデフォルトの暗号化 KMS キー を使用して暗号化されます。送信先のリージョンの KMS キー を指定する場合は、KMS キー へのアクセスが必要です。

9. さらに、スナップショットのコピーアクションを追加するには、[新しいリージョンを追加] をクリックします。

10. [Policy status after creation (作成後のポリシーの状態)] では、[Enable policy (ポリシーの有効化)] を選択すると、次のスケジュールした時刻にポリシーが実行されます。ポリシーが実行されないようにするには、[Disable policy (ポリシーの無効化)] を選択します。ここでポリシーを有効にしない場合、作成後に手動で有効にするまで、スナップショットのコピーは開始されません。

11. [Create policy] (ポリシーの作成) を選択します。

Command line

ポリシーを作成するには、create-lifecycle-policy コマンドを使用します。クロスアカウントコピーイベントポリシーを作成するには、`PolicyType` で、`EVENT_BASED_POLICY` を指定します。

例えば、次のコマンドは、ターゲットアカウント 222222222222 にクロスアカウントコピーイベントポリシーを作成します。ポリシーは、ソースアカウント 111111111111 によって共有されるスナップショットをコピーします。ポリシーは、スナップショットを sa-east-1 と eu-west-2 にコピーします。sa-east-1 にコピーされたスナップショットは暗号化されず、3 日間保持されます。eu-west-2 にコピーされたスナップショットは KMS キー 8af79514-350d-4c52-bac8-8985e84171c7 を使用して暗号化され、1 か月間保持されます。このポリシーは、デフォルトの IAM ロールを使用します。

```
$ 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
```

以下は、policyDetails.json ファイルの内容を示しています。

```
{
    "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"
            }
        }]
    }]
}
```

リクエストが成功すると、コマンドは新しく作成されたポリシーの ID を返します。以下は出力の例です。

```
{
    "PolicyId": "policy-9876543210abcdef0"
}
```

ステップ 4: IAM ロールに必要な KMS キー の使用を許可する (ターゲットアカウント)

暗号化されたスナップショットをコピーする場合は、(前の手順で選択した) IAM ロールに、ソースボリュームの暗号化に使用された カスタマーマネージド型キー を使用するためのアクセス権限を付与する必要があります。

> **Note**
>
> 暗号化されたスナップショットをコピーする場合のみ、この手順を実行してください。暗号化されていないスナップショットをコピーする場合は、この手順をスキップします。

以下のいずれかの方法を使用して、必要なポリシーを IAM ロールに追加します。

Console

1. IAM コンソール (https://console.aws.amazon.com/iam/) を開きます。
2. ナビゲーションペインで、[ロール] を選択します。前の手順でクロスアカウントコピーイベントポリシーを作成したときに選択した IAM ロールを検索して選択します。デフォルトのロールを使用することを選択した場合、ロールの名前は AWSDataLifecycleManagerDefaultRole になります。

3. [インラインポリシーの追加] を選択し、次に [JSON] タブを選択します
4. 既存のポリシーを次のように置き換え、ステップ 2 でソースボリュームの暗号化に使用され、ソースアカウントによって共有された KMS キーの ARN を指定します。

> **Note**
>
> 複数のソースアカウントからコピーする場合は、各ソースアカウントから対応する KMS キー ARN を指定する必要があります。

次の例では、ポリシーは、ソースアカウント 111111111111 で共有された KMS キー 1234abcd-12ab-34cd-56ef-1234567890ab とターゲットアカウント 222222222222 に存在する KMS キー 4567dcba-23ab-34cd-56ef-0987654321yz を使用するためのアクセス権限を IAM ロールに付与します。

> **Tip**
>
> 最小権限のプリンシパルに従うには、kms:CreateGrant へのフルアクセスを許可しないでください。代わりに、次の例に示すように、 AWS サービスによってユーザーに代わって権限が作成された場合にのみ、 kms:GrantIsForAWSResource条件キーを使用して、ユーザーが KMS キーに権限を作成できるようにします。

JSON

```
{
   "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. [Review policy] (ポリシーの確認) を選択します。
6. [名前] にポリシーのわかりやすい名前を入力し、[ポリシーの作成] を選択します。

Command line

お好みのテキストエディタを使用して、policyDetails.json という名前の新しい JSON ファイルを作成します。以下のポリシーを追加し、ステップ 2 でソースボリュームの暗号化に使用され、ソースアカウントによって共有された KMS キーの ARN を指定します。

> **Note**
>
> 複数のソースアカウントからコピーする場合は、各ソースアカウントから対応する KMS キー ARN を指定する必要があります。

次の例では、ポリシーは、ソースアカウント 111111111111 で共有された KMS キー 1234abcd-12ab-34cd-56ef-1234567890ab とターゲットアカウント 222222222222 に存在する KMS キー 4567dcba-23ab-34cd-56ef-0987654321yz を使用するためのアクセス権限を IAM ロールに付与します。

> **Tip**
>
> 最小権限のプリンシパルに従うには、kms:CreateGrant へのフルアクセスを許可しないでください。代わりに、次の例に示すように、 AWS サービスによってユーザーに代わって権限が作成された場合にのみ、 kms:GrantIsForAWSResource条件キーを使用して、ユーザーが KMS キーに権限を作成できるようにします。

JSON

```
{
   "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"
            ]
        }
    ]
}
```

ファイルを保存して閉じます。次に、put-role-policy コマンドを使用して、IAM ロールにポリシーを追加します。

例

```
$ aws iam put-role-policy \
    --role-name AWSDataLifecycleManagerDefaultRole \
    --policy-name CopyPolicy \
    --policy-document file://AdminPolicy.json
```

スナップショット説明フィルターの指定

ターゲットアカウントでスナップショットコピーポリシーを作成する場合は、スナップショットの説明フィルターを指定する必要があります。スナップショット説明フィルターにより、ポリシーによってコピーされるスナップショットを制御できる追加のフィルターレベルを指定できます。つまり、スナップショットは、指定したソースアカウントのいずれかによって共有され、指定したフィルターに一致するスナップショットの説明がある場合にのみ、ポリシーによりコピーされます。つまり、スナップショットが指定されたコースアカウントのいずれかによって共有されているのに、指定したフィルターに一致する説明がない場合、そのスナップショットはポリシーによってコピーされません。

スナップショットフィルターの説明は、正規表現を使用して指定する必要があります。コンソールとコマンドラインを使用してクロスアカウントコピーイベントポリシーを作成する場合、このフィールドは必須です。使用できる正規表現の例を次に示します。

- `.*`— このフィルターは、すべてのスナップショットの説明に一致します。この式を使用すると、ポリシーは、指定したソースアカウントの 1 つが共有しているすべてのスナップショットをコピーします。
- `Created for policy: policy-0123456789abcdef0.*`— このフィルターは、`policy-0123456789abcdef0` の ID を持ったポリシーによって作成されたスナップショットにのみ一致します。このような式を使用すると、ポリシーは、指定したソースアカウントのいずれかによってアカウントと共有され、指定された ID を持つポリシーによって作成されたスナップショットのみをコピーします。
- `.*production.*`— このフィルターは、説明のいずれかの場所に `production` の単語が含まれているスナップショットに一致します。この式を使用すると、ポリシーは、指定したソースアカウントのいずれかで共有され、説明に指定されたテキストを含むすべてのスナップショットをコピーします。

クロスアカウントスナップショットコピーポリシーに関する考慮事項

以下はアカウント間のコピーイベントポリシーの考慮事項です。

- ソースアカウントの EBS スナップショットポリシーとターゲットアカウントのクロスアカウントコピーイベントポリシーは、同じ AWS リージョンで作成する必要があります。スナップショットが共有されると、ターゲットアカウントポリシーは、コピーアクションで指定された異なる送信先リージョンにスナップショットをコピーできます。
- コピーできるのは、暗号化されていないスナップショットまたは カスタマーマネージド型キー を使用して暗号化されたスナップショットだけです。
- Amazon Data Lifecycle Manager の外部で共有されるスナップショットをコピーするクロスアカウントコピーイベントポリシーを作成できます。
- ターゲットアカウントのスナップショットを暗号化する場合、クロスアカウントコピーイベントポリシー用に選択された IAM ロールには、必要な KMS キー を使用するためのアクセス権限が必要です。

その他のリソース

詳細については、「Automating copying encrypted Amazon EBS snapshots across AWS accounts AWS storage」ブログを参照してください。

Amazon Data Lifecycle Manager デフォルトポリシーの変更

Amazon Data Lifecycle Manager ポリシーを変更するときは、次の点に注意してください。

- ターゲットタグを削除して AMI またはスナップショットポリシーを変更すると、それらのタグを持つボリュームまたはインスタンスはポリシーで管理されなくなります。
- スケジュール名を変更すると、古いスケジュール名で作成されたスナップショットまたは AMI はポリシーで管理されなくなります。
- 経過時間ベースの保持スケジュールを、新たな期間を使用するスケジュールに修正すると、新たな期間は、変更後に作成された新たなスナップショットまたは AMI にのみ適用されます。新たなスケジュールは、変更前に作成されたスナップショットまたは AMI の保持スケジュールに影響を及ぼすことはありません。
- 作成後、ポリシーの保持スケジュールをカウントベースから経過時間ベースに変更することはできません。この変更を行うには、新たなポリシーを作成する必要があります。
- 期間ベースの保持スケジュールを持つポリシーを無効にすると、ポリシーが無効になっている間に有効期限が切れるように設定されたスナップショットまたは AMI は無期限に保持されます。スナップショットを削除するか、AMI を手動で登録解除する必要があります。ポリシーを再度有効にすると、Amazon Data Lifecycle Manager は保持期間の終了時にスナップショットの削除または AMI の登録解除を再開します。
- カウントベースの保持スケジュールを持つポリシーを無効にすると、ポリシーはスナップショットまたは AMI の作成と削除を停止します。ポリシーを再度有効にすると、Amazon Data Lifecycle Manager はスナップショットと AMI の作成を再開し、保持しきい値に達するとスナップショットまたは AMI の削除を再開します。
- スナップショットアーカイブを有効にしたポリシーを含むポリシーを無効にした時点で、アーカイブ層に残されているスナップショットに対しては、Amazon Data Lifecycle Manager による管理が行われなくなります。不要になったスナップショットは、手動で削除する必要があります。
- カウントベースのスケジュールでスナップショットのアーカイブを有効にした場合、このアーカイブルールは、以後スケジュールに従って作成およびアーカイブされるすべての新しいスナップショットに適用されます。また、このスケジュールによって以前に作成およびアーカイブされた既存のスナップショットにも適用されます。
- 期間ベースのスケジュールに対しスナップショットのアーカイブを有効にした場合、アーカイブルールは、有効化した時点以降に作成された新しいスナップショットにのみ適用されます。スナップショットのアーカイブを有効にする前に作成された既存のスナップショットは、それらのスナップショットが最初に作成およびアーカイブされた時点で設定されていたスケジュールに従って、引き続きそれぞれのストレージ階層から削除されます。

- カウントベースのスケジュールでスナップショットのアーカイブを無効にすると、スケジュールはただちにスナップショットのアーカイブを停止します。スケジュールによって以前にアーカイブされたスナップショットはアーカイブ階層に残り、Amazon Data Lifecycle Manager によって削除されることはありません。
- 期間ベースのスケジュールにおいてスナップショットのアーカイブを無効にすると、ポリシーで作成されアーカイブが予定されているスナップショットは、スケジュールされた (`aws:dlm:expirationTime` システムタグが示す) アーカイブの日時に完全に削除されます。
- スケジュールにおいてスナップショットのアーカイブを無効にした場合、スケジュールは、その時点でスナップショットのアーカイブを停止します。スケジュールによって以前にアーカイブされたスナップショットはアーカイブ階層に残り、Amazon Data Lifecycle Manager によって削除されることはありません。
- カウントベースのスケジュールにおいてアーカイブ保持数を変更した場合、この新しい保存数は、以前にスケジュールによってアーカイブされた既存のスナップショットの数も含みます。
- 期間ベースのスケジュールにおいてアーカイブの保持期間を変更した場合は、この変更を行った時点後にアーカイブされたスナップショットに対してのみ、新しい保持期間が適用されます。

次のいずれかの手順に従ってライフサイクルポリシーを変更します。

Console

ライフサイクルポリシーを変更するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインで [Elastic Block Store]、[ライフサイクルマネージャー] の順に選択します。
3. リストからライフサイクルポリシーを選択します。
4. [アクション]、[ライフサイクルポリシーの変更] の順に選択します。
5. 必要に応じてポリシー設定を修正します。具体例を挙げると、スケジュールを修正する、タグを追加もしくは削除する、またはポリシーを有効化もしくは無効化することができます。
6. [ポリシーの変更] を選択します。

Command line

ライフサイクルポリシーに関する情報を変更するには、update-lifecycle-policy コマンドを使用します。構文を簡略化するために、この例では、ポリシーの詳細を含む JSON ファイル、policyDetailsUpdated.json を参照しています。

```
aws dlm update-lifecycle-policy \
    --state DISABLED \
    --execution-role-arn
 arn:aws:iam::12345678910:role/AWSDataLifecycleManagerDefaultRole" \
    --policy-details file://policyDetailsUpdated.json
```

次は、policyDetailsUpdated.json ファイルの例です。

```
{
   "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
        }
    ]
}
```

更新されたポリシーを表示するには、`get-lifecycle-policy` コマンドを使用します。状態、タグの値、スナップショットの間隔、およびスナップショットの開始時刻が変更されたことがわかります。

Amazon Data Lifecycle Manager のポリシーを削除

Amazon Data Lifecycle Manager ポリシーを削除するときは、次の点に注意してください。

- ポリシーを削除した場合でも、そのポリシーによって作成されたスナップショットまたは AMI は自動的に削除されません。これらのスナップショットや AMI が不要になった場合は、手動で削除する必要があります。
- スナップショットのアーカイブが有効になっているポリシーを削除すると、その時点でアーカイブ層にあったスナップショットには、Amazon Data Lifecycle Manager による管理が行われなくなります。不要になったスナップショットは、手動で削除する必要があります。
- アーカイブが有効化された期間ベースのスケジュールを含むポリシーを削除すると、そのポリシーによって作成されアーカイブが予定されていたスナップショットは、スケジュールされた (`aws:dlm:expiretime` システムタグで示されている) アーカイブの日時に完全に削除されます。

次のいずれかの手順に従ってライフサイクルポリシーを削除します。

Console

ライフサイクルポリシーを削除するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインで [Elastic Block Store]、[ライフサイクルマネージャー] の順に選択します。
3. リストからライフサイクルポリシーを選択します。
4. [アクション]、[ライフサイクルポリシーの削除] の順に選択します。
5. 確認を求めるメッセージが表示されたら、[ポリシーの削除] を選択します。

Command line

ライフサイクルポリシーを削除し、ポリシーで指定されたターゲットタグを解放して再利用できるようにするには、delete-lifecycle-policy コマンドを使用します。

> **Note**
>
> Amazon Data Lifecycle Manager によって作成されたスナップショットだけを削除できます。

```
aws dlm delete-lifecycle-policy --policy-id policy-0123456789abcdef0
```

Amazon Data Lifecycle Manager API リファレンスには、Amazon Data Lifecycle Manager クエリ API の各アクションとデータ型の説明と構文があります。

または、いずれかの AWS SDKs を使用して、使用しているプログラミング言語またはプラットフォームに合わせて API にアクセスすることもできます。詳細については、AWS SDK を参照してください。

IAM を使用して Amazon Data Lifecycle Manager へのアクセスを制御

Amazon Data Lifecycle Manager へのアクセスには、認証情報が必要です。それらの資格情報には、インスタンス、ボリューム、スナップショット、AMIなどの AWS リソースにアクセスするためのアクセス許可が必要です。

Amazon Data Lifecycle Manager を使用するには、次の IAM 許可が必要です。

> **Note**
>
> - `ec2:DescribeAvailabilityZones`、`ec2:DescribeRegions`、`kms:ListAliases`、および `kms:DescribeKey` 許可は、コンソールユーザーにのみ必要です。コンソールへのアクセスが不要な場合は、許可を削除できます。
> - AWSDataLifecycleManagerDefaultRole ロールの ARN 形式は、コンソールを使用して作成されたか、 AWS CLIを使用して作成されたかによって異なります。コンソールを

使用してロールが作成された場合、ARN 形式は arn:aws:iam::*account_id*:role/service-role/AWSDataLifecycleManagerDefaultRole です。ロールが を使用して作成された場合 AWS CLI、ARN 形式は ですarn:aws:iam::*account_id*:role/AWSDataLifecycleManagerDefaultRole。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "dlm:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": [
                "arn:aws:iam::111122223333:role/
AWSDataLifecycleManagerDefaultRole",
                "arn:aws:iam::111122223333:role/
AWSDataLifecycleManagerDefaultRoleForAMIManagement",
                "arn:aws:iam::111122223333:role/service-role/
AWSDataLifecycleManagerDefaultRole",
                "arn:aws:iam::111122223333:role/service-role/
AWSDataLifecycleManagerDefaultRoleForAMIManagement"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "iam:ListRoles",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeAvailabilityZones",
                "ec2:DescribeRegions",
                "kms:ListAliases",
                "kms:DescribeKey"
```

```
            ],
            "Resource": "*"
        }
    ]
}
```

暗号化のアクセス許可

Amazon Data Lifecycle Manager および暗号化されたリソースを操作する場合は、次の点を考慮してください。

- ソースボリュームが暗号化されている場合は、Amazon Data Lifecycle Manager のデフォルトロール (AWSDataLifecycleManagerDefaultRole および AWSDataLifecycleManagerDefaultRoleForAMIManagement) に、ボリュームの暗号化に使用される KMS キーを使用するアクセス許可があることを確認してください。
- 暗号化されていないスナップショットまたは暗号化されていないスナップショットによってバックアップされた AMIs に対してクロスリージョンコピーを有効にし、送信先リージョンで暗号化を有効にすることを選択した場合は、デフォルトのロールに、送信先リージョンで暗号化を実行するために必要な KMS キーを使用するアクセス許可があることを確認してください。
- 暗号化されたスナップショットまたは暗号化されたスナップショットにバックアップされた AMIs に対してクロスリージョンコピーを有効にする場合は、デフォルトのロールに、送信元と送信先の KMS キーの両方を使用するアクセス許可があることを確認してください。
- 暗号化されたスナップショットのスナップショットアーカイブを有効にする場合は、Amazon Data Lifecycle Manager のデフォルトロール (AWSDataLifecycleManagerDefaultRole には、スナップショットの暗号化に使用される KMS キーを使用するアクセス許可があることを確認してください。

詳細については、AWS Key Management Service デベロッパーガイドの「他のアカウントのユーザーに KMS キーの使用を許可する」を参照してください。

詳細については、「AWS Identity and Access Management ユーザーガイド」の「IAM ユーザーのアクセス許可を変更する」を参照してください。

AWS Amazon Data Lifecycle Manager の マネージドポリシー

AWS 管理ポリシーは、 によって作成および管理されるスタンドアロンポリシーです AWS。 AWS 管理ポリシーは、多くの一般的なユースケースにアクセス許可を付与するように設計されています。

AWS 管理ポリシーを使用すると、ポリシーを自分で記述する必要があったよりも、ユーザー、グループ、ロールに適切なアクセス許可を割り当てることが効率的になります。

ただし、 AWS 管理ポリシーで定義されているアクセス許可を変更することはできません。 AWS は、 AWS 管理ポリシーで定義されているアクセス許可を更新することがあります。行われた更新は、ポリシーがアタッチされているすべてのプリンシパルエンティティ (ユーザー、グループ、ロール) に影響します。

Amazon Data Lifecycle Manager は、一般的なユースケース用の AWS マネージドポリシーを提供します。これらのポリシーでは、より効率的に適切なアクセス許可を定義し、リソースへのアクセスを制御できます。Amazon Data Lifecycle Manager が提供する AWS マネージドポリシーは、Amazon Data Lifecycle Manager に渡すロールにアタッチされるように設計されています。

トピック

- AWSDataLifecycleManagerServiceRole
- AWSDataLifecycleManagerServiceRoleForAMIManagement
- AWSDataLifecycleManagerSSMFullAccess
- AWS マネージドポリシーの更新

AWSDataLifecycleManagerServiceRole

AWSDataLifecycleManagerServiceRole ポリシーは、Amazon Data Lifecycle Manager に適切なアクセス許可を付与し、Amazon EBSスナップショット ポリシーおよびクロスアカウントコピーイベントポリシーを作成、管理します。

JSON

```
{
    "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",
                "ec2:DescribeAvailabilityZones"
            ],
            "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

[AWSDataLifecycleManagerServiceRoleForAMIManagement] ポリシーは、Amazon Data Lifecycle Manager に適切なアクセス権限を付与し、Amazon EBS-backed AMI ポリシーを作成および管理します。

JSON

```
{
    "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

すべての Amazon EC2 インスタンスで事前スクリプトと事後スクリプトを実行するために必要な Systems Manager アクションを実行する権限を Amazon Data Lifecycle Manager に付与します。

> **Important**
>
> このポリシーでは、事前スクリプトと事後スクリプトを使用するときに、`aws:ResourceTag` 条件キーを使って特定の SSM ドキュメントへのアクセスを制限します。Amazon Data Lifecycle Manager が SSM ドキュメントにアクセスできるようにするには、SSM ドキュメントに がタグ付けされていることを確認する必要がありますDLMScriptsAccess:true。

JSON

```
{
    "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 マネージドポリシーの更新

AWS サービスは、 AWS 管理ポリシーを維持および更新します。 AWS 管理ポリシーのアクセス許可は変更できません。サービスは、新機能をサポートするために、 AWS 管理ポリシーに追加のアクセス許可を追加することがあります。この種類の更新はポリシーがアタッチされている、すべてのアイデンティティ (ユーザー、グループおよびロール) に影響を与えます。サービスは、新機能が起動されたとき、または新しいオペレーションが利用可能になったときに、 AWS マネージドポリシーを更新する可能性が最も高いです。サービスは AWS マネージドポリシーからアクセス許可を削除しないため、ポリシーの更新によって既存のアクセス許可が損なわれることはありません。

次の表は、このサービスがこれらの変更の追跡を開始してからの Amazon Data Lifecycle Manager の AWS マネージドポリシーの更新に関する詳細を示しています。このページへの変更に関する自動通知を受けるには、Amazon EBS ユーザーガイドのドキュメント履歴 の RSS フィードを購読してください。

変更	説明	日付
AWSDataLifecycleManagerServiceRole — ポリシーのアクセス許可を更新しました。	ローカルゾーンに関する情報を取得するアクセス許可をスナップショットポリシーに付与する `ec2:DescribeAvailabilityZones` アクションを、Amazon Data Lifecycle Manager に追加しました。	2024 年 12 月 16 日
AWSDataLifecycleManagerSSMFullAccess – ポリシーのアクセス	`AWSSystemsManagerSAP-CreateDLMSnapshotForSAPHANA` SSM ド	2023 年 11 月 17 日

変更	説明	日付
許可を更新しました。	キュメントを使用した SAP HANA で、アプリケーション整合性のあるスナップショットをサポートするようにポリシーを更新しました。	
AWSDataLifecycleManagerSSMFullAccess — 新しい AWS 管理ポリシーを追加しました。	Amazon Data Lifecycle Manager に AWSDataLifecycleManagerSSMFullAccess AWS 管理ポリシーが追加されました。	2023 年 11 月 7 日

変更	説明	日付
AWSDataLifecycleManagerServiceRole – スナップショットのアーカイブをサポートするための、アクセス許可を追加しました。	Amazon Data Lifecycle Manager に `ec2:ModifySnapshotTier` および `ec2:DescribeSnapshotTierStatus` アクションが追加され、スナップショットのアーカイブおよびスナップショットのアーカイブステータスの確認に必要なアクセス許可をスナップショットポリシーに付与できるようになりました。	2022 年 9 月 30 日

変更	説明	日付
[AWSDataLifecycleManagerServiceRoleForAMIManagement]— AMI の非推奨をサポートする権限を追加しました。	Amazon Data Lifecycle Manager は、`ec2:EnableImageDeprecation` および`ec2:DisableImageDeprecation` アクションを使用して、AMI の非推奨を有効または無効にするアクセス権限を EBS-backed AMI ポリシーに付与します。	2021 年 8 月 23 日
Amazon Data Lifecycle Manager が変更追跡を開始しました	Amazon Data Lifecycle Manager は AWS 、管理ポリシーの変更の追跡を開始しました。	2021 年 8 月 23 日

Amazon Data Lifecycle Manager 用の IAM サービスロール

AWS Identity and Access Management (IAM) ロールは、 AWS アイデンティティができることとできないことを決定するアクセス許可ポリシーを持つアイデンティティであるという点で、ユーザーと似ています AWS。ただし、ユーザーは 1 人の特定の人に一意に関連付けられますが、ロールはそれを必要とする任意の人が引き受けるようになっています。サービスロールは、ユーザーに代わってアクションを実行するために AWS サービスが引き受けるロールです。お客様に代わってバックアップ操作を実行するサービスとして、Amazon Data Lifecycle Managerでは、お客様に代わってポリシー

操作を実行するときに、想定するロールを渡す必要があります。IAM ロールの詳細については、IAM ユーザーガイドのIAM ロールを参照してください。

Amazon Data Lifecycle Manager に渡すロールには、Amazon Data Lifecycle Manager がポリシー操作に関連するアクション (スナップショットと AMI の作成、スナップショットと AMI のコピー、スナップショットの削除、AMI の登録解除など) を実行できるようにするアクセス権限を持つ IAM ポリシーが必要です。Amazon Data Lifecycle Manager のポリシータイプごとに異なるアクセス権限が必要です。ロールには、Amazon Data Lifecycle Manager が信頼されたエンティティとして登録されている必要があります。これにより、Amazon Data Lifecycle Manager はロールを引き受けることができます。

トピック

- Amazon Data Lifecycle Manager のデフォルトのサービスロール
- Amazon Data Lifecycle Manager のカスタムサービスロール

Amazon Data Lifecycle Manager のデフォルトのサービスロール

Amazon Data Lifecycle Manager は、以下のデフォルトのサービスロールを使用します：

- AWSDataLifecycleManagerDefaultRole — スナップショットを管理するためのデフォルトのロール。`dlm.amazonaws.com`サービスのみを信頼してロールを引き受け、Amazon Data Lifecycle Manager は、お客様に代わってスナップショットおよびクロスアカウントのスナップショットコピーポリシーで必要なアクションを実行できます。このロールは `AWSDataLifecycleManagerServiceRole` AWS マネージドポリシーを使用します。

  > **Note**
  >
  > ロールの ARN 形式は、コンソールを使用して作成されたか、 AWS CLIを使用して作成されたかによって異なります。コンソールを使用してロールが作成された場合、ARN 形式は `arn:aws:iam::account_id:role/service-role/AWSDataLifecycleManagerDefaultRole` です。ロールが を使用して作成された場合 AWS CLI、ARN 形式は ですarn:aws:iam::account_id:role/AWSDataLifecycleManagerDefaultRole。

- AWSDataLifecycleManagerDefaultRoleForAMIManagement— AMI を管理するためのデフォルトロール。`dlm.amazonaws.com`サービスのみを信頼してロールを引き受けます。これにより、Amazon Data Lifecycle Manager が、お客様に代わってEBS-backed AMI ポリシーで必要なアクションを実行できるようになります。このロールは

AWSDataLifecycleManagerServiceRoleForAMIManagement AWS マネージドポリシーを使用します。

Amazon Data Lifecycle Manager コンソールを使用している場合、Amazon Data Lifecycle Managerは、スナップショットまたはクロスアカウントのスナップショットコピーポリシーを最初に作成したときに、AWSDataLifecycleManagerDefaultRole サービスロールを自動的に作成します。同時にEBS-backed AMIポリシーを最初に作成したときに、AWSDataLifecycleManagerDefaultRoleForAMIManagement サービスロールを自動的に作成します。

コンソールを使用していない場合は、サービスロールを手動で作成するには、デフォルトロールの作成コマンドを実行します。この場合--resource-typeを指定してsnapshot AWSDatalifeCycleManagerDefaultRole を作成するか、またはimageAWSDatalifeCycleManagerDefaultRoleForAMIManagementを作成します。

```
$ aws dlm create-default-role --resource-type snapshot|image
```

デフォルトのサービスロールを削除したのち、再度作成する必要がある場合は、同じ手順でアカウントにロールを再作成できます。

Amazon Data Lifecycle Manager のカスタムサービスロール

デフォルトのサービスロールを使用する代わりに、必要なアクセス許可を持つカスタム IAM ロールを作成し、ライフサイクルポリシーの作成時に、そのロールを選択することもできます。

カスタム IAM ロールを作成するには

1. 次のアクセス許可でロールを作成します。

   - スナップショットライフサイクルポリシーを管理するためのアクセス許可

     JSON

```
{
    "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",
                "ec2:DescribeAvailabilityZones"
            ],
            "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"
                }
            }
        }
```

```
    ]
}
```

- AMI ライフサイクルポリシーを管理するためのアクセス許可

  JSON

```
{
    "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/*"
        }
    ]
}
```

詳細については、IAM ユーザーガイド のロールの作成を参照してください。

2. ロールに信頼関係を追加します。

   a. IAM コンソールで、[ロール] を選択します。

   b. 作成したロールを選択し、信頼関係を選択します。

   c. [信頼関係の編集] を選択して、次のポリシーを追加し、[信頼ポリシーの更新] を選択します。

      JSON

```
{
 "Version":"2012-10-17",
 "Statement": [{
  "Effect": "Allow",
  "Principal": {
   "Service": "dlm.amazonaws.com"
  },
  "Action": "sts:AssumeRole"
 }]
}
```

      Confused Deputy Problem (混乱した使節の問題) から自分を守るために、aws:SourceAccount および aws:SourceArn の条件キーを使用することをお勧めします。例えば、前述の信頼ポリシーに次の条件ブロックを追加できます。aws:SourceAccount は、ライフサイクルポリシーの所有者であり、aws:SourceArn は、ライフサイクルポリシーの ARN です。ライフサイクルポリシー

ID が不明である場合は、ARN のその部分にワイルドカード (*) を選択することができ、ライフサイクルポリシーを作成後に信頼ポリシーを更新します。

```
"Condition": {
    "StringEquals": {
        "aws:SourceAccount": "account_id"
    },
    "ArnLike": {
        "aws:SourceArn": "arn:partition:dlm:region:account_id:policy/policy_id"
    }
}
```

Amazon Data Lifecycle Manager のポリシーをモニタリング

次の機能を使用して、スナップショットと AMI のライフサイクルをモニタリングできます。

機能

- コンソールと AWS CLI
- EventBridge を使用した Data Lifecycle Manager ポリシーのモニタリング
- CloudWatch を使用して、Data Lifecycle Manager のポリシーをモニタリング
- を使用した Amazon Data Lifecycle Manager API コールのログ記録 AWS CloudTrail

コンソールと AWS CLI

ライフサイクルポリシーは、Amazon EC2 コンソールまたは AWS CLIを使用して表示できます。ポリシーによって作成された各スナップショットと AMI には、タイムスタンプとポリシー関連のタグがあります。タグを使用してスナップショットと AMI をフィルタリングして、意図したとおりにバックアップが作成されていることを確認できます。

EventBridge を使用した Data Lifecycle Manager ポリシーのモニタリング

Amazon EBS と Amazon Data Lifecycle Manager は、ライフサイクルポリシーアクションに関するイベントを発行します。 AWS Lambda および Amazon CloudWatch Events を使用して、プログラムでイベント通知を処理できます。イベントは、ベストエフォートベースで出力されます。詳細については、Amazon EventBridge ユーザーガイドを参照してください。

利用できるイベントは次のとおりです。

> **Note**
>
> AMI ライフサイクルポリシーアクションでは、イベントは発生しません。

- `createSnapshot` – `CreateSnapshot` アクションが成功または失敗したときに発生する Amazon EBS イベント。詳細については、「Amazon EBS 用 Amazon EventBridge イベント」を参照してください。
- `DLM Policy State Change` – ライフサイクルポリシーがエラー状態になったときに発生する Amazon Data Lifecycle Manager イベント。このイベントには、エラーを引き起こした原因の説明が含まれています。

  次に、IAM ロールによって付与されたアクセス権限が不十分な場合のイベントの例を示します。

```
{
    "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"
    }
}
```

  制限を超えた場合のイベントの例を次に示します。

```
{
    "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 – 事前スクリプトまたは事後スクリプトが開始、成功、または失敗したときに発生するイベント。

  VSS バックアップが成功した場合のイベントの例を次に示します。

```
{
    "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": ""
    }
}
```

CloudWatch を使用して、Data Lifecycle Manager のポリシーをモニタリング

Amazon Data Lifecycle Manager のライフサイクルポリシーは、CloudWatch を使ってモニタリングすることができます。CloudWatch は、raw データを収集し、読み取り可能なほぼリアルタイムのメトリクスに加工することができます。これらのメトリクスを使用して、ポリシーによって作成、削除、コピーされた Amazon EBS スナップショットと EBS Backed AMI の数を正確に把握できます。また、特定のしきい値を監視するアラームを設定し、これらのしきい値に達したときに通知を送信したりアクションを実行したりできます。

メトリクスは 15 か月間保持されるため、履歴情報にアクセスして長期間にわたるライフサイクルポリシーのパフォーマンスをより的確に把握できます。

Amazon CloudWatch の詳細については、Amazon CloudWatch ユーザーガイドを参照してください。

トピック

- サポートされるメトリクス
- ポリシーの CloudWatch メトリクスを表示する
- ポリシーのグラフメトリクス
- ポリシーの CloudWatch アラームを作成する
- ユースケースの例
- 失敗したアクションを報告するポリシーの管理

サポートされるメトリクス

AWS/EBS 名前空間には、Amazon Data Lifecycle Manager の以下のメトリクスが含まれます。メトリクスは、ポリシータイプによって異なります。

すべてのメトリクスは、DLMPolicyId ディメンションで測定できます。最も有用な統計データは sum と average、そして測定単位は count です。

タブを選択すると、そのポリシータイプでサポートされているメトリクスが表示されます。

EBS snapshot policies

メトリクス	説明
Resources Targeted	スナップショットまたは EBS-backed AMI ポリシーで指定されたタグがターゲットとするリソースの数。
Snapshots CreateStarted	スナップショットポリシーによって開始されたスナップショット作成アクションの数。後続の再試行が複数ある場合でも、各アクションは 1 回だけ記録されます。 スナップショット作成アクションが失敗すると、Amazon Data Lifecycle Manager は SnapshotsCreateFailed メトリクスを送信します。
Snapshots CreateCompleted	スナップショットポリシーにより作成されたスナップショットの数。これには、スケジュールされた時刻から 60 分以内に成功した再試行が含まれます。
Snapshots CreateFailed	スナップショットポリシーにより作成できなかったスナップショットの数。これには、スケジュールされた時刻から 60 分以内に失敗した再試行が含まれます。
Snapshots SharedCompleted	スナップショットポリシーによってアカウント間で共有されたスナップショットの数。
Snapshots DeleteCompleted	スナップショットポリシーまたは EBS-backed AMI ポリシーにより削除されたスナップショットの数。このメトリクスは、ポリシーによって作成されたスナップショットにのみ適用されます。このポリシーは、ポリシーによって作成されたクロスリージョンスナップショットコピーには適用されません。

メトリクス	説明
	このメトリクスには、EBS-backed AMI ポリシーが AMI の登録を解除したときに削除されるスナップショットが含まれます。
SnapshotsDeleteFailed	スナップショットポリシーまたは EBS-backed AMI ポリシーにより削除できなかったスナップショットの数。このメトリクスは、ポリシーによって作成されたスナップショットにのみ適用されます。このポリシーは、ポリシーによって作成されたクロスリージョンスナップショットコピーには適用されません。 このメトリクスには、EBS-backed AMI ポリシーが AMI の登録を解除したときに削除されるスナップショットが含まれます。
SnapshotsCopiedRegionStarted	スナップショットポリシーによって開始されたクロスリージョンスナップショットコピーアクションの数。
SnapshotsCopiedRegionCompleted	スナップショットポリシーによって作成されたクロスリージョンスナップショットコピーの数。これには、スケジュールされた時刻から 24 時間以内に成功した再試行が含まれます。
SnapshotsCopiedRegionFailed	スナップショットポリシーにより作成できなかったクロスリージョンスナップショットコピーの数。これには、スケジュールされた時刻から 24 時間以内に失敗した再試行が含まれます。
SnapshotsCopiedRegionDeleteCompleted	スナップショットポリシーによって削除された、保持ルールで指定されたクロスリージョンスナップショットコピーの数。
SnapshotsCopiedRegionDeleteFailed	スナップショットポリシーにより、保持ルールで指定された削除できなかったクロスリージョンのスナップショットコピーの数。
snapshotsArchiveDeletionFailed	スナップショットポリシーによりアーカイブ層から削除できなかったアーカイブ済みスナップショットの数。

メトリクス	説明
snapshots ArchiveSc heduled	スナップショットポリシーによりアーカイブが予定されているスナップショットの数。
snapshots ArchiveCo mpleted	スナップショットポリシーによりアーカイブが正常に行われたスナップショットの数。
snapshots ArchiveFailed	スナップショットポリシーによりアーカイブできなかったスナップショットの数。
snapshots ArchiveDe letionCom pleted	スナップショットポリシーによりアーカイブ層から正常に削除できたアーカイブ済みスナップショットの数。
PreScript Started	事前スクリプトが正常に開始されたインスタンスの数。 スクリプトの再試行が有効になっている場合、このメトリクスはポリシー実行ごとに複数回出力される可能性があります。
PreScript Completed	事前スクリプトが正常に完了したインスタンスの数。事前スクリプトが指定したタイムアウト期間外に完了した場合でも、メトリクスは出力されます。 スクリプトの再試行が有効になっている場合、このメトリクスはポリシー実行ごとに複数回出力される可能性があります。
PreScript Failed	事前スクリプトが正常に完了しなかったインスタンスの数。事前スクリプトが指定したタイムアウト期間外に完了した場合でも、メトリクスは出力されます。 スクリプトの再試行が有効になっている場合、このメトリクスはポリシー実行ごとに複数回出力される可能性があります。

メトリクス	説明
PostScriptStarted	ポストスクリプトが正常に開始されたインスタンスの数。 スクリプトの再試行が有効になっている場合、このメトリクスはポリシー実行ごとに複数回出力される可能性があります。
PostScriptCompleted	ポストスクリプトが正常に完了したインスタンスの数。事後スクリプトが指定したタイムアウト期間外に完了した場合でも、メトリクスは出力されます。 スクリプトの再試行が有効になっている場合、このメトリクスはポリシー実行ごとに複数回出力される可能性があります。
PostScriptFailed	事後スクリプトが正常に完了しなかったインスタンスの数。事後スクリプトが指定したタイムアウト期間外に完了した場合でも、メトリクスは出力されます。 スクリプトの再試行が有効になっている場合、このメトリクスはポリシー実行ごとに複数回出力される可能性があります。
VSSBackupStarted	VSS バックアップが正常に開始されたインスタンスの数。 スクリプトの再試行が有効になっている場合、このメトリクスはポリシー実行ごとに複数回出力される可能性があります。
VSSBackupCompleted	VSS バックアップが正常に完了したインスタンスの数。VSS バックアップがタイムアウト期間外に完了した場合でも、メトリクスは出力されます。 スクリプトの再試行が有効になっている場合、このメトリクスはポリシー実行ごとに複数回出力される可能性があります。
VSSBackupFailed	VSS バックアップが正常に完了しなかったインスタンスの数。VSS バックアップがタイムアウト期間外に完了した場合でも、メトリクスは出力されます。 スクリプトの再試行が有効になっている場合、このメトリクスはポリシー実行ごとに複数回出力される可能性があります。

EBS-backed AMI policies

EBS-backed AMI ポリシーでは、次のメトリクスを使用できます。

メトリクス	説明
`Resources Targeted`	スナップショットまたは EBS-backed AMI ポリシーで指定されたタグがターゲットとするリソースの数。
`Snapshots DeleteCompleted`	スナップショットポリシーまたは EBS-backed AMI ポリシーにより削除されたスナップショットの数。このメトリクスは、ポリシーによって作成されたスナップショットにのみ適用されます。このポリシーは、ポリシーによって作成されたクロスリージョンスナップショットコピーには適用されません。 このメトリクスには、EBS-backed AMI ポリシーが AMI の登録を解除したときに削除されるスナップショットが含まれます。
`Snapshots DeleteFailed`	スナップショットポリシーまたは EBS-backed AMI ポリシーにより削除できなかったスナップショットの数。このメトリクスは、ポリシーによって作成されたスナップショットにのみ適用されます。このポリシーは、ポリシーによって作成されたクロスリージョンスナップショットコピーには適用されません。 このメトリクスには、EBS-backed AMI ポリシーが AMI の登録を解除したときに削除されるスナップショットが含まれます。
`Snapshots CopiedRegionDeleteCompleted`	スナップショットポリシーによって削除された、保持ルールで指定されたクロスリージョンスナップショットコピーの数。
`Snapshots CopiedRegionDeleteFailed`	スナップショットポリシーにより、保持ルールで指定された削除できなかったクロスリージョンのスナップショットコピーの数。

メトリクス	説明
ImagesCreateStarted	EBS-backed AMI ポリシーによって開始された CreateImage アクションの数。
ImagesCreateCompleted	EBS-backed AMI ポリシーによって作成された AMI の数。
ImagesCreateFailed	EBS-backed AMI ポリシーによって作成できなかった AMI の数。
ImagesDeregisterCompleted	EBS-backed AMI ポリシーによって登録解除された AMI の数。
ImagesDeregisterFailed	EBS-backed AMI ポリシーによって登録解除できなかった AMI の数。
ImagesCopiedRegionStarted	EBS-backed AMI ポリシーによって開始されたクロスリージョンコピーアクションの数。
ImagesCopiedRegionCompleted	EBS-Backed AMI ポリシーによって作成されたクロスリージョン AMI コピーの数。
ImagesCopiedRegionFailed	EBS-backed AMI ポリシーによって作成できなかったクロスリージョン AMI コピーの数。

メトリクス	説明
ImagesCopiedRegionDeregisterCompleted	EBS-backed AMI ポリシーによって保持ルールで指定された、登録解除されたクロスリージョン AMI コピーの数。
ImagesCopiedRegionDeregisteredFailed	EBS-backed AMI ポリシーによって登録解除できなかった、保持ルールで指定されたクロスリージョン AMI コピーの数。
EnableImageDeprecationCompleted	EBS-backed AMI ポリシーによって非推奨としてマークされた AMI の数。
EnableImageDeprecationFailed	EBS-backed AMI ポリシーによって非推奨としてマークされなかった AMI の数。
EnableCopiedImageDeprecationCompleted	EBS-Backed AMI ポリシーによって非推奨としてマークされたクロスリージョン AMI コピーの数。
EnableCopiedImageDeprecationFailed	グリッドでポリシーを選択し、Monitoringタブを選びます。EBSでサポートされているAMIポリシーによって非推奨としてマークされなかったクロスリージョンAMIコピーの数。

Cross-account copy event policies

クロスアカウントコピーイベントポリシーでは、以下のメトリクスを使用できます。

メトリクス	説明
SnapshotsCopiedAccountStarted	クロスアカウントコピーイベントポリシーによって開始された、クロスアカウントスナップショットコピーアクションの数。
SnapshotsCopiedAccountCompleted	クロスアカウントコピーイベントポリシーによって別のアカウントからコピーされたスナップショットの数。これには、スケジュールされた時刻から 24 時間以内に成功した再試行が含まれます。
SnapshotsCopiedAccountFailed	クロスアカウントコピーイベントポリシーによって別のアカウントからコピーできなかったスナップショットの数。これには、スケジュールされた時刻から 24 時間以内に失敗した再試行が含まれます。
SnapshotsCopiedAccountDeleteCompleted	クロスアカウントのコピーイベントポリシーによって、保持ルールで指定された削除されたクロスレジオンスナップショットコピーの数。
SnapshotsCopiedAccountDeleteFailed	クロスアカウントのコピーイベントポリシーによって、保持ルールで指定された削除できなかったクロスレジオンスナップショットコピーの数。

ポリシーの CloudWatch メトリクスを表示する

AWS マネジメントコンソール または コマンドラインツールを使用して、Amazon Data Lifecycle Manager が Amazon CloudWatch に送信するメトリクスを一覧表示できます。

Amazon EC2 console

Amazon EC2 コンソールを使用してメトリクスを表示するには

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。

2. ナビゲーションペインでLifecycle Managerを選択します。
3. グリッドでポリシーを選択し、モニタリングタブを選びます。

CloudWatch console

Amazon CloudWatch コンソールを使用してメトリクスを表示するには

1. CloudWatch コンソールの https://console.aws.amazon.com/cloudwatch/ を開いてください。
2. ナビゲーションペインで [メトリクス] を選択します。
3. [EBS] 名前空間を選択し、[Data Lifecycle Manager metrics] (Data Lifecycle Manager メトリクス) を選択します。

AWS CLI

Amazon Data Lifecycle Manager で利用可能なメトリクスをすべて表示するには

list-metrics コマンドを使用します。

```
$ C:\> aws cloudwatch list-metrics \
    --namespace AWS/EBS
```

特定のポリシーのすべてのメトリクスを表示するには

list-metrics コマンドを使用して、DLMPolicyId ディメンションを指定します。

```
$ C:\> aws cloudwatch list-metrics \
    --namespace AWS/EBS \
    --dimensions Name=DLMPolicyId,Value=policy-abcdef01234567890
```

すべてのポリシーにわたって単一のメトリクスを表示するには

list-metrics コマンドを使用して、--metric-name オプションを指定します。

```
$ C:\> aws cloudwatch list-metrics \
    --namespace AWS/EBS \
    --metric-name SnapshotsCreateCompleted
```

ポリシーのグラフメトリクス

ポリシーの作成が完了したら、Amazon EC2 コンソールを開いて、Monitoring タブにポリシーのモニタリンググラフを表示できます。各グラフは利用可能な Amazon EC2 メトリクスのいずれかに基づいています。

以下のグラフが利用可能です：

- ターゲットとなるリソース (ResourcesTargetedに基づく)
- スナップショットの作成が開始されました (SnapshotsCreateStarted)
- スナップショットの作成が完了しました (SnapshotsCreateCompletedに基づく)
- スナップショットの作成に失敗しました (SnapshotsCreateFailedに基づく)
- スナップショットの共有が完了しました (SnapshotsSharedCompletedに基づく)
- スナップショットの削除が完了しました (SnapshotsDeleteCompletedに基づく)
- スナップショットの削除に失敗しました (SnapshotsDeleteFailedに基づく)
- スナップショットのクロスリージョンコピーが開始されました (SnapshotsCopiedRegionStartedに基づく)
- スナップショットのクロスリージョンコピーが完了しました (SnapshotsCopiedRegionCompletedに基づく)
- スナップショットのクロスリージョンコピーに失敗しました (SnapshotsCopiedRegionFailedに基づく)
- スナップショットのクロスリージョンコピーの削除が完了しました (SnapshotsCopiedRegionDeleteCompletedに基づく)
- スナップショットのクロスリージョンコピーの削除に失敗しました (SnapshotsCopiedRegionDeleteFailedに基づく)
- スナップショットのクロスアカウントコピーが開始されました (SnapshotsCopiedAccountStartedに基づく)
- スナップショットのクロスアカウントコピーが完了しました (SnapshotsCopiedAccountCompletedに基づく)
- スナップショットのクロスアカウントコピーに失敗しました (SnapshotsCopiedAccountFailedに基づく)
- スナップショットのクロスアカウントコピーの削除が完了しました (SnapshotsCopiedAccountDeleteCompletedに基づく)

- スナップショットのクロスアカウントコピーの削除に失敗しました (SnapshotsCopiedAccountDeleteFailedに基づく)
- AMI の作成が開始されました (ImagesCreateStartedに基づく)
- AMI の作成が完了しました (ImagesCreateCompletedに基づく)
- AMI の作成に失敗しました (ImagesCreateFailedに基づく)
- AMI の登録解除が完了しました (ImagesDeregisterCompletedに基づく)
- AMI の登録解除に失敗しました (ImagesDeregisterFailedに基づく)
- AMI のクロスリージョンコピーを開始 (ImagesCopiedRegionStartedに基づく)
- AMI のクロスリージョンコピーが完了しました (ImagesCopiedRegionCompletedに基づく)
- AMI のクロスリージョンコピーに失敗しました (ImagesCopiedRegionFailedに基づく)
- AMI のクロスリージョンコピーの登録解除が完了しました (ImagesCopiedRegionDeregisterCompletedに基づく)
- AMI のクロスリージョンコピーの登録解除に失敗しました (ImagesCopiedRegionDeregisteredFailedに基づく)
- AMI の有効化の廃止が完了しました (EnableImageDeprecationCompletedに基づく)
- AMI の有効化の廃止に失敗しました (EnableImageDeprecationFailedに基づく)
- AMI クロスリージョンコピーの有効化の廃止が完了しました (EnableCopiedImageDeprecationCompletedに基づく)
- AMI クロスリージョンコピーの有効化の廃止に失敗しました (EnableCopiedImageDeprecationFailedに基づく)

ポリシーの CloudWatch アラームを作成する

ポリシーの CloudWatch メトリクスをモニタリングする CloudWatch アラームを作成できます。CloudWatch は、指定したしきい値にメトリクスが到達すると、自動的に通知を送信します。CloudWatch アラームを作成するには、CloudWatch コンソールを使用します。

CloudWatch コンソールを使用してアラームを作成する方法について詳細は、Amazon CloudWatch ユーザーガイドの次のトピックを参照してください。

- 静的しきい値に基づいて CloudWatch アラームを作成する
- 異常検出に基づいて CloudWatch アラームを作成する

ユースケースの例

ユースケースの例を次に示します。

トピック

- 例 1: ResourcesTargeted メトリクス
- 例 2: SnapshotDeleteFailed メトリクス
- 例 3: SnapshotsCopiedRegionFailed メトリクス

例 1: ResourcesTargeted メトリクス

ResourcesTargeted メトリクスを使って、特定のポリシーが実行されるたびに対象となるリソースの総数をモニタリングすることができます。これにより、ターゲットリソースの数が予想されるしきい値を下回ったり上回ったりしたときにアラームをトリガーできます。

例えば、日時ポリシーで 50 個以下のボリュームのバックアップを作成することを想定している場合、1 時間の間に ResourcesTargeted の sum が 50 より大きくなったときにメールで通知するアラームを作成することができます。これにより、誤ってタグ付けされたボリュームからスナップショットが予期せず作成されていないことを確認できます。

以下のコマンドを使用して、このアラームを作成できます。

```
$ 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
```

例 2: SnapshotDeleteFailed メトリクス

この SnapshotDeleteFailed メトリクスを使用して、ポリシーのスナップショット保持ルールに従ってスナップショットを削除する際の失敗をモニタリングできます。

例えば、12 時間ごとにスナップショットを自動的に削除するポリシーを作成した場合、1 時間の間に SnapshotDeletionFailed の sum が 0 より大きくなったときにエンジニアリングチームに通知するアラームを作成することができます。これにより、不適切なスナップショット保持を調査し、不要なスナップショットによってストレージコストが増加しないようにすることができます。

以下のコマンドを使用して、このアラームを作成できます。

```
$ 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
```

例 3: SnapshotsCopiedRegionFailed メトリクス

SnapshotsCopiedRegionFailed メトリクスを使用して、ポリシーが他のリージョンへのスナップショットのコピーに失敗した場合を特定します。

例えば、ポリシーによりリージョン間でスナップショットを毎日コピーしている場合、1 時間の間に SnapshotCrossRegionCopyFailed の sum が 0 より大きくなったときにエンジニアリングチームに SMS を送信するアラームを作成することができます。これは、系統内の後続のスナップショットがポリシーによって正常にコピーされたかどうかを検証する場合に便利です。

以下のコマンドを使用して、このアラームを作成できます。

```
$ 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
```

失敗したアクションを報告するポリシーの管理

ポリシーの 1 つが、失敗したアクションメトリクスについて予期しないゼロ以外の値を報告した場合の対処方法の詳細については、「What do I do if Amazon Data Lifecycle Manager reports failed actions in Amazon CloudWatch metrics?」の記事を参照してください。

を使用した Amazon Data Lifecycle Manager API コールのログ記録 AWS CloudTrail

を使用すると AWS CloudTrail、ユーザーアクティビティと API の使用状況を追跡して、内部ポリシーと規制標準への準拠を示すことができます。詳細については、「AWS CloudTrail ユーザーガイド」を参照してください。

Amazon Data Lifecycle Manager (Amazon Data Lifecycle Manager) は AWS CloudTrail、Amazon Data Lifecycle Manager のユーザー、ロール、または のサービスによって実行されたアクションを記録する AWS サービスである と統合されています。CloudTrail は、Amazon Data Lifecycle Manager コンソールからの呼び出しや Amazon Data Lifecycle Manager API へのコード呼び出しを含む、Amazon Data Lifecycle Manager のすべての APIs。証跡を作成する場合は、Amazon S3バケットへの CloudTrail イベントの継続的な配信を有効にすることができます。証跡を設定しない場合でも、CloudTrail コンソールの [イベント履歴] で最新のイベントを表示できます。CloudTrail で収集された情報を使用して、Amazon Data Lifecycle Manager に対するリクエスト、リクエスト元の IP アドレス、リクエスト者、リクエスト日時などの詳細を確認できます。

CloudTrail の詳細については、「AWS CloudTrail ユーザーガイド」を参照してください。

CloudTrail の Amazon Data Lifecycle Manager 情報

CloudTrail は、 AWS アカウントの作成時にアカウントで有効になります。Amazon Data Lifecycle Manager でアクティビティが発生すると、そのアクティビティはイベント履歴の他の AWS サービスイベントとともに CloudTrail イベントに記録されます。 AWS アカウントで最近のイベントを表示、検索、ダウンロードできます。詳細については、CloudTrail イベント履歴でのイベントの表示を参照してください。

Amazon Data Lifecycle Manager のイベントなど、 AWS アカウントのイベントの継続的な記録については、証跡を作成します。証跡により、ログファイルを CloudTrail で Amazon S3 バケットに配信できます。デフォルトでは、コンソールで追跡を作成するときに、追跡がすべてのリージョンに適用されます。証跡は、 AWS パーティション内のすべてのリージョンからのイベントをログに記録し、

指定した Amazon S3 バケットにログファイルを配信します。さらに、CloudTrail ログで収集されたイベントデータをさらに分析して処理するように他の AWS サービスを設定できます。詳細については、以下を参照してください。

- 証跡を作成するための概要
- CloudTrail がサポートするサービスと統合
- CloudTrail 用 Amazon SNS 通知の構成
- 複数のリージョンから CloudTrail ログファイルを受け取るおよび複数のアカウントから CloudTrail ログファイルを受け取る

すべての Amazon Data Lifecycle Manager アクションは CloudTrail によってログに記録されます。例えば、`CreateLifecyclePolicy` および `DeleteLifecyclePolicy` の各アクションを呼び出すと、CloudTrail ログファイルにエントリが生成されます。アクションの完全なリストについては、「アクション」を参照してください。

各イベントまたはログエントリには、誰がリクエストを生成したかという情報が含まれます。アイデンティティ情報は、以下を判別するのに役立ちます。

- リクエストを行ったユーザー。
- リクエストがロールまたはフェデレーションユーザーのテンポラリなセキュリティ認証情報を使用して行われたかどうか。
- リクエストが別の AWS サービスによって行われたかどうか。

詳細については、「CloudTrail userIdentity 要素」を参照してください。

Amazon Data Lifecycle Manager ログファイルエントリについて

「トレイル」は、指定した Amazon S3 バケットにイベントをログファイルとして配信するように設定できます。CloudTrail のログファイルは、単一か複数のログエントリを含みます。イベントはあらゆるソースからの単一のリクエストを表し、リクエストされたアクション、アクションの日時、リクエストのパラメータなどの情報が含まれます。CloudTrail ログファイルは、公開 API コールの順序付けられたスタックトレースではないため、特定の順序では表示されません。

Example例: CreateLifecyclePolicy

`CreateLifecyclePolicy` アクションの CloudTrail ログエントリの例は、以下のとおりです。

```
{
```

```
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "Root",
        "principalId": "123456789012",
        "arn": "arn:aws:iam::123456789012:root",
        "accountId": "123456789012",
        "accessKeyId": "AKIAJA2ELRVCPEXAMPLE",
        "userName": "user",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2018-07-24T18:01:05Z"
            }
        }
    },
    "eventTime": "2018-07-24T18:20:28Z",
    "eventSource": "dlm.amazonaws.com",
    "eventName": "CreateLifecyclePolicy",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "54.240.230.179",
    "userAgent": "console.ec2.amazonaws.com",
    "requestParameters": {
        "ExecutionRoleArn": "arn:aws:iam::123456789012:role/service-role/
AWSDataLifecycleManagerServiceRole",
        "PolicyDetails": {
            "ResourceTypes": [
                "VOLUME"
            ],
            "Schedules": [
                {
                    "CreateRule": {
                        "Interval": 12,
                        "IntervalUnit": "HOURS",
                        "Times": [
                            "09:00"
                        ]
                    },
                    "Name": "Default Schedule",
                    "RetainRule": {
                        "Count": 3
                    },
                    "TagsToAdd": [
                        {
                            "Key": "Name",
```

```
                            "Value": "backup-my-volume"
                        }
                    ]
                }
            ],
            "TargetTags": [
                {
                    "Key": "Name",
                    "Value": "my-volume"
                }
            ]
        },
        "Description": "test-cloudtrail",
        "State": "DISABLED"
    },
    "responseElements": {
        "PolicyId": "policy-04ff8755fce0599eb"
    },
    "requestID": "3d714ca6-8f6e-11e8-92a4-35fd765427f0",
    "eventID": "28ab3121-6040-4a40-80c7-ae59b3adf405",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789012"
}
```

Example例: DeleteLifecyclePolicy

`DeleteLifecyclePolicy` アクションの CloudTrail ログエントリの例は、以下のとおりです。

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "Root",
        "principalId": "123456789012",
        "arn": "arn:aws:iam::123456789012:root",
        "accountId": "123456789012",
        "accessKeyId": "AKIAJA2ELRVCPEXAMPLE",
        "userName": "user",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2018-07-24T18:01:05Z"
            }
        }
```

```
    },
    "eventTime": "2018-07-24T19:33:33Z",
    "eventSource": "dlm.amazonaws.com",
    "eventName": "DeleteLifecyclePolicy",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "54.240.230.241",
    "userAgent": "console.ec2.amazonaws.com",
    "requestParameters": {
        "policyId": "policy-04ff8755fce0599eb"
    },
    "responseElements": null,
    "requestID": "73260971-8f78-11e8-a156-598016e53fb2",
    "eventID": "3740f2fb-0d6a-4712-a7ad-eb9f17103fb2",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789012"
}
```


Amazon Data Lifecycle Manager 用のサービスエンドポイント

エンドポイントは、 AWS ウェブサービスのエントリポイントとして機能する URL です。Amazon Data Lifecycle Manager は、次のエンドポイントタイプをサポートします。

- IPv4 エンドポイント
- IPv4 と IPv6 の両方をサポートするデュアルスタックのエンドポイント
- FIPS エンドポイント

リクエストを行うと、使用するエンドポイントとリージョンを指定できます。エンドポイントを指定しない場合、デフォルトで IPv4 エンドポイントが使用されます。別のエンドポイントタイプを使用するには、リクエストで指定する必要があります。これを行う方法の例については、「エンドポイントの指定」を参照してください。

Amazon Data Lifecycle Manager については、「Amazon Web Services 全般のリファレンス」の「Amazon Data Lifecycle Manager endpoints and quotas」を参照してください。

トピック

- IPv4 エンドポイント
- デュアルスタック (IPv4 および IPv6) エンドポイント
- FIPS エンドポイント

- エンドポイントの指定

IPv4 エンドポイント

IPv4 エンドポイントは IPv4 トラフィックのみをサポートします。IPv4 エンドポイントは、全リージョンで利用できます。

リージョンをエンドポイント名の一部として指定する必要があります。エンドポイント名には、次の命名規則が使用されます。

- dlm.*region*.amazonaws.com

例えば、米国東部 (バージニア北部) リージョンの IPv4 エンドポイントは `dlm.us-east-1.amazonaws.com` です。

デュアルスタック (IPv4 および IPv6) エンドポイント

デュアルスタックエンドポイントは、IPv4 と IPv6 トラフィックの両方をサポートします。デュアルスタックエンドポイントは、すべてのリージョンで利用できます。

IPv6 を使用するには、デュアルスタックエンドポイントを使用する必要があります。デュアルスタックエンドポイントにリクエストを行うと、エンドポイント URL は、ネットワークとクライアントが使用するプロトコルに応じて IPv6 または IPv4 アドレスに解決されます。

リージョンをエンドポイント名の一部として指定する必要があります。デュアルスタックエンドポイント名には、次の命名規則が使用されます。

- `dlm.`*region*`.api.aws`

例えば、米国東部 (バージニア北部) リージョンのデュアルスタックエンドポイントは `dlm.us-east-1.api.aws` です。

FIPS エンドポイント

Amazon Data Lifecycle Manager は、次のリージョンの FIPS 検証済みデュアルスタック (IPv4 および IPv6) エンドポイントを提供します。

- `us-east-1` — 米国東部 (バージニア北部)

- us-east-2 — 米国東部 (オハイオ)
- us-west-1 — 米国西部 (北カリフォルニア)
- us-west-2 — 米国西部 (オレゴン)
- ca-central-1 — カナダ (中部)
- ca-west-1 — カナダ西部 (カルガリー)

FIPS デュアルスタックエンドポイントは、命名規則 を使用しますdlm-fips.*region*.api.aws。例えば、米国東部 (バージニア北部) リージョンの FIPS デュアルスタックのエンドポイントは dlm-fips.us-east-1.api.aws です。

エンドポイントの指定

次の例は、 AWS CLIを使用して US East (N. Virginia) リージョンのエンドポイントを指定する方法を示しています。

- デュアルスタック

```
aws dlm create-default-role \
--resource-type snapshot \
--endpoint-url https://dlm.us-east-2.api.aws
```

- IPv4

```
aws dlm create-default-role \
--resource-type snapshot \
--endpoint-url https://dlm.us-east-2.amazonaws.com
```

VPC と Amazon EBS 間にプライベート接続を作成する

AWS PrivateLink を利用した インターフェイス VPC エンドポイントを作成することで、VPC と Amazon EBS 間にプライベート接続を確立できます。インターネットゲートウェイ、NAT デバイス、VPN 接続、または AWS Direct Connect 接続を使用せずに、VPC 内にあるかのように Amazon EBS にアクセスできます。VPC のインスタンスは、パブリック IP アドレスがなくても Amazon EBS と通信できます。

インターフェイスエンドポイントに対して有効にする各サブネットにエンドポイントネットワークインターフェイスを作成します。

詳細については、「 AWS PrivateLink ガイド」の「Access AWS のサービス through AWS PrivateLink」を参照してください。

> **Note**
>
> Amazon Data Lifecycle Manager は、すべての商用および AWS GovCloud (US) リージョンの IPv4 インターフェイス VPC エンドポイントと、商用リージョンの IPv6 インターフェイス VPC エンドポイントのみをサポートします。

Amazon EBS の VPC エンドポイントに関する考慮事項

Amazon EBS のインターフェイス VPC エンドポイントを設定する前に、「AWS PrivateLink ガイド」の「Access an AWS service using an interface VPC endpoint」を確認してください。

デフォルトで、エンドポイント経由での Amazon EBS への完全なアクセスが許可されます。VPC エンドポイントポリシーを使用してインターフェイスエンドポイントへのアクセスを制御できます。VPC エンドポイントに Amazon EBS へのアクセスをコントロールするエンドポイントポリシーをアタッチできます。このポリシーでは、以下の情報を指定します。

- アクションを実行できるプリンシパル。
- 実行可能なアクション。
- アクションを実行できるリソース。

詳細については、Amazon VPC ユーザーガイドのVPC エンドポイントによるサービスのアクセスコントロールを参照してください。

Amazon EBS のエンドポイントポリシーの例を次に示します。このポリシーは、エンドポイントにアタッチされると、Amazon Data Lifecycle Manager ポリシーに関する概要情報を取得するアクセス許可をすべてのユーザーに付与します。

```
{
  "Statement": [{
    "Action": "dlm:GetLifecyclePolicies",
    "Effect": "Allow",
    "Principal": "*",
    "Resource": "*"
  }]
```

```
}
```

Amazon EBS 用のインターフェイス VPC エンドポイントを作成する

Amazon EBS 用の VPC エンドポイントは、Amazon VPC コンソールまたは AWS Command Line Interface (AWS CLI) で作成できます。詳細については、「AWS PrivateLink ガイド」の「Create a VPC endpoint」を参照してください。

Amazon EBS 用の VPC エンドポイントを作成するには、次のサービス名を使用します。

- `com.amazonaws.region.dlm`

エンドポイントのプライベート DNS を有効にすると、リージョンのデフォルト DNS 名 (`dlm.us-east-1.amazonaws.com` など) を使用して、Amazon EBS への API リクエストを実行できます。

Amazon Data Lifecycle Manager の問題のトラブルシューティング

以下のドキュメントは、発生する可能性のある問題のトラブルシューティングに役立ちます。

トピック

- エラー: Role with name already exists

エラー: `Role with name already exists`

説明

コンソールを使用してポリシーを作成しようとしたときに、`Role with name AWSDataLifecycleManagerDefaultRole already exists` または `Role with name AWSDataLifecycleManagerDefaultRoleForAMIManagement already exists` エラーが発生します。

原因

デフォルトロールの ARN 形式は、コンソールまたは AWS CLIのどちらを使用して作成されたかによって異なります。ARN は異なりますが、ロールは同じロール名を使用するため、コンソールと AWS CLIでロール名の競合が発生します。

ソリューション

この問題を解決するには、次の操作を行います:

1. (事前スクリプトと事後スクリプトに対してのみ有効なスナップショットポリシーの場合) AWSDataLifecycleManagerSSMFullAccess AWS 管理ポリシーを AWSDataLifecycleManagerDefaultRole IAM ロールに手動でアタッチします。詳細については、「IAM アイデンティティのアクセス許可の追加」を参照してください。
2. Amazon Data Lifecycle Manager ポリシーを作成するときは、[IAM ロール] で [別のロールを選択] を選択し、[AWSDataLifecycleManagerDefaultRole] (スナップショットポリシー用) または [AWSDataLifecycleManagerDefaultRoleForAMIManagement] (AMI ポリシー用) のいずれかを選択します。
3. 引き続き、通常どおりポリシーを作成します。

AWS Backup for Amazon Elastic Block Store

AWS Backup は、 AWS 環境全体でデータ保護を一元化および自動化するフルマネージドサービスです。リソースごとにスナップショットスケジュールと保持を管理するのではなく、単一のバックアップポリシーを定義し、アカウントとリージョン全体に適用します。リソースにタグを付け、チームごとの設定を必要とせずに自動的にリソース AWS Backup を保護します。このサービスは、Amazon EBS、Amazon EC2、Amazon S3、Amazon EFS、Amazon RDS、Aurora など、多くの AWS サービスをサポートしています。その結果、1 つのポリシーで、ワークロードごとに個別のバックアップツールを使用せずに、ボリューム、データベース、ファイルシステム、オブジェクトストアをまとめて保護できます。

Amazon EBS の場合、 は Amazon EC2 インスタンスにアタッチされたすべてのボリュームのpoint-in-timeバックアップを調整するクラッシュコンシステントなマルチボリュームスナップショット AWS Backup を提供します。これは、複数のボリュームにまたがるデータベースやクラスター化されたアプリケーションに不可欠です。 AWS Backup 検索では、項目レベルの復旧を有効にして、復旧をさらに進めます。ボリューム全体を復元することなく、Amazon EBS スナップショットから個々のファイルまたはディレクトリを見つけて復元できます。

AWS Backup は、スナップショットにエンタープライズグレードのセキュリティとコンプライアンスも追加します。論理エアギャップボールトは、本番環境から復旧ポイントを分離します。ボールトロックはイミュータビリティを適用します。Amazon GuardDuty 統合は、マルウェアのバックアップをスキャンします。Backup Audit Manager は、バックアップ体制を継続的に評価し、監査対応コンプライアンスレポートを生成します。

開始するには、次のリソースを参照してください。

- AWS Backup デベロッパーガイド
- マルチボリュームのクラッシュコンシステントバックアップを作成する
- Amazon EBS ボリュームの復元
- Amazon EBS スナップショットから個々のファイルを検索して復元する

EBS direct API を使用して EBS スナップショットの内容にアクセスする

Amazon Elastic Block Store (Amazon EBS) direct API を使用して、EBS スナップショットの作成、スナップショットへのデータの直接書き込み、スナップショットのデータの読み取り、2 つのスナップショット間の違いや変更の特定を行うことができます。Amazon EBS のバックアップサービスを提供する独立系ソフトウェアベンダー (ISV) の場合は、EBS direct API を使用すると、スナップショットを介して EBS ボリュームの増分変更の追跡を効率化し、費用対効果を高めることができます。これを行うために、スナップショットから新しいボリュームを作成したり、Amazon Elastic Compute Cloud (Amazon EC2) インスタンスを使用して違いを比較したりする必要はありません。

増分スナップショットは、オンプレミスのデータから EBS ボリュームやクラウド内に直接作成し、迅速な災害対策に使用できます。災害が発生した場合は、スナップショットの読み書き機能を使用して、オンプレミスのデータを EBS スナップショットに書き込むことができます。その後、復旧後、スナップショットから AWS またはオンプレミスに復元できます。Amazon EBS との間でデータをコピーする複雑なメカニズムを構築して維持する必要はなくなりました。

このユーザーガイドでは、EBS direct API を構成する要素に関する概要と、これらの要素を効果的に使用する方法の例を示します。API のアクション、データ型、パラメータ、エラーの詳細については、EBS direct API リファレンスを参照してください。EBS direct API でサポートされている AWS リージョン、エンドポイント、サービスクォータの詳細については、の「Amazon EBS エンドポイントとクォータ」を参照してくださいAWS 全般のリファレンス。 APIs

トピック

- EBS direct API の料金
- EBS direct API の概念
- IAM を使用して EBS direct API へのアクセスを制御
- EBS direct API を使用して Amazon EBS スナップショットを読み取る
- EBS direct API を使用して Amazon EBS スナップショットへ書き込む
- EBS direct API の暗号化の結果
- EBS direct API チェックサムを使用してスナップショットデータを検証
- StartSnapshot API リクエストのべき等性を確保
- EBS direct API のエラー再試行
- EBS direct API のパフォーマンスを最適化

- EBS direct API 用のサービスエンドポイント
- AWS EBS direct APIsの SDK コード例
- VPC と EBS direct API 間にプライベート接続を作成
- を使用して EBS direct APIsログに記録する AWS CloudTrail
- EBS direct API に関するよくある質問

EBS direct API の料金

API の料金

EBS direct API の使用料金は、リクエストに応じて異なります。詳細については、Amazon EBS の料金表を参照してください。

- ListChangedBlocks および ListSnapshotBlocks API ではリクエストごとに課金されます。例えば、1,000 リクエストあたり 0.0006 USD を請求するリージョンで 100,000 の ListSnapshotBlocks API リクエストを行うと、0.06 USD (1,000 リクエストあたり 0.0006 USD × 100) が課金されます。
- GetSnapshotBlock では、返されたブロックごとに課金されます。例えば、返された 1,000 ブロックあたり 0.003 USD を請求するリージョンで 100,000 の GetSnapshotBlock API リクエストを行うと、0.30 USD (1,000 ブロックあたり 0.003 USD × 100) が課金されます。
- PutSnapshotBlock では、書き込まれたブロックごとに課金されます。例えば、書き込まれた 1,000 ブロックあたり 0.006 USD を請求するリージョンで 100,000 の PutSnapshotBlock API リクエストを行うと、0.60 USD (書き込まれた 1,000 ブロックあたり 0.006 USD × 100) が課金されます。

ネットワークコスト

データ転送コスト

EBS direct APIsと同じ AWS リージョンの Amazon EC2 インスタンス間で直接転送されるデータは、FIPS 以外のエンドポイントを使用する場合に無料です。詳細については、AWS サービスエンドポイントを参照してください。他のサービス AWS がデータ転送の途中にある場合は、関連するデータ処理コストが請求されます。これらのサービスには、PrivateLink エンドポイント、NAT ゲートウェイ、および Transit Gateway が含まれますが、これらに限定されません。

VPC インターフェイスのエンドポイント

プライベートサブネットの Amazon EC2 インスタンスまたは AWS Lambda 関数から EBS direct APIs を使用している場合は、NAT ゲートウェイを使用する代わりに VPC インターフェイスエンドポイントを使用して、ネットワークデータ転送コストを削減できます。詳細については、「VPC と EBS direct API 間にプライベート接続を作成」を参照してください。

EBS direct API の概念

EBS direct API の使用を開始する前に、以下の主要な概念を理解しておく必要があります。

スナップショット

スナップショットは、EBS ボリュームからデータをバックアップするための主な手段です。EBS direct API では、オンプレミスのディスクからスナップショットにデータをバックアップすることもできます。ストレージコストを節約するために、連続するスナップショットは増分で、以前のスナップショット以降に変更されたボリュームデータのみが含まれています。詳細については、「Amazon EBS スナップショット」を参照してください。

> **Note**
>
> EBS direct APIsは、 でのパブリックスナップショットとローカルスナップショットをサポートしていません AWS Outposts。

ブロック

ブロックは、スナップショット内のデータのフラグメントです。各スナップショットには、何千ものブロックを含めることができます。スナップショット内のすべてのブロックは固定サイズです。

ブロックインデックス

ブロックインデックスは、512 KiB ブロック単位の論理インデックスです。ブロックインデックスを識別するには、論理ボリューム内のデータの論理オフセットをブロックサイズで除算します (データの論理オフセット/524288)。データの論理オフセットは 512 KiB に整合させる必要があります。

ブロックトークン

ブロックトークンは、スナップショット内のブロックの識別ハッシュであり、ブロックデータの検索に使用されます。EBS direct API から返されるブロックトークンは一時的なものです。ブロックトー

クンは、これらのトークンに指定された有効期限のタイムスタンプに応じて変更されるか、同じスナップショットに対して別の ListSnapshotBlocks リクエストや ListChangedBlocks リクエストを実行した場合に変更されます。

チェックサム

チェックサムは、送信中や保存中に発生したエラーを検出するために、データのブロックから派生される小さいサイズのデータです。EBS direct API は、チェックサムを使用してデータの整合性を検証します。EBS スナップショットからデータを読み取るときに、送信されるデータブロックごとに Base64 でエンコードされた SHA256 チェックサムがサービスから提供されます。このチェックサムを使用してデータを検証できます。EBS スナップショットにデータを書き込むときは、送信するデータのブロックごとに Base64 でエンコードした SHA256 チェックサムを提供する必要があります。サービスは、提供されたチェックサムを使用して、受信したデータを検証します。詳細については、このガイドで後述するEBS direct API チェックサムを使用してスナップショットデータを検証を参照してください。

暗号化

暗号化は、データを読み取り不可能なコードに変換することで、データを保護します。このコードは、暗号化に使用された KMS キー にアクセスできるユーザーのみが解読できます。暗号化されたスナップショットは、EBS direct API を使用して読み書きできますが、いくつかの制限があります。詳細については、このガイドで後述するEBS direct API の暗号化の結果を参照してください。

API アクション

EBS direct API は、6 つのアクションで構成されています。3 つは読み取りアクションであり、他の 3 つは書き込みアクションです。読み取りアクションは以下のとおりです。

- ListSnapshotBlocks – 指定されたスナップショット内のブロックから、ブロックインデックスとブロックトークンを返します。
- ListChangedBlocks – 同じボリュームとスナップショット系列の 2 つの指定されたスナップショットのブロック間で異なる、ブロックインデックスとブロックトークンを返します。
- GetSnapshotBlock – 指定されたスナップショット ID、ブロックインデックス、ブロックトークンに対応するブロックのデータを返します。

書き込みアクションは以下のとおりです。

- StartSnapshot – 既存のスナップショットの増分スナップショット、あるいは新しいスナップショットとしてスナップショットを開始します。開始済みスナップショットは、CompleteSnapshot アクションを使用して完了するまで、保留状態となります。
- PutSnapshotBlock – 開始済みスナップショットに、個別のブロックとしてデータを追加します。データのブロックを送信する際に、Base64 でエンコードした SHA256 チェックサムを指定する必要があります。送信が完了すると、このチェックサムがサービスによって検証されます。サービスが計算したチェックサムと指定したチェックサムが一致しない場合、リクエストは失敗します。
- CompleteSnapshot – 保留状態にある開始済みスナップショットを完了します。これにより、スナップショットは完了状態に変更されます。

署名バージョン 4 の署名

署名バージョン 4 は、HTTP によって送信された AWS リクエストに認証情報を追加するプロセスです。セキュリティのため、 へのほとんどのリクエストは、アクセスキー ID とシークレットアクセスキーで構成されるアクセスキーで署名 AWS する必要があります。これらの 2 つのキーは、一般的にセキュリティ認証情報と呼ばれます。アカウントの認証情報を取得する方法については、「AWS セキュリティ認証情報」を参照してください。

HTTP リクエストを手動で作成する場合は、リクエストへの署名方法を知っている必要があります。AWS Command Line Interface (AWS CLI) またはいずれかの AWS SDKs を使用してリクエストを行うと AWS、これらのツールは、ツールの設定時に指定したアクセスキーを使用して自動的にリクエストに署名します。これらのツールを使う場合は、自分でリクエストに署名する方法を学ぶ必要はありません。

詳細については、IAM ユーザーガイドの AWS API リクエストの署名を参照してください。

IAM を使用して EBS direct API へのアクセスを制御

ユーザーが EBS direct API を使用するためには、次のポリシーが必要です。詳細については、「ユーザー許可の変更」を参照してください。

EBS direct API リソース、アクション、条件コンテキストキーの IAM 許可ポリシーでの使用については、「サービス認証リファレンス」の「Amazon Elastic Block Store 用のアクション、リソース、条件キー」を参照してください。

> **Important**
>
> 以下のポリシーを ユーザーに割り当てる際には注意が必要です。これらのポリシーを割り当てることで、Amazon EC2 API (CopySnapshot アクションや CreateVolume アクションなど) を介して、同じリソースへのアクセスが拒否されているユーザーにアクセスが許可される場合があります。

スナップショットを読み取るためのアクセス許可

次のポリシーでは、読み取り EBS direct APIsを特定の AWS リージョン内のすべてのスナップショットで使用できます。このポリシーで、*<Region>* はスナップショットのリージョンに置き換えます。

次のポリシーでは、特定のキーと値のタグを持つスナップショットに対する読み取り EBS direct API の使用を許可します。このポリシーで、*<Key>* はタグのキー値に置き換え、*<Value>* はタグの値に置き換えます。

JSON

```
{
    "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>"
                }
            }
        }
    ]
}
```

次のポリシーでは、特定の時間範囲に限り、アカウント内のすべてのスナップショットに対するすべての読み取り EBS direct API の使用を許可します。このポリシーは、aws:CurrentTime グローバル条件キーに基づいて EBS direct API の使用を許可します。このポリシーで、表示されている日時範囲は、必ずポリシーの日時範囲に置き換えます。

JSON

```
{
    "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"
                }
            }
        }
    ]
}
```

詳細については、「AWS Identity and Access Management ユーザーガイド」の「IAM ユーザーのアクセス許可を変更する」を参照してください。

スナップショットを書き込むためのアクセス許可

次のポリシーでは、APIs の書き込みを特定の AWS リージョンのすべてのスナップショットで使用できます。このポリシーで、*<Region>* はスナップショットのリージョンに置き換えます。

次のポリシーでは、特定のキーと値のタグを持つスナップショットに対する書き込み EBS direct API の使用を許可します。このポリシーで、*<Key>* はタグのキー値に置き換え、*<Value>* はタグの値に置き換えます。

JSON

```
{
    "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>"
                }
            }
        }
    ]
}
```

次のポリシーでは、すべての EBS direct API の使用を許可します。また、親スナップショット ID が指定されている場合に限り、`StartSnapshot` アクションを許可します。したがって、このポリシーでは、親スナップショットを使用せずに新しいスナップショットを開始することを禁止します。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ebs:*",
            "Resource": "*",
            "Condition": {
```

```
                "StringEquals": {
                    "ebs:ParentSnapshot": "arn:aws:ec2:*::snapshot/*"
                }
            }
        }
    ]
}
```

次のポリシーでは、すべての EBS direct API の使用を許可します。また、新しいスナップショットに対する user タグキーの作成のみを許可します。また、このポリシーでは、ユーザーがタグを作成するためのアクセス権を持っていることを確認します。タグを指定できるアクションは、StartSnapshot アクションのみです。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ebs:*",
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": "user"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "*"
        }
    ]
}
```

次のポリシーでは、特定の時間範囲に限り、アカウント内のすべてのスナップショットに対するすべての書き込み EBS direct API の使用を許可します。このポリシーは、aws:CurrentTime グローバ

ル条件キーに基づいて EBS direct API の使用を許可します。このポリシーで、表示されている日時範囲は、必ずポリシーの日時範囲に置き換えます。

JSON

```
{
    "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"
                }
            }
        }
    ]
}
```

詳細については、「AWS Identity and Access Management ユーザーガイド」の「IAM ユーザーのアクセス許可を変更する」を参照してください。

を使用するアクセス許可 AWS KMS keys

次のポリシーでは、特定の KMS キーを使用して、暗号化されたスナップショットを復号するための許可を付与します。EBS 暗号化のデフォルトの KMS キーを使用して新しいスナップショットを暗号化するための許可も付与します。このポリシー内で、*<Region>* を KMS キーのリージョンに、*<AccountId>* を KMS キーの AWS アカウント ID に、*<KeyId>* を KMS キーの ID にそれぞれ置き換えます。

> **Note**
>
> デフォルトでは、アカウントのすべてのプリンシパルは Amazon EBS 暗号化用のデフォルトの AWS マネージド KMS キーにアクセスでき、EBS 暗号化および復号オペレーションに使用できます。カスタマーマネージドキーを使用している場合は、新しいキーポリシーを作成するか、カスタマーマネージドキーの既存のキーポリシーを変更して、カスタマーマネージドキーへのアクセス権をプリンシパルに付与する必要があります。詳細については、「AWS Key Management Service デベロッパーガイド」の「AWS KMSでのキーポリシー」を参照してください。

> **Tip**
>
> 最小権限のプリンシパルに従うには、`kms:CreateGrant` へのフルアクセスを許可しないでください。代わりに、次の例に示すように、 AWS サービスによってユーザーに代わって権限が作成された場合にのみ、 `kms:GrantIsForAWSResource`条件キーを使用して、ユーザーが KMS キーに権限を作成できるようにします。

詳細については、「AWS Identity and Access Management ユーザーガイド」の「IAM ユーザーのアクセス許可を変更する」を参照してください。

EBS direct API を使用して Amazon EBS スナップショットを読み取る

次の手順では、EBS direct API を使用してスナップショットを読み取る方法について説明します。

1. スナップショット内にあるすべてのブロックのブロックインデックスとブロックトークンを表示するには、ListSnapshotBlocks アクションを使用します。同じボリュームやスナップショット系列の 2 つのスナップショット間で異なるブロックのブロックインデックスとブロックトークンのみを表示するには、ListChangeBlocks アクションを使用します。これらのアクションは、データを取得するブロックのブロックトークンとブロックインデックスを識別するのに役立ちます。
2. GetSnapshotBlock アクションを使用し、データを取得するブロックのブロックインデックスとブロックトークンを指定します。

> **Note**
>
> アーカイブされたスナップショットで EBS direct API を使用することはできません。

次に、EBS direct API を使用しながらスナップショットを読み取る場合の例を示します。

トピック

- スナップショット内のブロックの一覧表示
- 2 つのスナップショット間で異なるブロックの一覧表示
- スナップショットからのブロックデータの取得

スナップショット内のブロックの一覧表示

AWS CLI

次の list-snapshot-blocks コマンドの例では、スナップショット snap-0987654321 内のブロックのブロックインデックスとブロックトークンを返します。--starting-block-index パラメータは、結果を 1000 より大きいブロックインデックスに制限します。--max-results パラメータは、結果を最初の 100 ブロックに制限します。

```
aws ebs list-snapshot-blocks --snapshot-id snap-0987654321 --starting-block-
index 1000 --max-results 100
```

前のコマンドに対する次のレスポンスの例では、スナップショット内のブロックインデックスとブロックトークンを一覧表示します。get-snapshot-block コマンドを使用し、データを取得するブロックのブロックインデックスとブロックトークンを指定します。ブロックトークンは、表示されている有効期限まで有効です。

```
{
      "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

次の ListSnapshotBlocks リクエストの例では、スナップショット snap-0acEXAMPLEcf41648 にあるブロックのブロックインデックスとブロックトークンを返します。startingBlockIndex パラメータは、結果を 1000 より大きいブロックインデックスに制限します。maxResults パラメータは、結果を最初の 100 ブロックに制限します。

```
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>
```

前のリクエストに対する次のレスポンスの例では、スナップショット内のブロックインデックスとブロックトークンを一覧表示します。GetSnapshotBlock アクションを使用し、データを取得するブロックのブロックインデックスとブロックトークンを指定します。ブロックトークンは、表示されている有効期限まで有効です。

```
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
  }
```

2 つのスナップショット間で異なるブロックの一覧表示

2 つのスナップショット間で、変更されたブロックを一覧表示するためにページ分割されたリクエストを実行する場合は、次の点に注意してください。

- レスポンスには、1 つ以上の空の ChangedBlocks 配列を含めることができます。例:
  - スナップショット 1 - ブロックインデックス 0 - 999 を含む 1000 個のブロックの完全なスナップショット。
  - スナップショット 2 - ブロックインデックス 999 を持つ変更されたブロックが 1 個だけの増分スナップショット。

  これらのスナップショットの変更されたブロックを StartingBlockIndex = 0 および MaxResults = 100 で一覧表示し、ChangedBlocks の空の配列を返します。変更されたブロックが 10 番目の結果セットで返されるまで nextToken を使用して残りの結果をリクエストする必要があります。これにはブロックインデックス 900 - 999 を持つブロックが含まれます。
- レスポンスは、スナップショット内の書き込まれていないブロックをスキップできます。例:
  - スナップショット 1 - ブロックインデックス 2000 - 2999 を含む 1000 個のブロックの完全なスナップショット。
  - スナップショット 2 - ブロックインデックス 2000 を持つ変更されたブロックが 1 個だけの増分スナップショット。

  これらのスナップショットの変更されたブロックを StartingBlockIndex = 0 および MaxResults = 100 で一覧表示すると、レスポンスはブロックインデックス 0 - 1999 をスキップし、ブロックインデックス 2000 を含みます。レスポンスには、空の ChangedBlocks 配列は含まれません。

AWS CLI

次の list-changed-blocks コマンドの例では、スナップショット snap-1234567890 とスナップショット snap-0987654321 の間で異なるブロックのブロックインデックスとブロックトークンを返します。--starting-block-index パラメータは、結果を 0 より大きいブロックインデックスに制限します。--max-results パラメータは、結果を最初の 500 ブロックに制限します。

```
aws ebs list-changed-blocks --first-snapshot-id snap-1234567890 --second-snapshot-id snap-0987654321 --starting-block-index 0 --max-results 500
```

前のコマンドに対する次のレスポンスの例は、2 つのスナップショット間でブロックインデックス 0、6000、6001、6002、6003 が異なることを示しています。さらに、ブロックインデックス 6001、6002、および 6003 は、指定された最初のスナップショット ID にのみ存在し、2 番目のスナップショット ID には存在しません。これは、レスポンスに 2 番目のブロックトークンが表示されないためです。

get-snapshot-block コマンドを使用し、データを取得するブロックのブロックインデックスとブロックトークンを指定します。ブロックトークンは、表示されている有効期限まで有効です。

```
{
      "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

次の ListChangedBlocks リクエストの例では、snap-0acEXAMPLEcf41648 スナップショットと snap-0c9EXAMPLE1b30e2f スナップショットの間で異なるブロックのブロックインデックスとブロックトークンを返します。startingBlockIndex パラメータは、結果を 0 より大きいブロックインデックスに制限します。maxResults パラメータは、結果を最初の 500 ブロックに制限します。

```
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>
```

前のリクエストに対する次のレスポンスの例では、2 つのスナップショット間でブロックインデックス 0、3072、6002、6003 が異なることを示しています。さらに、ブロックインデックス 6002 および 6003 は、指定した最初のスナップショット ID にのみ存在し、2 番目のスナップショット ID には存在しません。これは、レスポンスに 2 番目のブロックトークンが表示されないためです。

GetSnapshotBlock アクションを使用し、データを取得するブロックのブロックインデックスとブロックトークンを指定します。ブロックトークンは、表示されている有効期限まで有効です。

```
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
  }
```

スナップショットからのブロックデータの取得

AWS CLI

次の get-snapshot-block コマンドの例では、スナップショット 6001 内でブロックインデックスが AAABASBpSJ2UAD3PLxJnCt6zun4/T4sU25Bnb8jB5Q6FRXHFqAIAqE04hJoR で、ブロックトークンが snap-1234567890 のデータを返します。バイナリデータは、Windows コンピュータの C:\Temp ディレクトリ内の data ファイルに出力されます。Linux または UNIX コ

ンピュータでコマンドを実行する場合は、出力パスを /tmp/data に置き換え、データを data ディレクトリ内の /tmp ファイルに出力します。

```
aws ebs get-snapshot-block --snapshot-id snap-1234567890 --block-index 6001 --block-token AAABASBpSJ2UAD3PLxJnCt6zun4/T4sU25Bnb8jB5Q6FRXHFqAIAqE04hJoR C:/Temp/data
```

前のコマンドに対するの次のレスポンスの例は、返されたデータのサイズ、データを検証するためのチェックサム、チェックサムのアルゴリズムを示しています。バイナリデータは、リクエストコマンドで指定したディレクトリとファイルに自動的に保存されます。

```
{
      "DataLength": "524288",
      "Checksum": "cf0Y6/Fn0oFa4VyjQPOa/iD0zhTflPTKzxGv2OKowXc=",
      "ChecksumAlgorithm": "SHA256"
  }
```

AWS API

次の GetSnapshotBlock リクエストの例では、スナップショット 3072 内でブロックインデックスが AAUBARGCaufCqBRZC8tEkPYGGkSv3vqvOjJ2xKDi3ljDFiytUxBLXYgTmkid で、ブロックトークンが snap-0c9EXAMPLE1b30e2f のデータを返します。

```
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>
```

前のリクエストに対する次のレスポンスの例は、返されたデータのサイズ、データを検証するためのチェックサム、チェックサムの生成に使用されたアルゴリズムを示しています。バイナリデータは、レスポンスの本文で送信され、次の例では *BlockData* として示されています。

```
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
```

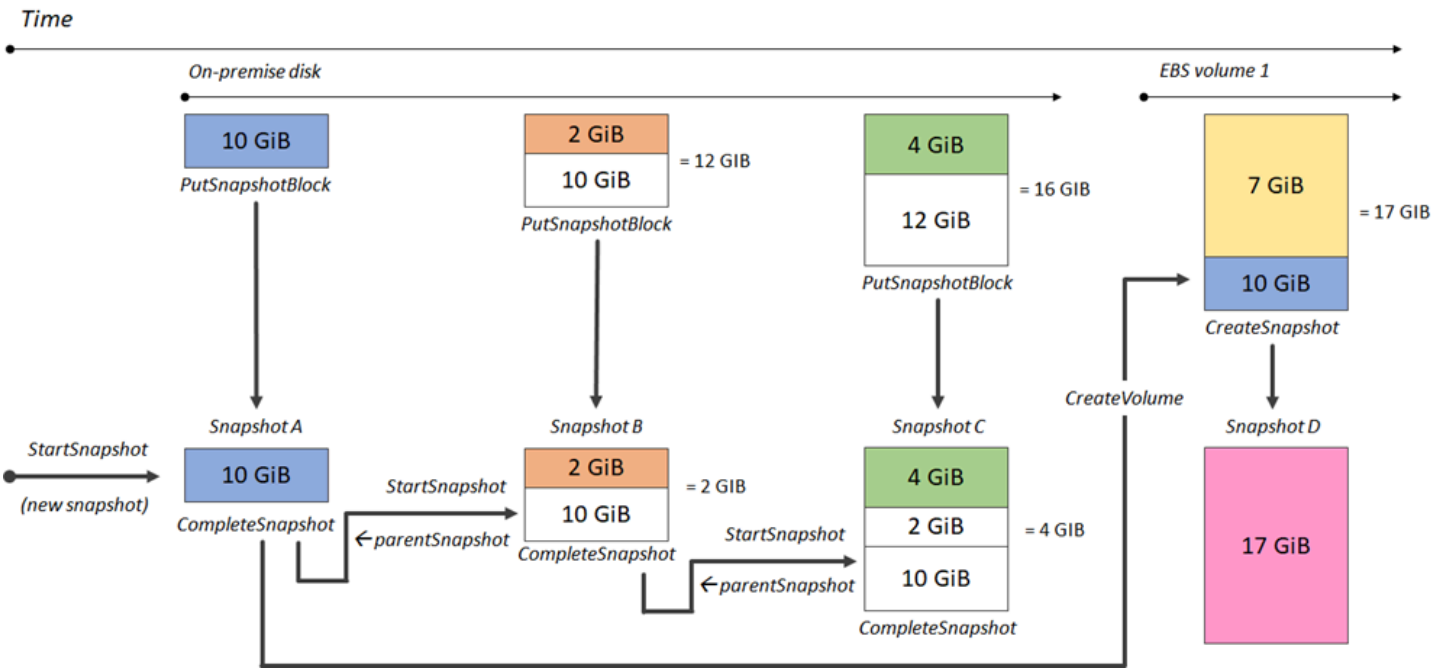
EBS direct API を使用して Amazon EBS スナップショットへ書き込む

次の手順では、EBS direct API を使用して増分スナップショットを書き込む方法について説明します。

1. StartSnapshot アクションを使用して親スナップショット ID を指定し、既存のスナップショットの増分スナップショットとしてスナップショットを開始します。または、親スナップショット ID を省略して新しいスナップショットを開始します。このアクションは、保留状態にある新しいスナップショットの ID を返します。
2. PutSnapshotBlock アクションを使用し、保留中のスナップショットの ID を指定してデータを個別のブロックとして追加します。データのブロックを送信する際に、Base64 でエンコードした SHA256 チェックサムを指定する必要があります。サービスは、受信したデータのチェックサムを計算し、指定したチェックサムと照合してデータを検証します。チェックサムが一致しない場合、アクションは失敗します。
3. 保留中のスナップショットに対するデータの追加が完了したら、CompleteSnapshot アクションを使用して非同期ワークフローを開始し、スナップショットをシールして完了状態に移行させます。

上記の手順を繰り返し、以前に作成したスナップショットを親とする新しい増分スナップショットを作成します。

例えば、次の図のスナップショット A は、最初に開始した新しいスナップショットです。スナップショット A を親スナップショットとしてスナップショット B を開始します。スナップショット B を親スナップショットとしてスナップショット C を開始および作成します。スナップショット A、B、C は、増分スナップショットです。スナップショット A を使用して EBS ボリューム 1 を作成します。スナップショット D を EBS ボリューム 1 から作成します。スナップショット D は A の増分スナップショットであり、B または C の増分スナップショットではありません。



次に、EBS direct API を使用しながらスナップショットに書き込む場合の例を示します。

トピック

- スナップショットの開始
- スナップショットへのデータの書き込み
- スナップショットの完了

スナップショットの開始

AWS CLI

次の start-snapshot コマンドの例では、8 を親スナップショットとして使用し、snap-123EXAMPLE1234567 GiB スナップショットを開始します。新しいスナップショットは、親スナップショットの増分スナップショットになります。指定した 60 分のタイムアウト期間内にスナップショットに対する書き込みリクエストまたは完了リクエストが行われない場合、スナップショットはエラー状態に移行します。550e8400-e29b-41d4-a716-446655440000 クライアントトークンは、リクエストのべき等を保証します。クライアントトークンを省略すると、 AWS SDK によって自動的にクライアントトークンが生成されます。べき等の詳細については、StartSnapshot API リクエストのべき等性を確保を参照してください。

```
aws ebs start-snapshot --volume-size 8 --parent-snapshot snap-123EXAMPLE1234567 --
timeout 60 --client-token 550e8400-e29b-41d4-a716-446655440000
```

前のコマンドに対する次のレスポンスの例は、スナップショット ID、 AWS アカウント ID、ステータス、ボリュームサイズ (GiB)、スナップショット内の各ブロックのサイズを示しています。スナップショットは pending 状態で開始されます。スナップショットにデータを書き込むには、後続の put-snapshot-block コマンドでスナップショット ID を指定します。次に complete-snapshot コマンドを使用してスナップショットを完了し、ステータスを completed に変更します。

```
{
    "SnapshotId": "snap-0aaEXAMPLEe306d62",
    "OwnerId": "111122223333",
    "Status": "pending",
    "VolumeSize": 8,
    "BlockSize": 524288
}
```

AWS API

次の StartSnapshot リクエストの例では、スナップショット 8 を親スナップショットとして使用し、snap-123EXAMPLE1234567 GiB スナップショットを開始します。新しいスナップショットは、親スナップショットの増分スナップショットになります。指定した 60 分のタイムアウト期間内にスナップショットに対する書き込みリクエストまたは完了リクエストが行われない場合、スナップショットはエラー状態に移行します。550e8400-e29b-41d4-a716-446655440000 クライアントトークンは、リクエストのべき等を保証します。クライアントトークンを省略すると、 AWS SDK によって自動的にクライアントトークンが生成されます。べき等の詳細については、StartSnapshot API リクエストのべき等性を確保を参照してください。

```
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
}
```

前のリクエストに対する次のレスポンスの例は、スナップショット ID、 AWS アカウント ID、ステータス、ボリュームサイズ (GiB)、スナップショット内の各ブロックのサイズを示しています。スナップショットは保留状態で開始されます。スナップショットにデータを書き込むには、後続の PutSnapshotBlocks リクエストでスナップショット ID を指定します。

```
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
}
```

スナップショットへのデータの書き込み

AWS CLI

次の put-snapshot-block コマンドの例では、524288 バイトのデータをスナップショット 1000 のブロックインデックス snap-0aaEXAMPLEe306d62 に書き込みます。Base64 でエンコードされた QOD3gmEQOXATfJx2Aa34W4FU2nZGyXfqtsUuktOw8DM= チェックサムが、SHA256 アルゴリズムを使用して生成されています。送信されるデータは、/tmp/data ファイルにあります。

```
aws ebs put-snapshot-block --snapshot-id snap-0aaEXAMPLEe306d62
 --block-index 1000 --data-length 524288 --block-data /tmp/data --
checksum QOD3gmEQOXATfJx2Aa34W4FU2nZGyXfqtsUuktOw8DM= --checksum-algorithm SHA256
```

前のコマンドに対する次のレスポンスの例では、サービスによって受信されたデータのデータ長、チェックサム、チェックサムアルゴリズムが返されます。

```
{
    "DataLength": "524288",
    "Checksum": "QOD3gmEQOXATfJx2Aa34W4FU2nZGyXfqtsUuktOw8DM=",
    "ChecksumAlgorithm": "SHA256"
}
```

AWS API

次の PutSnapshot リクエストの例では、524288 バイトのデータをスナップショット 1000 のブロックインデックス snap-052EXAMPLEc85d8dd に書き込みます。Base64 でエンコードされた QOD3gmEQOXATfJx2Aa34W4FU2nZGyXfqtsUuktOw8DM= チェックサムが、SHA256 アルゴリズムを使用して生成されています。データは、リクエストの本文で送信され、次の例では *BlockData* として示されています。

```
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
```

前のリクエストに対する次のレスポンスの例では、サービスが受信したデータのデータ長、チェックサム、チェックサムアルゴリズムが返されます。

```
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
```

```
{}
```

スナップショットの完了

AWS CLI

次の complete-snapshot コマンドの例では、スナップショット snap-0aaEXAMPLEe306d62 を完了します。このコマンドは、5 ブロックをスナップショットに書き込むことを指定します。6D3nmwi5f2F0wlh7xX8QprrJBFzDX8aacdOcA3KCM3c= チェックサムは、スナップショットに書き込まれたデータセット全体のチェックサムを示します。チェックサムの詳細については、このガイドの前半にあるEBS direct API チェックサムを使用してスナップショットデータを検証を参照してください。

```
aws ebs complete-snapshot --snapshot-id snap-0aaEXAMPLEe306d62 --changed-blocks-
count 5 --checksum 6D3nmwi5f2F0wlh7xX8QprrJBFzDX8aacdOcA3KCM3c= --checksum-
algorithm SHA256 --checksum-aggregation-method LINEAR
```

前のコマンドに対するレスポンスの例を次に示します。

```
{
    "Status": "pending"
}
```

AWS API

次の CompleteSnapshot リクエストの例では、スナップショット snap-052EXAMPLEc85d8dd を完了します。このコマンドは、5 ブロックをスナップショットに書き込むことを指定します。6D3nmwi5f2F0wlh7xX8QprrJBFzDX8aacdOcA3KCM3c= チェックサムは、スナップショットに書き込まれたデータセット全体のチェックサムを示します。

```
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>
```

前のリクエストに対するレスポンスの例を次に示します。

```
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"}
```

EBS direct API の暗号化の結果

StartSnapshot を使用して新しいスナップショットを開始する場合、暗号化ステータスは、Encrypted、KmsKeyArn、および ParentSnapshotId に指定した値と、 AWS アカウントがデフォルトで暗号化が有効になっているかどうかによって異なります。

> **Note**
>
> - 暗号化で EBS direct API を使用するには、追加の IAM 許可が必要になる場合があります。詳細については、「を使用するアクセス許可 AWS KMS keys」を参照してください。
> - AWS アカウントで Amazon EBS 暗号化がデフォルトで有効になっている場合、暗号化されていないスナップショットを作成することはできません。
> - AWS アカウントで Amazon EBS 暗号化がデフォルトで有効になっている場合、暗号化されていない親スナップショットを使用して新しいスナップショットを開始することはできません。最初に親スナップショットをコピーして、これを暗号化する必要があります。詳細については、「Amazon EBS スナップショットのコピー」を参照してください。

トピック

- 暗号化の結果: 暗号化されていない親スナップショット
- 暗号化の結果: 暗号化された親スナップショット
- 暗号化の結果: 親スナップショットなし

暗号化の結果: 暗号化されていない親スナップショット

次の表は、暗号化されていない親スナップショットを指定する場合の設定の可能な組み合わせごとの暗号化の結果を示しています。

ParentSnapshotId	暗号化された	KmsKeyArn	デフォルトでの暗号化	結果
暗号化されていない	省略	省略	有効	リクエストは ValidationException で失敗します。
			Disabled	スナップショットは暗号化されていません。
		指定	有効	
			無効	
暗号化されていない	正	省略	有効	リクエストは ValidationException で失敗します。
			Disabled	
		指定	有効	
			無効	
暗号化されていない	誤	省略	有効	リクエストは ValidationException で失敗します。
			Disabled	
		指定	有効	
			無効	

暗号化の結果: 暗号化された親スナップショット

次の表は、暗号化された親スナップショットを指定する場合の設定の可能な組み合わせごとの暗号化の結果を示しています。

ParentSnapshotId	暗号化された	KmsKeyArn	デフォルトでの暗号化	結果
暗号化された	省略	省略	有効	スナップショットは、親スナップショットと同じ KMS キーを使用して暗号化されます。
			Disabled	
		指定	有効	リクエストは `ValidationException` で失敗します。
			Disabled	
暗号化された	正	省略	有効	リクエストは `ValidationException` で失敗します。
			Disabled	
		指定	有効	
			無効	
暗号化された	誤	省略	有効	リクエストは `ValidationException` で失敗します。
			Disabled	
		指定	有効	
			無効	

暗号化の結果: 親スナップショットなし

次の表は、親スナップショットを使用しない場合の設定の可能な組み合わせごとの暗号化の結果を示しています。

ParentSnapshotId	暗号化された	KmsKeyArn	デフォルトでの暗号化	結果
省略	正	省略	有効	スナップショットは、アカウントのデフォルトの KMS
			Disabled	

ParentSnapshotId	暗号化された	KmsKeyArn	デフォルトでの暗号化	結果
				キーを使用して暗号化されます。*
		指定	有効	スナップショットは、KmsKeyArn 用に指定された KMS キーを使用して暗号化されます。
			Disabled	
省略	誤	省略	有効	リクエストは `ValidationException` で失敗します。
			Disabled	スナップショットは暗号化されていません。
		指定	有効	リクエストは `ValidationException` で失敗します。
			Disabled	
省略	省略	省略	有効	スナップショットは、アカウントのデフォルトの KMS キーを使用して暗号化されます。*
			Disabled	スナップショットは暗号化されていません。
		指定	有効	スナップショットは、KmsKeyArn 用に指定された KMS キーを使用して暗号化されます。
			Disabled	

* このデフォルトの KMS キーは、カスタマーマネージドキーまたは Amazon EBS 暗号化用のデフォルトの AWS マネージド KMS キーです。

EBS direct API チェックサムを使用してスナップショットデータを検証

GetSnapshotBlock アクションは、スナップショットのブロック内のデータを返します。PutSnapshotBlock アクションは、スナップショットのブロックにデータを追加します。転送されるブロックデータは、署名バージョン 4 の署名プロセスの対象外であるため、署名されません。そのため、データの整合性の検証には、次のようにチェックサムが使用されます

- GetSnapshotBlock アクションを使用する場合、レスポンスは x-amz-Checksum ヘッダーによるブロックデータの Base64 で暗号された SHA256 チェックサムと、x-amz-Checksum-Algorithm ヘッダーによるチェックサムアルゴリズムを提供します。返されたチェックサムを使用して、データの整合性が検証されます。生成したチェックサムが Amazon EBS から提供されたチェックサムと一致しない場合は、データが有効でないと判断し、リクエストを再試行する必要があります。
- PutSnapshotBlock アクションを使用する場合、リクエストは x-amz-Checksum ヘッダーによるブロックデータの Base64 で暗号された SHA256 チェックサムと、x-amz-Checksum-Algorithm ヘッダーによるチェックサムアルゴリズムを提供する必要があります。提供したチェックサムと Amazon EBS によって生成されたチェックサムが照合され、データの整合性が検証されます。両者のチェックサムが対応しない場合、リクエストは失敗します。
- CompleteSnapshot アクションを使用する場合、リクエストは、オプションとしてスナップショットに追加するデータセット全体に対する Base64 でエンコードされた SHA256 チェックサムの集計を提供できます。x-amz-Checksum ヘッダーによるチェックサム、x-amz-Checksum-Algorithm ヘッダーによるチェックサムアルゴリズム、x-amz-Checksum-Aggregation-Method ヘッダーによるチェックサム集計方法を提供します。線形集計による集計チェックサムを生成するには、書き込まれた各ブロックのチェックサムをブロックインデックスの昇順に配置し、これらを連結して単一の文字列を形成します。次に、SHA256 アルゴリズムを使用して、この文字列全体のチェックサムを生成します。

これらのアクションのチェックサムは、署名バージョン 4 の署名プロセスの一部です。

StartSnapshot API リクエストのべき等性を確保

べき等性は、API リクエストが必ず 1 回だけで完了するようにします。べき等リクエストでは、元のリクエストが正常に完了した場合、その後の再試行は元の成功したリクエストの結果を返し、追加の効果はありません。

StartSnapshot API は、クライアントトークンを使用したべき等をサポートしています。クライアントトークンは、API リクエストを行うときに指定する一意の文字列です。API リクエストが正常に完了した後で、同じクライアントトークンと同じリクエストパラメータを使用して API リクエストを再試行すると、元のリクエストの結果が返されます。リクエストの再試行で同じクライアントトークンを使用し、リクエストパラメータの 1 つ以上を変更すると、ConflictException エラーが返されます。

独自のクライアントトークンを指定しない場合、 AWS SDKs はリクエストのクライアントトークンを自動的に生成して、それがべき等であることを確認します。

クライアントトークンには、64 文字の ASCII 文字を含む任意の文字列を指定できます。同じクライアントトークンを異なるリクエストに再利用しないでください。

API を使用して独自のクライアントトークンでべき等の StartSnapshot リクエストを行うには

ClientToken リクエストパラメータを指定します。

```
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
}
```

を使用して独自のクライアントトークンを使用してべき等な StartSnapshot リクエストを行うには AWS CLI

client-token リクエストパラメータを指定します。

```
$ 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
```

EBS direct API のエラー再試行

AWS SDK には、エラーレスポンスを返したリクエストを自動的に再試行するロジックが実装されています。 AWS SDK による再試行では、ユーザーによる設定が可能です。詳細については、「SDK のドキュメント」を参照してください。

失敗した場合の自動再試行を AWS CLI により設定できます。の再試行の設定の詳細については AWS CLI、「 AWS Command Line Interface ユーザーガイド」のAWS CLI 「再試行」を参照してください。

AWS クエリ API は、失敗したリクエストの再試行ロジックをサポートしていません。HTTP または HTTPS リクエストを使用している場合は、クライアントアプリケーション内に再試行ロジックを実装する必要があります。

次の表に、発生する API エラーレスポンスを示します。一部の API エラーは再試行可能です。クライアントアプリケーションは、再試行可能なエラーの受け取りに失敗したリクエストを必ず再試行する必要があります。

エラー	Response Code (レスポンスコード)	説明	スローの原因	再試行可能なのは?
`InternalServerException`	500	ネットワークまたは AWS サーバー側の問題により、リクエストが失敗しました。	すべての API	はい
`ThrottlingException`	400	API リクエストの数が、アカウントに対して許可された API リクエストの最大スロットリング制限を超えました。	すべての API	はい

エラー	Response Code (レスポンスコード)	説明	スローの原因	再試行可能なのは?
`RequestThrottleException`	400	API リクエストの数が、スナップショットに対して許可された API リクエストの最大スロットリング制限を超えました。	GetSnapshotBlock | PutSnapshotBlock	はい
メッセージ「`Failed to read block data`」付き `ValidationException`	400	提供されたデータブロックは読み込まれませんでした。	PutSnapshotBlock	はい
他のメッセージ付き `ValidationException`	400	リクエスト構文の形式が正しくないか、入力が AWS のサービスで指定された制約を満たしていません。	すべての API	いいえ
`ResourceNotFoundException`	404	指定されたスナップショット ID が存在しません。	すべての API	いいえ

エラー	Response Code (レスポンスコード)	説明	スローの原因	再試行可能なのは?
`ConflictException`	409	指定されたクライアントトークンは、以前に異なるリクエストパラメータを持つ同様のリクエストで使用されていました。詳細については、「StartSnapshot API リクエストのべき等性を確保」を参照してください。	StartSnapshot	いいえ
`AccessDeniedException`	403	必要なオペレーションを実行するための許可がありません。	すべての API	いいえ
`ServiceQuotaExceededException`	402	リクエストを実行すると、アカウントの 1 つ以上の従属 Service Quotas を超えるため、リクエストは失敗しました。	すべての API	いいえ

エラー	Response Code (レスポンスコード)	説明	スローの原因	再試行可能なのは?
`InvalidSignatureException`	403	認可署名リクエストの有効期限切れです。リクエストを再試行するには、認可署名を更新する必要があります。	すべての API	いいえ

EBS direct API のパフォーマンスを最適化

API リクエストは複数を同時に実行できます。PutSnapshotBlock のレイテンシーが 100 ミリ秒であると仮定すると、スレッドは 1 秒間に 10 個のリクエストを処理できます。さらに、クライアントアプリケーションが複数のスレッドと接続 (100 件の接続など) を作成すると仮定すると、合計で 1 秒あたり 1000 (10 * 100) 件のリクエストを行うことができます。これは、1 秒あたり約 500 MB のスループットに相当します。

次のリストは、アプリケーションで確認すべき項目を示しています。

- スレッドごとに別個の接続を使用しているか? アプリケーションで接続数が制限されている場合、接続が利用可能になるまで複数のスレッドが待機するため、スループットが低下します。
- アプリケーションで 2 つの書き込みリクエスト間に待機時間があるか? 待機時間があると、スレッドの有効なスループットが低下します。
- インスタンスの帯域幅制限: インスタンスの帯域幅が他のアプリケーションによって共有されている場合、PutSnapshotBlock リクエストに使用可能なスループットが制限される場合があります。

ボトルネックを避けるために、アカウントで実行されている他のワークロードに注意してください。また、EBS direct API ワークフロー内に再試行メカニズムを組み込んで、スロットリング、タイムアウト、サービスの利用不可に対処する必要があります。

EBS direct API Service Quotas を確認し、1 秒あたりに実行できる API リクエストの最大数を判断します。詳細については、AWS 全般のリファレンスのAmazon Elastic Block Store エンドポイントとクォータを参照してください。

EBS direct API 用のサービスエンドポイント

エンドポイントは、 AWS ウェブサービスのエントリポイントとして機能する URL です。EBS direct API は以下のエンドポイントタイプをサポートします。

- IPv4 エンドポイント
- IPv4 と IPv6 の両方をサポートするデュアルスタックのエンドポイント
- FIPS エンドポイント

リクエストを行うと、使用するエンドポイントとリージョンを指定できます。エンドポイントを指定しない場合、デフォルトで IPv4 エンドポイントが使用されます。別のエンドポイントタイプを使用するには、リクエストで指定する必要があります。これを行う方法の例については、「エンドポイントの指定」を参照してください。

リージョンの詳細については、「Amazon EC2 ユーザーガイド」の「リージョンとアベイラビリティーゾーン」を参照してください。EBS direct API のエンドポイントのリストについては、「Amazon Web Services 全般のリファレンス」の「EBS direct API のエンドポイント」を参照してください。

トピック

- IPv4 エンドポイント
- デュアルスタック (IPv4 および IPv6) エンドポイント
- FIPS エンドポイント
- エンドポイントの指定

IPv4 エンドポイント

IPv4 エンドポイントは IPv4 トラフィックのみをサポートします。IPv4 エンドポイントは、全リージョンで利用できます。

EBS direct API は、リクエストの実行に使用できるリージョン IPv4 エンドポイントのみをサポートします。リージョンをエンドポイント名の一部として指定する必要があります。エンドポイント名には、次の命名規則が使用されます。

- ebs.*region*.amazonaws.com

例えば、us-east-2 IPv4 エンドポイントへリクエストを指示する場合、ebs.us-east-2.amazonaws.com をエンドポイントとして指定する必要があります。EBS direct API のエンドポイントのリストについては、「Amazon Web Services 全般のリファレンス」の「EBS direct API のエンドポイント」を参照してください。

料金

EBS direct API と Amazon EC2 インスタンス間で、同一リージョン内の IPv4 エンドポイントを使用して直接転送されたデータについては、課金されません。ただし、 AWS PrivateLink エンドポイント、NAT ゲートウェイ、Amazon VPC トランジットゲートウェイなどの中間サービスがある場合は、関連するコストが課金されます。

デュアルスタック (IPv4 および IPv6) エンドポイント

デュアルスタックエンドポイントは、IPv4 と IPv6 トラフィックの両方をサポートします。デュアルスタックエンドポイントは、すべてのリージョンで利用できます。

IPv6 を使用するには、デュアルスタックエンドポイントを使用する必要があります。デュアルスタックエンドポイントにリクエストを行うと、エンドポイント URL は、ネットワークとクライアントが使用するプロトコルに応じて IPv6 または IPv4 アドレスに解決されます。

EBS direct API はリージョンのデュアルスタックエンドポイントのみをサポートしているため、エンドポイント名の一部としてリージョンを指定する必要があります。デュアルスタックエンドポイント名には、次の命名規則が使用されます。

- ebs.*region*.api.aws

例えば、eu-west-1 リージョンのデュアルスタックエンドポイント名は、ebs.eu-west-1.api.aws です。EBS direct API のエンドポイントのリストについては、「Amazon Web Services 全般のリファレンス」の「EBS direct API のエンドポイント」を参照してください。

料金

EBS direct API と Amazon EC2 インスタンス間で、同一リージョン内のデュアルスタックエンドポイントを使用して直接転送されたデータについては、課金されません。ただし、 AWS PrivateLink エンドポイント、NAT ゲートウェイ、Amazon VPC トランジットゲートウェイなどの中間サービスがある場合は、関連するコストが課金されます。

FIPS エンドポイント

EBS direct API は、FIPS 検証済みの IPv4 およびデュアルスタック (IPv4 および IPv6) エンドポイントを提供します。

- us-east-1 — 米国東部 (バージニア北部)
- us-east-2 — 米国東部 (オハイオ)
- us-west-1 — 米国西部 (北カリフォルニア)
- us-west-2 — 米国西部 (オレゴン)
- ca-central-1 — カナダ (中部)
- ca-west-1 — カナダ西部 (カルガリー)

FIPS IPv4 エンドポイントに使用される命名規則は ebs-fips.*region*.amazonaws.com です。例えば、us-east-1 の FIPS IPv4 エンドポイントは ebs-fips.us-east-1.amazonaws.com です。

FIPS デュアルスタックエンドポイントの命名規則は ebs-fips.*region*.api.aws です。例えば、us-east-1 の FIPS デュアルスタックエンドポイントは ebs-fips.us-east-1.api.aws です。

FIPS エンドポイントの詳細については、「Amazon Web Services 全般のリファレンス」の「FIPS エンドポイント」を参照してください。

エンドポイントの指定

このセクションでは、リクエストを行うときにエンドポイントを指定する方法を例で示します。

AWS CLI

次の例は、 AWS CLIを使用して us-east-2 リージョンのエンドポイントを指定する方法を示しています。

- デュアルスタック

```
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

次の例は、 AWS SDK for Java 2.xを使用して us-east-2 リージョンのエンドポイントを指定する方法を示しています。

- デュアルスタック

```
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

次の例は、 AWS SDK for Goを使用して us-east-2 リージョンのエンドポイントを指定する方法を示しています。

- デュアルスタック

```
sess := session.Must(session.NewSession())
svc := ebs.New(sess, &aws.Config{
    Region: aws.String(endpoints.UsEast2RegionID),
    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.UsEast2RegionID),
    Endpoint: aws.String("https://ebs.us-east-2.amazonaws.com")
})
```

AWS EBS direct APIsの SDK コード例

次のコード例は、 AWS Software Development Kit (SDK) で EBS direct APIs を使用する方法を示しています。

アクション

- AWS SDK または CLI StartSnapshotで を使用する
- AWS SDK または CLI PutSnapshotBlockで を使用する
- AWS SDK または CLI CompleteSnapshotで を使用する

AWS SDK または CLI `StartSnapshot`で を使用する

次の例は、`StartSnapshot` を使用する方法を説明しています。

Rust

SDK for Rust

> **Note**
>
> GitHub には、その他のリソースもあります。用例一覧を検索し、AWS コード例リポジトリでの設定と実行の方法を確認してください。

```
async fn start(client: &Client, description: &str) -> Result<String, Error> {
    let snapshot = client
        .start_snapshot()
        .description(description)
        .encrypted(false)
```

```
        .volume_size(1)
        .send()
        .await?;

    Ok(snapshot.snapshot_id.unwrap())
}
```

- API の詳細については、AWS SDK for Rust API リファレンスの「StartSnapshot」を参照してください。

AWS SDK または CLI **PutSnapshotBlock**で を使用する

次の例は、PutSnapshotBlock を使用する方法を説明しています。

Rust

SDK for Rust

> **Note**
>
> GitHub には、その他のリソースもあります。用例一覧を検索し、AWS コード例リポジトリでの設定と実行の方法を確認してください。

```
async fn add_block(
    client: &Client,
    id: &str,
    idx: usize,
    block: Vec<u8>,
    checksum: &str,
) -> Result<(), Error> {
    client
        .put_snapshot_block()
        .snapshot_id(id)
        .block_index(idx as i32)
        .block_data(ByteStream::from(block))
        .checksum(checksum)
        .checksum_algorithm(ChecksumAlgorithm::ChecksumAlgorithmSha256)
        .data_length(EBS_BLOCK_SIZE as i32)
        .send()
```

```
        .await?;

    Ok(())
}
```

- API の詳細については、AWS SDK for Rust API リファレンスの「PutSnapshotBlock」を参照してください。

AWS SDK または CLI **CompleteSnapshot**で を使用する

次の例は、CompleteSnapshot を使用する方法を説明しています。

Rust

SDK for Rust

> **Note**
>
> GitHub には、その他のリソースもあります。用例一覧を検索し、AWS コード例リポジトリでの設定と実行の方法を確認してください。

```
async fn finish(client: &Client, id: &str) -> Result<(), Error> {
    client
        .complete_snapshot()
        .changed_blocks_count(2)
        .snapshot_id(id)
        .send()
        .await?;

    println!("Snapshot ID {}", id);
    println!("The state is 'completed' when all of the modified blocks have been
 transferred to Amazon S3.");
    println!("Use the get-snapshot-state code example to get the state of the
 snapshot.");

    Ok(())
}
```

- API の詳細については、AWS SDK for Rust API リファレンスの「CompleteSnapshot」を参照してください。

VPC と EBS direct API 間にプライベート接続を作成

AWS PrivateLink を利用した インターフェイス VPC エンドポイントを作成することで、VPC と EBS direct API 間にプライベート接続を確立できます。インターネットゲートウェイ、NAT デバイス、VPN 接続、または AWS Direct Connect 接続を使用せずに、VPC 内にあるかのように EBS direct API にアクセスできます。VPC のインスタンスは、パブリック IP アドレスがなくても EBS direct API と通信できます。

インターフェイスエンドポイントに対して有効にする各サブネットにエンドポイントネットワークインターフェイスを作成します。

詳細については、「 AWS PrivateLink ガイド」の「Access AWS のサービス through AWS PrivateLink」を参照してください。

EBS direct API VPC エンドポイントに関する考慮事項

EBS direct API でインターフェイス VPC エンドポイントを設定する前に、「AWS PrivateLink ガイド」の「考慮事項」を確認してください。

デフォルトでは、エンドポイントを通じた EBS direct API へのフルアクセスが許可されています。VPC エンドポイントポリシーを使用してインターフェイスエンドポイントへのアクセスを制御できます。VPC エンドポイントに EBS direct API へのアクセスを制御するエンドポイントポリシーをアタッチできます。このポリシーでは、以下の情報を指定します。

- アクションを実行できるプリンシパル。
- 実行可能なアクション。
- アクションを実行できるリソース。

詳細については、Amazon VPC ユーザーガイドのVPC エンドポイントによるサービスのアクセスコントロールを参照してください。

EBS direct API のエンドポイントポリシーの例を次に示します。エンドポイントにアタッチされると、このポリシーは、キー `Environment` と値 `Test` でタグ付けされたスナップショットを除く、すべてのリソースに対するすべての EBS direct API アクションへのアクセスを許可します。

```
{
    "Statement": [
        {
            "Effect": "Deny",
            "Action": "ebs:*",
            "Principal": "*",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Environment": "Test"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ebs:*",
            "Principal": "*",
            "Resource": "*"
        }
    ]
}
```

EBS direct API でインターフェイス VPC エンドポイントを作成する

EBS direct API 用の VPC エンドポイントは、Amazon VPC コンソールまたは AWS Command Line Interface (AWS CLI) で作成できます。詳細については、「AWS PrivateLink ガイド」の「Create a VPC endpoint」を参照してください。

EBS direct API 用の VPC エンドポイントを作成するには、次のいずれかのサービス名を使用します。

- `com.amazonaws.region.ebs`
- `com.amazonaws.region.ebs-fips` — 連邦情報処理標準 (FIPS) 刊行物 140-2 米国政府標準に準拠したインターフェイス VPC エンドポイントを作成します。

> **Note**
>
> FIPS 準拠のインターフェイス VPC エンドポイントは、`us-east-1` | `us-east-2` | `us-west-1` | `us-west-2` | `ca-central-1` | `ca-west-1` のリージョンで作成できま

す。FIPS 準拠のインターフェイス VPC エンドポイントは、IPv4 トラフィックと IPv6 トラフィックの両方をサポートします。

エンドポイントのプライベート DNS を有効にすると、リージョンのデフォルト DNS 名 (`ebs.us-east-1.amazonaws.com` など) を使用して、EBS direct API への API リクエストを実行できます。

を使用して EBS direct APIsログに記録する AWS CloudTrail

EBS direct APIsは、ユーザー AWS CloudTrail、ロール、または サービスによって実行されたアクションを記録する AWS サービスである と統合されています。CloudTrail は、EBS direct API へのコールをイベントとしてキャプチャします。キャプチャされた呼び出しには、 からの呼び出し AWS マネジメントコンソール と EBS direct APIs。CloudTrail で収集された情報を使用して、EBS direct API に対するリクエスト、リクエスト元の IP アドレス、リクエストの作成日時、その他の詳細を確認できます。

各イベントまたはログエントリには、誰がリクエストを生成したかという情報が含まれます。アイデンティティ情報は、以下を判別するのに役立ちます。

- ルートユーザーまたはユーザー認証情報のどちらを使用してリクエストが送信されたか。
- リクエストが IAM Identity Center ユーザーに代わって行われたかどうか。
- リクエストがロールまたはフェデレーションユーザーのテンポラリなセキュリティ認証情報を使用して行われたかどうか。
- リクエストが、別の AWS のサービスによって送信されたかどうか。

CloudTrail は、アカウントを作成する AWS アカウント と でアクティブになり、CloudTrail イベント履歴に自動的にアクセスできます。CloudTrail の [イベント履歴] では、 AWS リージョンで過去 90 日間に記録された管理イベントの表示、検索、およびダウンロードが可能で、変更不可能な記録を確認できます。詳細については、「AWS CloudTrail ユーザーガイド」の「CloudTrail イベント履歴の使用」を参照してください。[イベント履歴] の閲覧には CloudTrail の料金はかかりません。

AWS アカウント 過去 90 日間のイベントの継続的な記録については、証跡または CloudTrail Lake イベントデータストアを作成します。

CloudTrail 証跡

証跡により、CloudTrail はログファイルを Amazon S3 バケットに配信できます。を使用して作成されたすべての証跡 AWS マネジメントコンソール はマルチリージョンです。 AWS CLIを使用

する際は、単一リージョンまたは複数リージョンの証跡を作成できます。アカウント AWS リージョン 内のすべての でアクティビティをキャプチャするため、マルチリージョン証跡を作成することをお勧めします。単一リージョンの証跡を作成する場合、証跡の AWS リージョンに記録されたイベントのみを表示できます。証跡の詳細については、「AWS CloudTrail ユーザーガイド」の「AWS アカウントの証跡の作成」および「組織の証跡の作成」を参照してください。

証跡を作成すると、進行中の管理イベントのコピーを 1 つ無料で CloudTrail から Amazon S3 バケットに配信できますが、Amazon S3 ストレージには料金がかかります。CloudTrail の料金の詳細については、「AWS CloudTrail の料金」を参照してください。Amazon S3 の料金に関する詳細については、「Amazon S3 の料金」を参照してください。

CloudTrail Lake イベントデータストア

[CloudTrail Lake] を使用すると、イベントに対して SQL ベースのクエリを実行できます。CloudTrail Lake は、行ベースの JSON 形式の既存のイベントを Apache ORC 形式に変換します。ORC は、データを高速に取得するために最適化された単票ストレージ形式です。イベントは、イベントデータストアに集約されます。イベントデータストアは、高度なイベントセレクタを適用することによって選択する条件に基づいた、イベントのイミュータブルなコレクションです。どのイベントが存続し、クエリに使用できるかは、イベントデータストアに適用するセレクタが制御します。CloudTrail Lake の詳細については、 AWS CloudTrail ユーザーガイドのAWS CloudTrail 「Lake の使用」を参照してください。

CloudTrail Lake のイベントデータストアとクエリにはコストがかかります。イベントデータストアを作成する際に、イベントデータストアに使用する料金オプションを選択します。料金オプションによって、イベントの取り込みと保存にかかる料金、および、そのイベントデータストアのデフォルトと最長の保持期間が決まります。CloudTrail の料金の詳細については、「AWS CloudTrail の料金」を参照してください。

CloudTrail の EBS direct API データイベント

データイベントでは、リソース上またはリソース内で実行されたリソースオペレーションについての情報が得られます。これらのイベントは、データプレーンオペレーションとも呼ばれます。データイベントは、多くの場合、高ボリュームのアクティビティです。デフォルトでは、CloudTrail はデータイベントをログ記録しません。CloudTrail [イベント履歴] にはデータイベントは記録されません。

追加の変更がイベントデータに適用されます。CloudTrail の料金の詳細については、「AWS CloudTrail の料金」を参照してください。

CloudTrail コンソール、または CloudTrail CloudTrail API オペレーションを使用して AWS CLI、APIs direct API リソースタイプのデータイベントを記録できます。データイベントをログに記録する方法の詳細については、「AWS CloudTrail ユーザーガイド」の「AWS マネジメントコンソールを使用したデータイベントのログ記録」および「AWS Command Line Interfaceを使用したデータイベントのログ記録」を参照してください。

次の EBS direct API オペレーションをデータイベントとしてログに記録できます。

- ListSnapshotBlocks
- ListChangedBlocks
- GetSnapshotBlock
- PutSnapshotBlock

> **Note**
>
> 共有されているスナップショットに対してアクションを実行した場合、データイベントはスナップショットを所有する AWS アカウントに送信されません。

CloudTrail の EBS direct API 管理イベント

管理イベントは、 のリソースで実行される管理オペレーションに関する情報を提供します AWS アカウント。これらのイベントは、コントロールプレーンオペレーションとも呼ばれます。CloudTrail は、デフォルトで管理イベントをログ記録します。

EBS direct API サービスは、次のコントロールプレーンオペレーションを管理イベントとして CloudTrail にログ記録します。

- StartSnapshot
- CompleteSnapshot

EBS direct API イベント例

各イベントは任意の送信元からの単一のリクエストを表し、リクエストされた API オペレーション、オペレーションの日時、リクエストパラメータなどに関する情報を含みます。CloudTrail ログファイルは、パブリック API コールの順序付けられたスタックトレースではないため、イベントは特定の順序で表示されません。

EBS direct API の CloudTrail イベント例を以下に示します。

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"
```

```
    }
}
```

CloudTrail レコードの内容については、「AWS CloudTrail ユーザーガイド」の「CloudTrail record contents」を参照してください。

EBS direct API に関するよくある質問

ステータスが保留中になっているスナップショットに、EBS direct API からアクセスできますか?

いいえ。スナップショットは、完了ステータスの場合のみアクセスできます。

ブロックインデックスは、EBS direct API から数値順に返されますか?

はい。返されるブロックインデックスは一意で、数値順になっています。

MaxResults パラメータ値が 100 未満のリクエストを送信できますか?

いいえ。使用できる最小の MaxResult パラメータ値は 100 です。MaxResult パラメータ値が 100 未満のリクエストを送信し、スナップショット内に 100 を超えるブロックがあった場合、API は最低 100 の結果を返します。

複数の API リクエストを同時に実行できますか?

API リクエストは複数を同時に実行できます。ボトルネックを避けるために、アカウントで実行されている他のワークロードに注意してください。また、EBS direct API ワークフロー内に再試行メカニズムを組み込んで、スロットリング、タイムアウト、サービスの利用不可に対処する必要があります。詳細については、EBS direct API のパフォーマンスを最適化を参照してください。

EBS direct API Service Quotas を確認して、1 秒あたりに実行できる API リクエストの数を判断します。詳細については、AWS 全般のリファレンスのAmazon Elastic Block Store エンドポイントとクォータを参照してください。

ListChangedBlocks アクションを実行したときに、スナップショット内にブロックがあっても空のレスポンスが返されることがありますか?

はい。スナップショット内の変更されたブロックが少ない場合、レスポンスは空になる場合があります。ただし、API は次ページのトークン値を返します。次ページのトークン値を使用して、結果の次ページに進みます。API から返された次ページのトークン値が null である場合は、結果の最終ページに達したことを確認できます。

NextToken パラメータと StartingBlockIndex パラメータを一緒に指定した場合、どちらのパラメータが使用されますか?

NextToken が使用され、StartingBlockIndex は無視されます。

ブロックトークンとネクストトークンの有効期間はどれくらいですか?

ブロックトークンの有効期間は 7 日で、ネクストトークンの有効期間は 60 分です。

暗号化されたスナップショットはサポートされますか?

はい。暗号化されたスナップショットには、EBS ダイレクト API を使用してアクセスできます。

暗号化されたスナップショットにアクセスするには、スナップショットの暗号化に使用される KMS キーと復 AWS KMS 号アクションにアクセスできる必要があります。ユーザーに割り当てる AWS KMS ポリシーについては、このガイドの前半IAM を使用して EBS direct API へのアクセスを制御のセクションを参照してください。

パブリックスナップショットはサポートされていますか?

パブリックスナップショットはサポートされていません。

での Amazon EBS ローカルスナップショットは AWS Outposts サポートされていますか?

の Amazon EBS ローカルスナップショット AWS Outposts はサポートされていません。

スナップショットブロックのリストは、スナップショット内のすべてのブロックインデックスとブロックトークンを返すのですか、それともデータが書き込まれたものだけを返すのですか?

データが書き込まれたブロックインデックスとブロックトークンのみを返します。

セキュリティ分析および運用に関するトラブルシューティングを行うために、アカウントで EBS direct API によって実行された API コールの履歴を取得できますか?

はい。アカウントで EBS direct API によって実行された API コールの履歴を取得するには、AWS マネジメントコンソールで AWS CloudTrail を有効にします。詳細については、「を使用して EBS direct APIsログに記録する AWS CloudTrail」を参照してください。

削除した EBS ボリューム、EBS スナップショット、EBS-backed AMIs をごみ箱で復元する

ごみ箱は、誤って削除された EBS ボリューム、EBS スナップショット、および EBS-backed AMI を復元することを可能にするデータ復旧機能です。ごみ箱を使用する場合、リソースが削除されると、リソースは、完全に削除されるまでの時間として指定した期間、ごみ箱に保持されます。

リソースは、保持期間が終了する前であればいつでもごみ箱から復元できます。ごみ箱からリソースを復元すると、そのリソースはごみ箱から削除され、アカウント内の他のそのタイプのリソースと同じ方法で使用できます。保持期間が終了し、リソースが復元されない場合、リソースはごみ箱から完全に削除され、復旧できなくなります。

ごみ箱は、ビジネスクリティカルなデータを誤って削除しないように保護することで、ビジネス継続性を確保するのに役立ちます。

ごみ箱は、Amazon Elastic Block Store (Amazon EBS) のサービス機能として評価されます。Amazon EBS がリストされたコンプライアンスプログラムによるAWS 対象範囲内のサービス (FedRAMP、HIPAA BAA、SOC など) は、ごみ箱にも適用されます。

トピック

- サポートされているリソース
- ごみ箱の仕組み
- ごみ箱に関する考慮事項
- クォータ
- 関連サービス
- 料金
- IAM でごみ箱へのアクセスを制御
- ごみ箱の保持ルールを作成
- 既存のごみ箱の保持ルールを更新
- ごみ箱の保持ルールをロックして、更新または削除されないようにする
- ごみ箱の保持ルールをロック解除して、更新または削除できるようにする
- ごみ箱の保持ルールをタグ
- ごみ箱の保持ルールを削除することで、ごみ箱がリソース保持するのを防止
- ごみ箱から削除されたスナップショットを復元

- 削除したボリュームをごみ箱から復旧する
- 削除された AMI をごみ箱から復元
- Amazon EventBridge を使用してごみ箱をモニタリングする
- を使用してごみ箱をモニタリングする AWS CloudTrail
- ごみ箱のサービスエンドポイント
- VPC とごみ箱の間にプライベート接続を作成する

サポートされているリソース

ごみ箱は、次のリソースタイプをサポートしています。

- Amazon EBS ボリューム
- Amazon EBS snapshots

> **Important**
>
> ごみ箱の保存ルールは、アーカイブストレージ階層のアーカイブされたスナップショットにも適用されます。ごみ箱の保持ルールに一致するアーカイブスナップショットを削除すると、アーカイブされたスナップショットは、保持ルールで定義されている保持期間中、ごみ箱に保持されます。アーカイブされたスナップショットは、ごみ箱に入っている間、アーカイブされたスナップショットの料金で請求されます。

- Amazon EBS-backed Amazon マシンイメージ (AMI)

> **Note**
>
> 保持ルールは無効になっている AMI にも適用されます。

ごみ箱の仕組み

ごみ箱を有効にして使用するには、リソースを保護する AWS リージョンに保持ルールを作成する必要があります。保持ルールでは、以下を指定します。

- 保護するリソースタイプ (ボリューム、スナップショット、または AMIs)。
- 保持ルールのタイプ:

  - タグレベルの保持ルール — これらの保持ルールでは、リソースタグを使用して保護するリソースを識別します。保持ルールごとに、1 つ以上のタグのキーと値のペアを指定します。これらのタグキーと値のペアのうち少なくとも 1 つを含む (指定されたタイプの) リソースは、削除時に自動的にごみ箱に保持されます。タグに基づいてアカウント内の特定のリソースを保護するには、このタイプの保持ルールを使用します。
  - リージョンレベルの保持ルール — これらの保持ルールは、デフォルトでは、リソースにタグが付けられていない場合でも、リージョン内のすべての (指定されたタイプの) リソースに適用されます。ただし、除外タグを指定して、特定のタグを持つリソースを除外できます。リージョン内のすべての指定タイプのリソースを保護する場合は、このタイプの保持ルールを使用します。
- 削除されたリソースを保持する保持期間。この期間が終了すると、リソースはごみ箱から完全に削除されます。サポートされている保持期間は次のとおりです。
  - EBS ボリューム: 1 ~ 7 日
  - EBS スナップショットと EBS-backed AMIs: 1 ~ 365 日

リソースがごみ箱に入っている間は、いつでもそのリソースを復元して使用できます。リソースは、次のいずれかの結果になるまで、ごみ箱に残ります。

- 使用するために手動で復元した場合。ごみ箱からリソースを復元すると、そのリソースはごみ箱から削除され、直ちに使用できるようになります。復旧されたリソースは、アカウント内のそのタイプの他のリソースと同じ方法で使用できます。
- 保持期間が終了した場合。保存期間が終了し、リソースがごみ箱から復元されていない場合、リソースはごみ箱から完全に削除され、表示や復元はできなくなります。

ごみ箱に関する考慮事項

ごみ箱および保持ルールを使用する場合は、次の考慮事項が適用されます。

一般的な考慮事項

- 削除されたリソースは、既存の保持ルールと一致する場合にのみごみ箱に移動されます。保持ルールと一致しないリソースを削除した場合、またはその時点で保持ルールが存在しない場合、そのリソースはごみ箱に移動されることなく完全に削除されます。
-

> **Important**
>
> 保持ルールは、アカウント内のリージョンごとにリソースタイプごとに作成された最初の保持ルールの結果整合性モデルに従います。リージョン内のリソースタイプの最初の保持

ルールを作成すると、そのルールがアクティブにならず、リソースの保持がすぐに開始される可能性があります。ただし、同じリージョンで同じリソースタイプに対して作成した後続の保持ルールはアクティブになり、ほぼすぐにリソースの保持が開始されます。

- 削除時にリソースが複数の保持ルールと一致する場合は、保持期間が最も長い保持ルールが優先されます。
- リソースをごみ箱から手動で削除することはできません。リソースは、保持期間が終了すると自動的に削除されます。
- リソースがごみ箱に入っている間は、そのリソースを表示したり、復元したり、タグを変更することしかできません。リソースを使用するには、まずリソースを復元する必要があります。
- AWS Backup や Amazon Data Lifecycle Manager AWS のサービスなどの がある場合、 は保持ルールに一致するリソースを削除し、そのリソースはごみ箱によって自動的に保持されます。必要に応じて、これらのリソースにタグを付け、それらのタグを除外タグとして保持ルールに追加することで、削除時にこれらのリソースがごみ箱に入らないようにすることができます。
- リソースをごみ箱に送信すると、次のシステム生成タグがリソースに割り当てられます。
  - タグキー — `aws:recycle-bin:resource-in-bin`
  - タグ値 — `true`

  このタグを手動で編集または削除することはできません。リソースをごみ箱から復元すると、タグは自動的に削除されます。

ボリュームの考慮事項

- インスタンスの終了またはルートボリュームの交換によって削除されたボリュームは、ごみ箱によって保護されます。
- 作成に失敗したボリュームは、削除時にごみ箱によって保護されません。
- 失敗したインスタンス起動のボリュームは、削除時にごみ箱によって保護されません。
- マネージドインスタンスのボリュームは、削除時にごみ箱によって保護されません。
- 進行中のボリュームの作成または変更は、ボリュームがごみ箱に入ったときに一時停止されません。つまり、ボリュームがボリューム初期化の Amazon EBS プロビジョンドレートで作成された場合でも、それに応じて課金されます。
- ごみ箱のボリュームは、通常のボリュームと同じ方法でクォータにカウントされます。
- ごみ箱のボリュームは、ごみ箱の終了時間が経過した後は請求されません。これらのボリュームは復元できませんが、まだ削除されていない場合は検出できます。

- `deleteVolume` イベントは、ボリュームがごみ箱から削除された後にのみ送信されます。このイベントは、ボリュームがごみ箱に入ったときに出力されません。

スナップショットに関する考慮事項

-
> **Important**
>
> AMI および関連するスナップショットの保持ルールがある場合は、スナップショットの保持期間を AMI の保持期間と同等以上にしてください。これにより、AMI 自体を削除する前に AMI に関連付けられたスナップショットがごみ箱で削除されないようになります。これは、このようなスナップショットを削除すると、AMI が復旧不能になるためです。

- スナップショットが削除されたときに高速スナップショット復元が有効になっている場合、スナップショットがごみ箱に送信された直後に、高速スナップショット復元は自動的に無効になります。
  - スナップショットの高速スナップショット復元が無効になる前にスナップショットを復元しても、そのスナップショットは有効なままになります。
  - スナップショットを復元すると、高速スナップショット復元が無効になった後も、無効のままになります。必要に応じて、高速スナップショット復元を手動で再度有効にする必要があります。
- 削除時に共有されていたスナップショットは、ごみ箱に移動されると自動的に共有が解除されます。スナップショットを復元すると、以前の共有権限がすべて自動的に復元されます。
- などの別の AWS サービスによって作成されたスナップショット AWS Backup がごみ箱に送信され、後でそのスナップショットをごみ箱から復元した場合、そのスナップショットはそれを作成した AWS サービスによって管理されなくなります。スナップショットが不要になった場合は、手動で削除する必要があります。

AMI に関する考慮事項

- Amazon EBS-backed AMI のみがサポートされます。
-
> **Important**
>
> AMI および関連するスナップショットの保持ルールがある場合は、スナップショットの保持期間を AMI の保持期間と同等以上にしてください。これにより、AMI 自体を削除する前に AMI に関連付けられたスナップショットがごみ箱で削除されないようになります。これは、このようなスナップショットを削除すると、AMI が復旧不能になるためです。

- 削除時に共有されていた AMI は、ごみ箱に移動されると自動的に共有が解除されます。AMI を復元すると、以前の共有許可がすべて自動的に復元されます。
- AMI をごみ箱から復元する前に、まず関連するすべてのスナップショットをごみ箱から復元し、それらが `available`状態であることを確認します。
- AMI に関連付けられているスナップショットをごみ箱から削除すると、AMI は復旧できなくなります。AMI は、保持期間が経過すると削除されます。
- AWS Backup などの別の AWS サービスによって作成された AMI がごみ箱に送信され、後でその AMI をごみ箱から復元した場合、その AMI はそれを作成した AWS サービスによって管理されなくなります。AMI が不要になった場合は、手動で削除する必要があります。

Amazon Data Lifecycle Manager スナップショットポリシーに関する考慮事項

- Amazon Data Lifecycle Manager が保持ルールと一致するスナップショットを削除すると、そのスナップショットは自動的にごみ箱に保持されます。
- Amazon Data Lifecycle Manager がポリシーの保持しきい値に達したときにスナップショットを削除してごみ箱に移動し、そのスナップショットをごみ箱から手動で復元した場合は、スナップショットが不要になったら手動で削除する必要があります。Amazon Data Lifecycle Manager は、スナップショットを管理しなくなります。
- ポリシーによって作成されたスナップショットを手動で削除し、ポリシーの保持しきい値に達したときにそのスナップショットがごみ箱にある場合、Amazon Data Lifecycle Manager はスナップショットを削除しません。Amazon Data Lifecycle Manager は、スナップショットがごみ箱に保存されている間は、スナップショットを管理しません。

  ポリシーの保持しきい値に達する前にスナップショットがごみ箱から復元された場合、Amazon Data Lifecycle Manager は、ポリシーの保持しきい値に達したときにスナップショットを削除します。

  ポリシーの保持しきい値に達した後にスナップショットがごみ箱から復元された場合、Amazon Data Lifecycle Manager はそのスナップショットを削除しません。スナップショットが不要になった場合は、手動で削除する必要があります。

AWS バックアップに関する考慮事項

- AWS Backup が保持ルールに一致するスナップショットを削除すると、そのスナップショットはごみ箱によって自動的に保持されます。

アーカイブされたスナップショットに関する考慮事項

- ごみ箱の保存ルールは、アーカイブストレージ階層のアーカイブされたスナップショットにも適用されます。ごみ箱の保持ルールに一致するアーカイブスナップショットを削除すると、アーカイブされたスナップショットは、保持ルールで定義されている保持期間中、ごみ箱に保持されます。

  アーカイブされたスナップショットは、ごみ箱に入っている間、アーカイブされたスナップショットの料金で請求されます。

  保持ルールによってアーカイブされたスナップショットがごみ箱から最低期間である 90 日前に削除された場合、残りの日分の料金が請求されます。詳細については、「アーカイブされたスナップショットの料金と請求」を参照してください。

  ごみ箱にアーカイブされたスナップショットを使用するには、まずそのスナップショットをごみ箱から復元し、次にアーカイブ階層から標準階層に復元する必要があります。

クォータ

ごみ箱には、以下のクォータが適用されます。

クォータ	デフォルトのクォータ			
リージョンあたりの保持ルール数	250			
保持ルールごとにキーと値のペアにタグ付けする	50			

関連サービス

ごみ箱は以下のサービスと連携します。

- [AWS CloudTrail] — ごみ箱で発生したイベントを記録できます。詳細については、を使用してごみ箱をモニタリングする AWS CloudTrailを参照してください。

料金

ごみ箱および保持ルールの使用には、追加料金はかかりません。詳細については、Amazon EBS の料金表を参照してください。

- Amazon EBS ボリューム — ごみ箱内のボリュームは、アカウントの通常のボリュームと同じ料金で請求されます。
- Amazon EBS スナップショット — ごみ箱内のスナップショットは、アカウント内の通常のスナップショットと同じ料金で請求されます。
- EBS-backed AMI — ごみ箱内の AMI には追加料金は発生しません。

> **Note**
>
> 一部のリソースは、保持期間が終了し、完全に削除されてからしばらくの間、ごみ箱コンソールまたは AWS CLI および API 出力に表示される場合があります。これらのリソースの料金は請求されません。請求は、保持期間が終了するとすぐに停止します。

次の AWS 生成されたコスト配分タグは、コストの追跡と配分の目的で使用できます AWS Billing and Cost Management。

- キー: `aws:recycle-bin:resource-in-bin`
- 値: `true`

詳細については、「AWS Billing and Cost Management ユーザーガイド」の「Using AWS-generated tags」を参照してください。

IAM でごみ箱へのアクセスを制御

デフォルトでは、ユーザーには、ごみ箱、保持ルール、またはごみ箱にあるリソースを操作する許可はありません。ユーザーがこれらのリソースを利用するには、特定のリソースと API アクションを使用する許可を付与する IAM ポリシーを作成する必要があります。ポリシーを作成したら、ユーザー、グループ、ロールにアクセス許可を追加する必要があります。

トピック

- ごみ箱および保持ルールを操作するための許可

- ごみ箱内のリソースを操作するための許可
- [Condition keys for Recycle Bin] (ごみ箱の条件キー)

ごみ箱および保持ルールを操作するための許可

ごみ箱と保持ルールを使用するには、次の許可をユーザーに付与する必要があります。

- rbin:CreateRule
- rbin:UpdateRule
- rbin:GetRule
- rbin:ListRules
- rbin:DeleteRule
- rbin:TagResource
- rbin:UntagResource
- rbin:ListTagsForResource
- rbin:LockRule
- rbin:UnlockRule

ごみ箱コンソールを使用するには、ユーザーに tag:GetResources 許可が必要です。

以下は、コンソールユーザーの tag:GetResources 許可を含む IAM ポリシーの例です。一部の許可が不要な場合は、ポリシーから削除できます。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "rbin:CreateRule",
                "rbin:UpdateRule",
                "rbin:GetRule",
                "rbin:ListRules",
                "rbin:DeleteRule",
```

```
                "rbin:TagResource",
                "rbin:UntagResource",
                "rbin:ListTagsForResource",
                "rbin:LockRule",
                "rbin:UnlockRule",
                "tag:GetResources"
            ],
            "Resource": "*"
        }
    ]
}
```

アクセス許可を付与するにはユーザー、グループ、またはロールにアクセス許可を追加します。

- 以下のユーザーとグループ AWS IAM アイデンティティセンター:

  アクセス許可セットを作成します。「AWS IAM アイデンティティセンター ユーザーガイド」の「アクセス許可セットを作成する」の手順に従ってください。
- IAM 内で、ID プロバイダーによって管理されているユーザー:

  ID フェデレーションのロールを作成します。詳細については IAM ユーザーガイド の サードパーティー ID プロバイダー (フェデレーション) 用のロールを作成する を参照してください。
- IAM ユーザー:
  - ユーザーが担当できるロールを作成します。手順については IAM ユーザーガイド の IAM ユーザーのロールの作成 を参照してください。
  - (お奨めできない方法) ポリシーをユーザーに直接アタッチするか、ユーザーをユーザーグループに追加します。「IAM ユーザーガイド」の「ユーザー (コンソール) へのアクセス許可の追加」の指示に従います。

ごみ箱内のリソースを操作するための許可

ごみ箱内のリソースを操作するために必要な IAM 許可の詳細については、次を参照してください。

- ごみ箱内のボリュームを操作するためのアクセス許可
- ごみ箱のスナップショットを操作するための権限
- ごみ箱内の AMI を操作するための許可

[Condition keys for Recycle Bin] (ごみ箱の条件キー)

ごみ箱は、IAM ポリシーのCondition要素に使用できる次の条件キーを定義し、ポリシーステートメントが適用される条件を制御します。詳細については、[IAM User Guide] (IAM ユーザーガイド) の [IAM JSON policy elements: Condition] (IAM JSON ポリシー要素：条件) を参照してください。

トピック

- rbin:Request/ResourceType 条件キー
- rbin:Attribute/ResourceType 条件キー

rbin:Request/ResourceType 条件キー

このrbin:Request/ResourceType条件キーを使用して、ResourceTypeリクエストパラメータで指定された値に基づいて[CreateRule]と[ListRules]リクエストのアクセスをフィルタリングするために使用することができます。

例 1 - CreateRule

次のサンプルの IAM ポリシーは、ResourceTypeリクエストパラメーターに指定された値がEBS_SNAPSHOTまたはEC2_IMAGEである場合のみ IAM プリンシパルに [CreateRule] リクエストを行うことを許可します。これにより、プリンシパルはスナップショットと AMI に対してのみ新しい保存ルールを作成できます。

JSON

```
{
    "Version":"2012-10-17",
    "Statement" : [
        {
            "Effect" : "Allow",
            "Action" :[
                "rbin:CreateRule"
            ],
            "Resource" : "*",
            "Condition" : {
                "StringEquals" : {
                    "rbin:Request/ResourceType" : ["EBS_SNAPSHOT", "EC2_IMAGE"]
                }
            }
```

```
        }
    ]
}
```

例 2 - ListRules

次のサンプル IAM ポリシーは、ResourceTypeリクエストパラメーターに指定した値がEBS_SNAPSHOTの場合にのみ、IAM プリンシパルが ListRules に要求を行うことを許可します。これにより、プリンシパルはスナップショットの保存ルールのみを一覧表示でき、他のリソースタイプの保存ルールを一覧表示できなくなります。

JSON

```
{
    "Version":"2012-10-17",
    "Statement" : [
        {
            "Effect" : "Allow",
            "Action" :[
                "rbin:ListRules"
            ],
            "Resource" : "*",
            "Condition" : {
                "StringEquals" : {
                    "rbin:Request/ResourceType" : "EBS_SNAPSHOT"
                }
            }
        }
    ]
}
```

rbin:Attribute/ResourceType 条件キー

rbin:Attribute/ResourceType 条件キーを使用し、保持ルールの ResourceType 属性の値に基づいて DeleteRule、GetRule、UpdateRule、LockRule、UnlockRule、TagResource、UntagResource、ListTagsForResource リクエストへのアクセスをフィルタリングできます。

例 1 - UpdateRule

次のサンプル IAM ポリシーは、ResourceType要求されたリテンションルールの属性がEBS_SNAPSHOTまたはEC2_IMAGEの場合にのみ、IAM プリンシパルが UpdateRule に要求を行うことを許可します。これにより、プリンシパルはスナップショットと AMI の保持ルールのみを更新できます。

JSON

```
{
    "Version":"2012-10-17",
    "Statement" : [
        {
            "Effect" : "Allow",
            "Action" :[
                "rbin:UpdateRule"
            ],
            "Resource" : "*",
            "Condition" : {
                "StringEquals" : {
                    "rbin:Attribute/ResourceType" : ["EBS_SNAPSHOT", "EC2_IMAGE"]
                }
            }
        }
    ]
}
```

例 2 - DeleteRule

次のサンプル IAM ポリシーは、ResourceType要求されたリテンションルールの属性がEBS_SNAPSHOTの場合にのみ、IAM プリンシパルが DeleteRule に要求を行うことを許可します。これにより、プリンシパルはスナップショットの保存ルールのみを削除できます。

JSON

```
{
    "Version":"2012-10-17",
    "Statement" : [
        {
            "Effect" : "Allow",
            "Action" :[
                "rbin:DeleteRule"
```

```
            ],
            "Resource" : "*",
            "Condition" : {
                "StringEquals" : {
                    "rbin:Attribute/ResourceType" : "EBS_SNAPSHOT"
                }
            }
        }
    ]
}
```

ごみ箱の保持ルールを作成

保持ルールを作成するときは、次の必須パラメータを指定する必要があります。

- 保護するリソースタイプ (ボリューム、スナップショット、または AMIs)。
- 保持ルールのタイプ (タグレベルまたはリージョンレベル)。タグレベルのルールでは、特定のタグを持つリソースのみを保護します。リージョンレベルのルールでは、リージョン内のすべてのリソースを保護しますが、特定のタグを持つリソースを除外できます。
- 削除されたリソースを保持する保持期間。この期間が終了すると、リソースはごみ箱から完全に削除されます。サポートされている保持期間は次のとおりです。
  - EBS ボリューム: 1 ~ 7 日
  - EBS スナップショットと EBS-backed AMIs: 1 ~ 365 日

オプションで、それぞれ最大 255 文字のルール名と説明、およびルールの識別と整理に役立つタグを指定することもできます。名前、説明、タグには、個人を特定する情報、機密情報、または秘密情報を含めないことをお勧めします。

オプションで、作成時にリージョンレベルの保持ルールをロックすることもできます。作成時に保持ルールをロックする場合は、ロック解除の遅延期間 (7 ~ 30 日) も指定する必要があります。保持ルールは、意図的にロックしない限り、デフォルトでロック解除されたままになります。

> **Note**
>
> 保持ルールは、作成されたリージョンでのみ機能します。他のリージョンでごみ箱を使用する場合は、そのリージョンに追加の保持ルールを作成する必要があります。

ごみ箱保持ルールは、次のいずれかの方法で作成できます。

Recycle Bin console

タグレベルの保持ルールを作成するには

1. ごみ箱コンソールを https://console.aws.amazon.com/rbin/home/ で開きます。
2. ナビゲーションペインで、[Retention rules] (保持ルール)、[Create retention rule] (保持ルールの作成) の順に選択します。
3. (オプション) [Retention rule name] (保持ルール名) に、保持ルールのわかりやすい名前を入力します。
4. (オプション) [Retention rule descripion] (保持ルールの説明) に、保持ルールの簡単な説明を入力します。
5. [リソースタイプ] で、保護する保持ルールのリソースタイプを選択します。保持ルールは、このタイプのリソースのみをごみ箱に保持します。
6. [保持するリソースを選択] で、[特定のタグが付いたリソースを保持] を選択します。
7. [リソースタグ] で、ごみ箱に保持するリソースを識別するために使用するタグキーと値のペアを入力します。保持ルールでは、指定されたタグが少なくとも 1 つ含まれている指定タイプのリソースのみが保持されます。
8. [保持期間] に、削除したリソースをごみ箱に保持する日数を入力します。
9. [Create retention rule] (保持ルールの作成) を選択します。

リージョンレベルの保存ルールを作成するには

1. ごみ箱コンソールを https://console.aws.amazon.com/rbin/home/ で開きます。
2. ナビゲーションペインで、[Retention rules] (保持ルール)、[Create retention rule] (保持ルールの作成) の順に選択します。
3. (オプション) [Retention rule name] (保持ルール名) に、保持ルールのわかりやすい名前を入力します。
4. (オプション) [Retention rule descripion] (保持ルールの説明) に、保持ルールの簡単な説明を入力します。
5. [リソースタイプ] で、保護する保持ルールのリソースタイプを選択します。保持ルールは、このタイプのリソースのみをごみ箱に保持します。
6. [保持するリソースを選択] で、[すべてのリソースを保持] を選択します。

7. (オプション) 特定のタグを持つリソースを除外するには、[除外タグ] で、リソースを識別して除外するために使用するタグキーと値のペアを最大 5 つ入力します。これらのタグのいずれかを持つリソースは、保持ルールによって無視されます。
8. [保持期間] に、削除したリソースをごみ箱に保持する日数を入力します。
9. (オプション) 保持ルールをロックするには、[ルールのロックの設定] で [ロック] を選択し、[ロック解除の遅延期間] でロック解除の遅延期間を日単位で指定します。保持ルールを変更または削除することはできません。ルールを変更または削除するには、まずルールをロック解除してから、ロック解除の遅延期間が終了するまで待つ必要があります。詳細については、ごみ箱の保持ルールをロックして、更新または削除されないようにするを参照してください。

   保持ルールをロック解除したままにするには、[Rule lock settings] (ルールのロックの設定) で [Unlock] (ロック解除) を選択したままにします。ロック解除された保持ルールは、いつでも変更または削除できます。

   > **Note**
   >
   > 除外タグを持つリージョンレベルの保持ルールはロックできません。

10. [Create retention rule] (保持ルールの作成) を選択します。

AWS CLI

保持ルールを作成するには

create-rule AWS CLI コマンドを使用します。[--retention-period] に、ごみ箱に削除されたスナップショットを保持する日数を指定します。の場合は--resource-type、ボリュームEBS_VOLUME、EBS_SNAPSHOTスナップショット、または AMIs EC2_IMAGEに を指定します。タグレベルの保持ルールを作成するには、 に--resource-tags、保持するリソースを識別するために使用するタグを指定します。リージョンレベルの保持ルールを作成するには、--resource-tags を省略し、オプションで --exclude-resource-tags を指定して、特定のタグを持つリソースを除外します。リージョンレベルの保持ルールをロックするには、--lock-configuration を含めて、ロック解除の遅延期間を日単位指定します。

```
aws rbin create-rule \
--retention-period RetentionPeriodValue=number_of_days,RetentionPeriodUnit=DAYS \
--resource-type EBS_VOLUME|EBS_SNAPSHOT|EC2_IMAGE \
--description "rule_description" \
```

```
--lock-configuration
 'UnlockDelay={UnlockDelayUnit=DAYS,UnlockDelayValue=unlock_delay_in_days}' \
--resource-tags ResourceTagKey=tag_key,ResourceTagValue=tag_value \
--exclude-resource-tags ResourceTagKey=tag_key,ResourceTagValue=tag_value
```

例 1

次のコマンド例では、すべてのスナップショットを 7 日間保持するリージョンレベルのロック削除された保持ルールを作成します。

```
aws rbin create-rule \
--retention-period RetentionPeriodValue=7,RetentionPeriodUnit=DAYS \
--resource-type EBS_SNAPSHOT \
--description "Match all snapshots"
```

例 2

次のコマンド例では、purpose=production でタグ付けされた削除済みのスナップショットを 7 日間保持するタグレベルのルールを作成します。

```
aws rbin create-rule \
--retention-period RetentionPeriodValue=7,RetentionPeriodUnit=DAYS \
--resource-type EBS_SNAPSHOT \
--description "Match snapshots with a specific tag" \
--resource-tags ResourceTagKey=purpose,ResourceTagValue=production
```

例 3

次のコマンド例では、すべてのスナップショットを 7 日間保持するリージョンレベルのロックされた保持ルールを作成します。保持ルールは 7 日間のロック解除の遅延期間でロックされます。

```
aws rbin create-rule \
--retention-period RetentionPeriodValue=7,RetentionPeriodUnit=DAYS \
--resource-type EBS_SNAPSHOT \
--description "Match all snapshots" \
--lock-configuration 'UnlockDelay={UnlockDelayUnit=DAYS,UnlockDelayValue=7}'
```

例 4

次のコマンド例では、purpose:testing でタグ付けされたスナップショットを除くすべての削除済みスナップショットを 7 日間保持する、リージョンレベルのロック解除された保持ルールを作成します。

```
aws rbin create-rule \
--retention-period RetentionPeriodValue=7,RetentionPeriodUnit=DAYS \
--resource-type EBS_SNAPSHOT \
--description "Match only production snapshots" \
--exclude-resource-tags ResourceTagKey=purpose,ResourceTagValue=testing
```

既存のごみ箱の保持ルールを更新

ロック解除された保持ルールの説明、リソースタグ、保持期間は、作成後にいつでも更新できます。保持ルールのリソースタイプやロック解除の遅延期間を、保持ルールがロック解除されていても、更新することはできません。

ロックされた保持ルールは、どのような方法でも更新できません。ロックされた保持ルールを変更する必要がある場合は、まずロックを解除し、ロック解除の遅延期間が終了するまで待つ必要があります。

ロックされた保持ルールのロック解除の遅延期間を変更する必要がある場合は、保持ルールをロック解除し、現在のロック解除の遅延期間が終了するまで待つ必要があります。ロック解除の遅延期間が終了したら、保持ルールを再ロックし、新しいロック解除の遅延期間を指定する必要があります。

> **Note**
>
> 保持ルールの説明には、個人を特定する情報、機密情報、または機密情報を含めないことをお勧めします。

保持ルールを更新すると、その変更は保持される新しいリソースにのみ適用されます。この変更は、ごみ箱に移動済みのリソースには影響しません。例えば、保持ルールの保持期間を更新すると、更新後に削除されたスナップショットのみが新しい保持期間、保持されます。更新前にごみ箱に送られたスナップショットは、以前の (古い) 保持期間にわたって保持されます。

保持ルールの更新は、次のいずれかの方法で行うことができます。

Recycle Bin console

保持ルールを更新するには

1. ごみ箱コンソール (https://console.aws.amazon.com/rbin/home/) を開きます。

2. ナビゲーションペインで、[Retention rules] (保持ルール) を選択します。
3. グリッドで、更新する保持ルールを選択し、[Actions] (アクション)、[Edit retention rule] (保持ルールの編集) の順にクリックします。
4. [Rule details] (ルールの詳細) セクションで、[Retention rule name] (保持ルール名) そして [Retention rule description] (保持ルールの説明) を必要に応じて更新します。
5. [ルール設定] セクションで、[リソースタイプ]、[照合するリソースタグ]、[保持期間] を必要に応じて更新します。
6. [Tags] (タグ) セクションで、必要に応じて保持ルールタグを追加または削除します。
7. [Save retention rule] (保持ルールの保存) を選択します。

AWS CLI

保持ルールを更新するには

update-rule AWS CLI コマンドを使用します。の場合は--identifier、更新する保持ルールの ID を指定します。 の場合は --resource-types、ボリュームEBS_VOLUMEの場合は 、スナップショットEBS_SNAPSHOTの場合は 、AMI EC2_IMAGE の場合は を指定します。 AMIs

```
aws rbin update-rule \
--identifier rule_ID \
--retention-period RetentionPeriodValue=number_of_days,RetentionPeriodUnit=DAYS \
--resource-type EBS_VOLUME|EBS_SNAPSHOT|EC2_IMAGE \
--description "rule_description"
```

例

次の例では、保持ルール 6lsJ2Fa9nh9 を更新して、すべてのスナップショットを 7 日間保持するようにし、その説明を更新しています。

```
aws rbin update-rule \
--identifier 6lsJ2Fa9nh9 \
--retention-period RetentionPeriodValue=7,RetentionPeriodUnit=DAYS \
--resource-type EBS_SNAPSHOT \
--description "Retain for three weeks"
```

ごみ箱の保持ルールを口ックして、更新または削除されないようにする

ごみ箱を使用すると、リージョンレベルの保持ルールをいつでも口ックできます。

口ックされた保持ルールは、必要な IAM 許可を持つユーザーであっても変更または削除できません。保持ルールを口ックすることで、偶発的な、または悪意のある変更や削除から保護できます。

保持ルールを口ックするには、口ック解除の遅延期間を指定する必要があります。これは、保持ルールを口ック解除してから変更または削除できるようになるまで待つ必要がある期間です。口ック解除の遅延期間中は、保持ルールを変更または削除することはできません。保持ルールの変更または削除は、口ック解除の遅延期間の終了後にのみ行えます。

保持ルールのロック後は、口ック解除の遅延期間を変更できません。アカウントの権限が侵害された場合、口ック解除の遅延期間を設けることで、セキュリティ上の脅威を検出して対応するための追加の時間を確保できます。この期間は、セキュリティ違反を特定して対応するのにかかる時間よりも長くする必要があります。過去のセキュリティインシデントと、アカウント侵害の特定と是正に必要な時間を確認することで、適切な期間を設定することができます。

保持ルールの口ック状態が変更された場合に通知されるように、Amazon EventBridge ルールを使用することをお勧めします。詳細については、「Amazon EventBridge を使用してごみ箱をモニタリングする」を参照してください。

考慮事項

- タグレベルの保持ルール、または除外タグを持つリージョンレベルの保持ルールを口ックすることはできません。
- 保持ルールの口ックはいつでも解除できます。
- 口ック解除の遅延期間は 7〜30 日でなければなりません。
- 保持ルールはロック解除の遅延期間中に再口ックできます。保持ルールを再口ックすると、口ック解除の遅延期間がリセットされます。

リージョンレベルの保持ルールは、次のいずれかの方法で口ックできます。

Recycle Bin console

保持ルールをロックするには

1. ごみ箱コンソールを https://console.aws.amazon.com/rbin/home/ で開きます。
2. ナビゲーションパネルで、[Retention rules] (保持ルール) を選択します。
3. グリッドでロックする保持ルールを選択し、[Actions] (アクション)、[Edit retention rule lock] (保持ルールロックの編集) の順に選択します。
4. [Edit retention rule lock] (保持ルールロックの編集) 画面で [Lock] (ロック) を選択し、[Unlock delay period] (ロック解除の遅延期間) でロック解除の遅延期間を日単位で指定します。
5. [I acknowledge that locking the retention rule will prevent it from being modified or deleted] (保持ルールをロックすると変更や削除ができなくなることを確認) チェックボックスをオンにし、[Save] (保存) を選択します。

AWS CLI

ロック解除された保持ルールをロックするには

ロックルール AWS CLI コマンドを使用します。`--identifier` については、ロックする保持ルールの ID を指定します。`--lock-configuration` については、ロック解除の遅延期間を日単位で指定します。

```
aws rbin lock-rule \
--identifier rule_ID \
--lock-configuration
 'UnlockDelay={UnlockDelayUnit=DAYS,UnlockDelayValue=number_of_days}'
```

例

次のコマンド例では、`6lsJ2Fa9nh9` 保持ルールをロックし、ロック解除の遅延期間を 15 日間に設定します。

```
aws rbin lock-rule \
--identifier 6lsJ2Fa9nh9 \
--lock-configuration 'UnlockDelay={UnlockDelayUnit=DAYS,UnlockDelayValue=15}'
```

ごみ箱の保持ルールをロック解除して、更新または削除できるようにする

ロックされた保持ルールの変更または削除はできません。ロックされた保持ルールを変更する必要がある場合は、まずロックを解除する必要があります。保持ルールをロック解除したら、変更または削除する前にロック解除の遅延期間が終了するのを待つ必要があります。ロック解除の遅延期間中は、保持ルールを変更または削除することはできません。

ロック解除された保持ルールは、必要な IAM 許可を持つユーザーがいつでも変更および削除できます。保持ルールをロック解除したままにすると、偶発的または悪意のある変更や削除にさらされる可能性があります。

考慮事項

- 保持ルールはロック解除の遅延期間中に再ロックできます。
- ロック解除の遅延期間が過ぎた後で保持ルールを再ロックできます。
- ロック解除の遅延期間をバイパスすることはできません。
- 最初のロック後に、ロック解除の遅延時間を変更することはできません。

保持ルールのロック状態が変更された場合に通知されるように、Amazon EventBridge ルールを使用することをお勧めします。詳細については、「Amazon EventBridge を使用してごみ箱をモニタリングする」を参照してください。

リージョンレベルのロックされた保持ルールは、次のいずれかの方法でロック解除できます。

Recycle Bin console

保持ルールをロック解除するには

1. ごみ箱コンソールを https://console.aws.amazon.com/rbin/home/ で開きます。
2. ナビゲーションパネルで、[Retention rules] (保持ルール) を選択します。
3. グリッドでロック解除する保持ルールを選択し、[Actions] (アクション)、[Edit retention rule lock] (保持ルールロックの編集) の順に選択します。
4. [Edit retention rule lock] (保持ルールロックの編集) 画面で、[Unlock] (ロック解除) を選択し、[Save] (保存) を選択します。

AWS CLI

ロックされた保持ルールをロック解除するには

unlock-rule AWS CLI コマンドを使用します。--identifier で、ロック解除する保持ルールの ID を指定します。

```
aws rbin unlock-rule \
--identifier rule_ID
```

例

次のコマンド例では、保持ルール 6lsJ2Fa9nh9 をロック解除します。

```
aws rbin unlock-rule \
--identifier 6lsJ2Fa9nh9
```

ごみ箱の保持ルールをタグ

保持ルールにカスタムタグを割り当てて、目的、所有者、環境など、さまざまな方法で分類できます。これにより、割り当てたカスタムタグに基づいて特定の保持ルールを効率的に見つけることができます。

保持ルールにタグを割り当てるには、次のいずれかの方法を使用します。

Recycle Bin console

保持ルールにタグ付けするには

1. ごみ箱コンソール (https://console.aws.amazon.com/rbin/home/) を開きます。
2. ナビゲーションペインで、[Retention rules] (保持ルール) を選択します。
3. タグ付けする保持ルールを選択し、[Tags] (タグ) タブで、[Manage tags] (タグの管理) を選択します。
4. [タグを追加] を選択します。[Key] (キー) に、タグキーを入力します。[Value] (値) に、タグの値を入力します。
5. [Save] (保存) を選択します。

AWS CLI

保持ルールにタグ付けするには

tag-resource AWS CLI コマンドを使用します。--resource-arn で、タグ付けする保持ルールの Amazon リソースネーム (ARN) を指定し、--tags で、タグのキーと値のペアを指定します。

```
aws rbin tag-resource \
--resource-arn retention_rule_arn \
--tags key=tag_key,value=tag_value
```

例

次のコマンド例では、保持ルール arn:aws:rbin:us-east-1:123456789012:rule/nOoSBBtItF3 に purpose=production をタグ付けします。

```
aws rbin tag-resource \
--resource-arn arn:aws:rbin:us-east-1:123456789012:rule/nOoSBBtItF3 \
--tags key=purpose,value=production
```

保持ルールのタグを表示する

保持ルールに割り当てられたタグは、次のいずれかの方法で表示できます。

Recycle Bin console

保持ルールのタグを表示するには

1. ごみ箱コンソール (https://console.aws.amazon.com/rbin/home/) を開きます。
2. ナビゲーションペインで、[Retention rules] (保持ルール) を選択します。
3. タグを表示する保持ルールを選択し、[Tags] (タグ) タブを選択します。

AWS CLI

保持ルールに割り当てられたタグを表示するには

list-tags-for-resource AWS CLI コマンドを使用します。--resource-arn で、保持ルールの ARN を指定します。

```
aws rbin list-tags-for-resource \
--resource-arn retention_rule_arn
```

例

次のコマンド例では、保持ルール arn:aws:rbin:us-east-1:123456789012:rule/nOoSBBtItF3 のタグを一覧表示します。

```
aws rbin list-tags-for-resource \
--resource-arn arn:aws:rbin:us-east-1:123456789012:rule/nOoSBBtItF3
```

保持ルールからタグを削除する

イベント通知のタグは、次のいずれかの方法で削除することができます。

Recycle Bin console

保持ルールからタグを削除するには

1. ごみ箱コンソール (https://console.aws.amazon.com/rbin/home/) を開きます。
2. ナビゲーションペインで、[Retention rules] (保持ルール) を選択します。
3. タグを削除する保持ルールを選択し、[Tags] (タグ) タブで、[Manage tags] (タグの管理) を選択します。
4. 削除するタグの横にある [Remove] (削除) を選択します。
5. [Save] (保存) を選択します。

AWS CLI

保持ルールからタグを削除するには

untag-resource AWS CLI コマンドを使用します。--resource-arn で、保持ルールの ARN を指定します。--tagkeys で、削除するタグのタグキーを指定します。

```
aws rbin untag-resource \
--resource-arn retention_rule_arn \
--tagkeys tag_key
```

例

次のコマンド例では、キーが purpose のタグを保持ルール arn:aws:rbin:us-east-1:123456789012:rule/nOoSBBtItF3 から削除します。

```
aws rbin untag-resource \
--resource-arn arn:aws:rbin:us-east-1:123456789012:rule/nOoSBBtItF3 \
--tagkeys purpose
```

ごみ箱の保持ルールを削除することで、ごみ箱がリソース保持するのを防止

保持ルールはいつでも削除できます。保持ルールを削除すると、削除後にゴミ箱で新しいリソースが保持されなくなります。保持ルールが削除される前にごみ箱に移動されたリソースは、保持ルールで定義されている保持期間に従って、引き続きごみ箱に保持されます。期間が終了すると、リソースはごみ箱から完全に削除されます。

次のいずれかの方法を使用して、保持ルールを削除できます。

Recycle Bin console

保持ルールを削除するには

1. ごみ箱コンソール (https://console.aws.amazon.com/rbin/home/) を開きます。
2. ナビゲーションペインで、[Retention rules] (保持ルール) を選択します。
3. グリッドで削除する保持ルールを選択し、[Actions] (アクション)、[Delete retention rule] (保持ルールの削除) の順に選択します。
4. プロンプトが表示されたら、確認メッセージを入力し、[Delete retention rule] (保持ルールの削除) を選択します。

AWS CLI

保持ルールを削除するには

delete-rule AWS CLI コマンドを使用します。--identifier で、削除するリテンションルールの ID を指定します。

```
aws rbin delete-rule --identifier rule_ID
```

例

次のコマンド例では、保持ルール 6lsJ2Fa9nh9 を削除します。

```
aws rbin delete-rule --identifier 6lsJ2Fa9nh9
```

ごみ箱から削除されたスナップショットを復元

このトピックでは、ごみ箱から Amazon EBS スナップショットを復元する方法について説明します。

トピック

- ごみ箱のスナップショットを操作するための権限
- ごみ箱のスナップショットを表示する
- ごみ箱からスナップショットを復元する

ごみ箱のスナップショットを操作するための権限

デフォルトでは、ユーザーには、ごみ箱にあるスナップショットを操作する許可はありません。ユーザーがこれらのリソースを利用するには、特定のリソースと API アクションを使用する許可を付与する IAM ポリシーを作成する必要があります。ポリシーを作成したら、ユーザー、グループ、ロールにアクセス許可を追加する必要があります。

ごみ箱にあるスナップショットを表示および復旧するには、ユーザーに次の許可が必要です。

- `ec2:ListSnapshotsInRecycleBin`
- `ec2:RestoreSnapshotFromRecycleBin`

ごみ箱内のスナップショットのタグを管理するには、次の追加の許可をユーザーに付与する必要があります。

- `ec2:CreateTags`
- `ec2:DeleteTags`

ごみ箱コンソールを使用するには、ユーザーに `ec2:DescribeTags` 許可が必要です。

IAM ポリシーの例を次に示します。これには、コンソールユーザーの `ec2:DescribeTags` 許可と、タグを管理するための `ec2:CreateTags` および `ec2:DeleteTags` の許可が含まれます。許可が不要な場合は、ポリシーから削除できます。

アクセス許可を付与するにはユーザー、グループ、またはロールにアクセス許可を追加します。

- 以下のユーザーとグループ AWS IAM アイデンティティセンター:

  アクセス許可セットを作成します。「AWS IAM アイデンティティセンター ユーザーガイド」の「アクセス許可セットを作成する」の手順に従ってください。
- IAM 内で、ID プロバイダーによって管理されているユーザー:

  ID フェデレーションのロールを作成します。詳細については IAM ユーザーガイド の サードパーティー ID プロバイダー (フェデレーション) 用のロールを作成する を参照してください。
- IAM ユーザー:
  - ユーザーが担当できるロールを作成します。手順については IAM ユーザーガイド の IAM ユーザーのロールの作成 を参照してください。
  - (お奨めできない方法) ポリシーをユーザーに直接アタッチするか、ユーザーをユーザーグループに追加します。「IAM ユーザーガイド」の「ユーザー (コンソール) へのアクセス許可の追加」の指示に従います。

ごみ箱を使用するために必要な許可の詳細については、「ごみ箱および保持ルールを操作するための許可」を参照してください。

ごみ箱のスナップショットを表示する

スナップショットがごみ箱に入っている間は、次のような限定された情報を表示できます。

- スナップショットの ID。
- スナップショットの説明。
- スナップショットを作成したボリュームの ID。
- スナップショットが削除され、ごみ箱に入った日時。
- 保持期間の有効期限が切れる日時。この時点で、スナップショットはごみ箱から完全に削除されます。

ごみ箱のスナップショットは、次のいずれかの方法を使用して表示できます。

Recycle Bin console

コンソールを使用して、ごみ箱にあるスナップショットを表示するには

1. ごみ箱コンソール (https://console.aws.amazon.com/rbin/home/) を開きます。
2. ナビゲーションペインで、[Recycle Bin] (ごみ箱) を選択します。
3. グリッドには、現在ごみ箱にあるすべてのスナップショットがリストされます。特定のスナップショットの詳細を表示するには、グリッドで選択し、[Actions] (アクション)、[View details] (詳細を表示) の順にクリックします。

AWS CLI

を使用してごみ箱のスナップショットを表示するには AWS CLI

list-snapshots-in-recycle-bin AWS CLI コマンドを使用します。`--snapshot-id` オプションを使用して、特定のスナップショットを表示します。または、`--snapshot-id` オプションを省略して、ごみ箱内のすべてのスナップショットを表示します。

```
aws ec2 list-snapshots-in-recycle-bin --snapshot-id snapshot_id
```

たとえば、次のコマンドは、ごみ箱にあるスナップショット `snap-01234567890abcdef` に関する情報を提供します。

```
aws ec2 list-snapshots-in-recycle-bin --snapshot-id snap-01234567890abcdef
```

出力の例:

```
{
    "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"
        }
    ]
}
```

ごみ箱からスナップショットを復元する

スナップショットがごみ箱に入っている間は、いかなる方法でも使用することはできません。スナップショットを使用するには、まずスナップショットを復元する必要があります。ごみ箱からスナップショットを復元すると、そのスナップショットはすぐに使用でき、ごみ箱から削除されます。復元されたスナップショットは、アカウント内の他のスナップショットと同じ方法で使用できます。

次のいずれかの方法を使用して、ごみ箱からスナップショットを復元できます。

Recycle Bin console

コンソールを使用してごみ箱からスナップショットから復元する

1. ごみ箱コンソール (https://console.aws.amazon.com/rbin/home/) を開きます。
2. ナビゲーションペインで、[Recycle Bin] (ごみ箱) を選択します。
3. グリッドには、現在ごみ箱にあるすべてのスナップショットがリストされます。復元するスナップショットを選択し、[Recover] (復元) を選択します。
4. プロンプトが表示されたら、[Recover] (復元) を選択します。

AWS CLI

を使用して、削除されたスナップショットをごみ箱から復元するには AWS CLI

restore-snapshot-from-recycle-bin AWS CLI コマンドを使用します。`--snapshot-id` に、復元するスナップショットの ID を指定します。

```
aws ec2 restore-snapshot-from-recycle-bin --snapshot-id snapshot_id
```

例えば次のコマンドでは、スナップショット を `snap-01234567890abcdef` をごみ箱から復元します。

```
aws ec2 restore-snapshot-from-recycle-bin --snapshot-id snap-01234567890abcdef
```

出力の例:

```
{
    "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
}
```


削除したボリュームをごみ箱から復旧する

このトピックでは、Amazon EBS ボリュームをごみ箱から復旧する方法について説明します。

トピック

- ごみ箱内のボリュームを操作するためのアクセス許可
- ごみ箱のボリュームを表示する
- ごみ箱からボリュームを復元する

ごみ箱内のボリュームを操作するためのアクセス許可

デフォルトでは、ユーザーにはごみ箱にあるボリュームを操作するアクセス許可がありません。ユーザーがこれらのリソースを利用するには、特定のリソースと API アクションを使用する許可を付与する IAM ポリシーを作成する必要があります。ポリシーを作成したら、ユーザー、グループ、ロールにアクセス許可を追加する必要があります。

ごみ箱にあるボリュームを表示および復旧するには、ユーザーに次のアクセス許可が必要です。

- `ec2:ListVolumesInRecycleBin`
- `ec2:RestoreVolumeFromRecycleBin`

ごみ箱内のボリュームのタグを管理するには、次の追加のアクセス許可が必要です。

- `ec2:CreateTags`
- `ec2:DeleteTags`

ごみ箱コンソールを使用するには、ユーザーに `ec2:DescribeTags` 許可が必要です。

IAM ポリシーの例を次に示します。これには、コンソールユーザーの ec2:DescribeTags 許可と、タグを管理するための ec2:CreateTags および ec2:DeleteTags の許可が含まれます。許可が不要な場合は、ポリシーから削除できます。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowRecycleBinVolumeOperations",
      "Effect": "Allow",
      "Action": [
        "ec2:ListVolumesInRecycleBin",
        "ec2:RestoreVolumeFromRecycleBin"
      ],
      "Resource": "arn:aws:ec2:*:123456789012:volume/*"
    },
    {
      "Sid": "AllowVolumeTagOperations",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateTags",
        "ec2:DeleteTags",
        "ec2:DescribeTags"
      ],
      "Resource": "arn:aws:ec2:*:123456789012:volume/*"
    }
  ]
}
```

アクセス許可を付与するにはユーザー、グループ、またはロールにアクセス許可を追加します。

- 以下のユーザーとグループ AWS IAM アイデンティティセンター:

  アクセス許可セットを作成します。「AWS IAM アイデンティティセンター ユーザーガイド」の「アクセス許可セットを作成する」の手順に従ってください。

- IAM 内で、ID プロバイダーによって管理されているユーザー:

  ID フェデレーションのロールを作成します。詳細については IAM ユーザーガイド の サードパーティー ID プロバイダー (フェデレーション) 用のロールを作成する を参照してください。

- IAM ユーザー:

- ユーザーが担当できるロールを作成します。手順については IAM ユーザーガイド の IAM ユーザーのロールの作成 を参照してください。
- (お奨めできない方法) ポリシーをユーザーに直接アタッチするか、ユーザーをユーザーグループに追加します。「IAM ユーザーガイド」の「ユーザー (コンソール) へのアクセス許可の追加」の指示に従います。

ごみ箱を使用するために必要な許可の詳細については、「ごみ箱および保持ルールを操作するための許可」を参照してください。

ごみ箱のボリュームを表示する

ボリュームがごみ箱にある間は、次のような限られた情報を表示できます。

- ボリュームの ID。
- ボリュームのサイズ。
- ボリュームタイプ。
- ボリュームが削除され、ごみ箱に入った日時。
- 保持期間の有効期限が切れる日時。この時点で、ボリュームはごみ箱から完全に削除されます。

ごみ箱内のボリュームは、次のいずれかの方法を使用して表示できます。

Recycle Bin console

コンソールを使用してごみ箱のボリュームを表示するには

1. ごみ箱コンソール (https://console.aws.amazon.com/rbin/home/) を開きます。
2. ナビゲーションペインで、[Recycle Bin] (ごみ箱) を選択します。
3. グリッドには、現在ごみ箱にあるすべてのボリュームが一覧表示されます。特定のボリュームの詳細を表示するには、グリッドでそのボリュームを選択し、アクション、詳細の表示を選択します。

AWS CLI

を使用してごみ箱のボリュームを表示するには AWS CLI

list-volumes-in-recycle-bin AWS CLI コマンドを使用します。特定のボリュームを表示する--volume-idオプションを含めます。または、 --volume-idオプションを省略して、ごみ箱内のすべてのボリュームを表示します。

```
aws ec2 list-volumes-in-recycle-bin --volume-id volume_id
```

たとえば、次のコマンドは、ごみ箱vol-01234567890abcdefのボリュームに関する情報を提供します。

```
aws ec2 list-volumes-in-recycle-bin --volume-id vol-01234567890abcdef
```

出力の例:

```
{
    "VolumeRecycleBinInfo": [
        {
            "VolumeId": "vol-01234567890abcdef",
            "RecycleBinEnterTime": "2021-12-01T13:00:00.000Z",
            "RecycleBinExitTime": "2021-12-08T13:00:00.000Z"
        }
    ]
}
```

ごみ箱からボリュームを復元する

ごみ箱に入っている間は、いかなる方法でもボリュームを使用することはできません。ボリュームを使用するには、まずボリュームを復元する必要があります。ごみ箱からボリュームを復元すると、そのボリュームはすぐに使用でき、ごみ箱から削除されます。復元されたボリュームは、アカウント内の他のボリュームを使用するのと同じ方法で使用できます。

次のいずれかの方法を使用して、ごみ箱からボリュームを復元できます。

Recycle Bin console

コンソールを使用してごみ箱からボリュームを復元するには

1. ごみ箱コンソール (https://console.aws.amazon.com/rbin/home/) を開きます。
2. ナビゲーションペインで、[Recycle Bin] (ごみ箱) を選択します。

3. グリッドには、現在ごみ箱にあるすべてのボリュームが一覧表示されます。復元するボリュームを選択し、リカバリを選択します。
4. プロンプトが表示されたら、[Recover] (復元) を選択します。

AWS CLI

を使用して、削除されたボリュームをごみ箱から復元するには AWS CLI

restore-volume-from-recycle-bin AWS CLI コマンドを使用します。には--volume-id、復元するボリュームの ID を指定します。

```
aws ec2 restore-volume-from-recycle-bin --volume-id volume_id
```

たとえば、次のコマンドはごみ箱vol-01234567890abcdefからボリュームを復元します。

```
aws ec2 restore-volume-from-recycle-bin --volume-id vol-01234567890abcdef
```

出力の例:

```
{
    "VolumeId": "vol-01234567890abcdef",
    "State": "available",
    "Size": 100,
    "VolumeType": "gp3",
    "AvailabilityZone": "us-east-1a",
    "CreateTime": "2021-12-01T13:00:00.000000+00:00",
    "Encrypted": false
}
```

削除された AMI をごみ箱から復元

このトピックでは、Amazon EBS-backed AMIs をごみ箱から復旧する方法について説明します。

トピック

- ごみ箱内の AMI を操作するための許可
- ごみ箱内の AMI を表示する
- ごみ箱から AMI を復元する

ごみ箱内の AMI を操作するための許可

デフォルトでは、ユーザーには、ごみ箱にある AMI を操作する許可はありません。ユーザーがこれらのリソースを利用するには、特定のリソースと API アクションを使用する許可を付与する IAM ポリシーを作成する必要があります。ポリシーを作成したら、ユーザー、グループ、ロールにアクセス許可を追加する必要があります。

ごみ箱にある AMI を表示および復旧するには、ユーザーに次の許可が必要です。

- `ec2:ListImagesInRecycleBin`
- `ec2:RestoreImageFromRecycleBin`

ごみ箱内の AMI のタグを管理するには、次の追加の許可をユーザーに付与する必要があります。

- `ec2:CreateTags`
- `ec2:DeleteTags`

ごみ箱コンソールを使用するには、ユーザーに `ec2:DescribeTags` 許可が必要です。

IAM ポリシーの例を次に示します。これには、コンソールユーザーの `ec2:DescribeTags` 許可と、タグを管理するための `ec2:CreateTags` および `ec2:DeleteTags` の許可が含まれます。許可が不要な場合は、ポリシーから削除できます。

アクセス許可を付与するにはユーザー、グループ、またはロールにアクセス許可を追加します。

- 以下のユーザーとグループ AWS IAM アイデンティティセンター:

  アクセス許可セットを作成します。「AWS IAM アイデンティティセンター ユーザーガイド」の「アクセス許可セットを作成する」の手順に従ってください。
- IAM 内で、ID プロバイダーによって管理されているユーザー:

  ID フェデレーションのロールを作成します。詳細については IAM ユーザーガイド の サードパーティー ID プロバイダー (フェデレーション) 用のロールを作成する を参照してください。
- IAM ユーザー:
  - ユーザーが担当できるロールを作成します。手順については IAM ユーザーガイド の IAM ユーザーのロールの作成 を参照してください。

- (お奨めできない方法) ポリシーをユーザーに直接アタッチするか、ユーザーをユーザーグループに追加します。「IAM ユーザーガイド」の「ユーザー (コンソール) へのアクセス許可の追加」の指示に従います。

ごみ箱を使用するために必要な許可の詳細については、「ごみ箱および保持ルールを操作するための許可」を参照してください。

ごみ箱内の AMI を表示する

AMI がごみ箱に入っている間は、次のような限定された情報を表示できます。

- AMI の名前、説明、および一意の ID。
- AMI が削除され、ごみ箱に入った日時。
- 保持期間の有効期限が切れる日時。この日時に AMI は完全に削除されます。

ごみ箱内の AMI は、次のいずれかの方法を使用して表示できます。

Recycle Bin console

コンソールを使用して、ごみ箱にある委任された AMI を表示するには

1. console.aws.amazon.com/rbin/home/ でごみ箱コンソールを開きます。
2. ナビゲーションペインで、[Recycle Bin] (ごみ箱) を選択します。
3. グリッドには、現在ごみ箱にあるすべてのリソースが一覧表示されます。特定の AMI の詳細を表示するには、グリッドで選択し、[Actions] (アクション)、[View details] (詳細を表示) の順に選択します。

AWS CLI

を使用して、削除された AMIs をごみ箱に表示するには AWS CLI

list-images-in-recycle-bin AWS CLI コマンドを使用します。特定の AMI を表示するには、`--image-id` オプションを含めて、表示する AMI の ID を指定します。1 つのリクエストで最大 20 個の ID を指定できます。

ごみ箱内のすべての AMI を表示するには、`--image-id` オプションを省略します。`--max-items` の値を指定しない場合、コマンドはデフォルトで 1 ページあたり 1,000 個のアイテムを返

します。詳細については、「Amazon EC2 API リファレンス」の「Pagination」(ページネーション) を参照してください。

```
aws ec2 list-images-in-recycle-bin --image-id ami_id
```

例えば、次のコマンドは、ごみ箱にある AMI ami-01234567890abcdef に関する情報を表示します。

```
aws ec2 list-images-in-recycle-bin --image-id ami-01234567890abcdef
```

出力の例:

```
{
    "Images": [
        {
            "ImageId": "ami-0f740206c743d75df",
            "Name": "My AL2 AMI",
            "Description": "My Amazon Linux 2 AMI",
            "RecycleBinEnterTime": "2021-11-26T21:04:50+00:00",
            "RecycleBinExitTime": "2022-03-06T21:04:50+00:00"
        }
    ]
}
```

Important

次のエラーが表示された場合は、 AWS CLI バージョンの更新が必要になる場合があります。詳細については、「コマンドが見つからないエラー」を参照してください。

```
aws.exe: error: argument operation: Invalid choice, valid choices are: ...
```

ごみ箱から AMI を復元する

AMI がごみ箱に入っている間は、いかなる方法でも使用できません。AMI を使用するには、まずスナップショットを復元する必要があります。ごみ箱から AMI を復元すると、その AMI はすぐに使用でき、ごみ箱からは削除されます。復元された AMI は、アカウント内の他の AMI と同じ方法で使用できます。

次のいずれかの方法を使用して、ごみ箱から AMI を復元できます。

Recycle Bin console

コンソールを使用してごみ箱から AMI を復元するには

1. console.aws.amazon.com/rbin/home/ でごみ箱コンソールを開きます。
2. ナビゲーションペインで、[Recycle Bin] (ごみ箱) を選択します。
3. グリッドには、現在ごみ箱にあるすべてのリソースが一覧表示されます。復元する AMI を選択し、[復元] を選択します。
4. プロンプトが表示されたら、[Recover] (復元) を選択します。

AWS CLI

を使用して、削除された AMI をごみ箱から復元するには AWS CLI

restore-image-from-recycle-bin AWS CLI コマンドを使用します。`--image-id` に復元する AMI の ID を指定します。

```
aws ec2 restore-image-from-recycle-bin --image-id ami_id
```

例えば、次のコマンドでは AMI `ami-01234567890abcdef` をごみ箱から復元します。

```
aws ec2 restore-image-from-recycle-bin --image-id ami-01234567890abcdef
```

コマンドが正常に完了した場合、出力を返しません。

> **Important**
>
> 次のエラーが表示された場合は、 AWS CLI バージョンの更新が必要になる場合があります。詳細については、「コマンドが見つからないエラー」を参照してください。
>
> ```
> aws.exe: error: argument operation: Invalid choice, valid choices are: ...
> ```

Amazon EventBridge を使用してごみ箱をモニタリングする

ごみ箱は、保持ルールに基づいて実行されるアクションのイベントを Amazon EventBridge に送信します。EventBridge を使用することで、これらのイベントへの対応でプログラム的なアクションや通知を呼び出すルールを設定できます。例えば、保持ルールがロック解除され、ロック解除の遅延期間に入ったときにメールに通知を送信する EventBridge ルールを作成できます。詳細については、「イベントに反応する Amazon EventBridge ルールの作成」を参照してください。

EventBridge でのイベントは、JSON オブジェクトとして表されます。イベント固有のフィールドは、JSON オブジェクトの `detail` セクションに表示されます。`event` フィールドにはイベント名が入ります。`result` フィールドには、イベントを開始したアクションの完了時のステータスが入ります。詳細については、「Amazon EventBridge ユーザーガイド」の「Amazon EventBridge のイベントパターン」を参照してください。

Amazon EventBridge の詳細については、「Amazon EventBridge ユーザーガイド」の「What Is Amazon EventBridge?」を参照してください。

イベント

- RuleLocked
- RuleChangeAttempted
- RuleUnlockScheduled
- RuleUnlockingNotice
- RuleUnlocked

RuleLocked

以下は、保持ルールが正常にロックされた場合にごみ箱が生成するイベントの例です。このイベントは、CreateRule リクエストと LockRule リクエストによって生成できます。イベントを生成した API が `api-name` フィールドに表示されます。

```
{
 "version": "0",
 "id": "exampleb-b491-4cf7-a9f1-bf370example",
 "detail-type": "Recycle Bin Rule Locked",
 "source": "aws.rbin",
 "account": "123456789012",
 "time": "2022-08-10T16:37:50Z",
```

```
 "region": "us-west-2",
 "resources": [
 "arn:aws:rbin:us-west-2:123456789012:rule/a12345abcde"
 ],
 "detail":
 {
 "detail-version": " 1.0.0",
 "rule-id": "a12345abcde",
 "rule-description": "locked account level rule",
 "unlock-delay-period": "30 days",
 "api-name": "CreateRule"
 }
 }
```

RuleChangeAttempted

以下は、ロックされたルールを変更または削除しようとして失敗した場合にごみ箱が生成するイベントの例です。このイベントは、DeleteRule リクエストと UpdateRule リクエストによって生成できます。イベントを生成した API が api-name フィールドに表示されます。

```
{
 "version": "0",
 "id": "exampleb-b491-4cf7-a9f1-bf370example",
 "detail-type": "Recycle Bin Rule Change Attempted",
 "source": "aws.rbin",
 "account": "123456789012",
 "time": "2022-08-10T16:37:50Z",
 "region": "us-west-2",
 "resources": [
 "arn:aws:rbin:us-west-2:123456789012:rule/a12345abcde"
 ],
 "detail":
 {
 "detail-version": " 1.0.0",
 "rule-id": "a12345abcde",
 "rule-description": "locked account level rule",
 "unlock-delay-period": "30 days",
 "api-name": "DeleteRule"
 }
 }
```

RuleUnlockScheduled

以下は、保持ルールがロックされロック解除の遅延期間が開始された場合にごみ箱が生成するイベントの例です。

```
{
 "version": "0",
 "id": "exampleb-b491-4cf7-a9f1-bf370example",
 "detail-type": "Recycle Bin Rule Unlock Scheduled",
 "source": "aws.rbin",
 "account": "123456789012",
 "time": "2022-08-10T16:37:50Z",
 "region": "us-west-2",
 "resources": [
 "arn:aws:rbin:us-west-2:123456789012:rule/a12345abcde"
 ],
 "detail":
 {
 "detail-version": " 1.0.0",
 "rule-id": "a12345abcde",
 "rule-description": "locked account level rule",
 "unlock-delay-period": "30 days",
 "scheduled-unlock-time": "2022-09-10T16:37:50Z",
 }
 }
```

RuleUnlockingNotice

以下は、保持ルールがロック解除の遅延期間中に、ロック解除の遅延期間が終了する前日までごみ箱が毎日生成するイベントの例です。

```
{
 "version": "0",
 "id": "exampleb-b491-4cf7-a9f1-bf370example",
 "detail-type": "Recycle Bin Rule Unlocking Notice",
 "source": "aws.rbin",
 "account": "123456789012",
 "time": "2022-08-10T16:37:50Z",
 "region": "us-west-2",
 "resources": [
 "arn:aws:rbin:us-west-2:123456789012:rule/a12345abcde"
 ],
```

```
 "detail":
 {
 "detail-version": " 1.0.0",
 "rule-id": "a12345abcde",
 "rule-description": "locked account level rule",
 "unlock-delay-period": "30 days",
 "scheduled-unlock-time": "2022-09-10T16:37:50Z"
 }
 }
```

RuleUnlocked

以下は、保存ルールのロック解除の遅延期間が終了し、保持ルールを変更または削除できるようになったときにごみ箱が生成するイベントの例です。

```
{
 "version": "0",
 "id": "exampleb-b491-4cf7-a9f1-bf370example",
 "detail-type": "Recycle Bin Rule Unlocked",
 "source": "aws.rbin",
 "account": "123456789012",
 "time": "2022-08-10T16:37:50Z",
 "region": "us-west-2",
 "resources": [
 "arn:aws:rbin:us-west-2:123456789012:rule/a12345abcde"
 ],
 "detail":
 {
 "detail-version": " 1.0.0",
 "rule-id": "a12345abcde",
 "rule-description": "locked account level rule",
 "unlock-delay-period": "30 days",
 "scheduled-unlock-time": "2022-09-10T16:37:50Z"
 }
 }
```

を使用してごみ箱をモニタリングする AWS CloudTrail

ごみ箱サービスは と統合されています AWS CloudTrail。CloudTrail は、ユーザー、ロール、またはサービスによって実行されたアクションを記録する AWS サービスです。CloudTrail は、ごみ箱で実

行されるすべての API コールをイベントとしてキャプチャします。証跡を作成する場合は、Amazon Simple Storage Service (Amazon S3) バケットへの CloudTrail イベントの継続的な配信を有効にすることができます。証跡を設定しない場合でも、CloudTrail コンソールの [Event history] (イベント履歴) で最新の管理イベントを表示できます。CloudTrail で収集された情報を使用して、ごみ箱に対するリクエスト、リクエスト元の IP アドレス、リクエストの実行者、リクエスト日時などの詳細を把握できます。

CloudTrail の詳細については、「AWS CloudTrail ユーザーガイド」を参照してください。

CloudTrail でのごみ箱情報

CloudTrail は、 AWS アカウントの作成時にアカウントで有効になります。ごみ箱でサポートされているイベントアクティビティが発生すると、そのアクティビティはイベント履歴の他の AWS サービスイベントとともに CloudTrail イベントに記録されます。 AWS アカウントで最近のイベントを表示、検索、ダウンロードできます。詳細については、CloudTrail イベント履歴でのイベントの表示を参照してください。

ごみ箱のイベントなど、 AWS アカウントのイベントの継続的な記録については、証跡を作成します。証跡より、CloudTrail はログファイルを S3 バケットに配信できます。デフォルトでは、コンソールで証跡を作成すると、証跡はすべての AWS リージョンに適用されます。証跡は、 AWS パーティション内のすべてのリージョンからのイベントをログに記録し、指定した S3 バケットにログファイルを配信します。さらに、CloudTrail ログで収集されたイベントデータをさらに分析して処理するように他の AWS サービスを設定できます。詳細については、AWS CloudTrail ユーザーガイドの証跡作成の概要を参照してください。

サポートされている API アクション

ごみ箱の場合、CloudTrail を使用して次の API アクションを管理イベントとしてログできます。

- CreateRule
- UpdateRule
- GetRules
- ListRule
- DeleteRule
- TagResource
- UntagResource
- ListTagsForResource
- LockRule

- UnlockRule

管理イベントの記録については、CloudTrail ユーザーガイドの証跡での管理イベントの記録を参照してください。

アイデンティティ情報

各イベントまたはログエントリには、リクエストの生成者に関する情報が含まれます。アイデンティティ情報は、以下を判別するのに役立ちます。

- ルートユーザーまたはユーザー認証情報のどちらを使用してリクエストが送信されたか
- リクエストがロールまたはフェデレーションユーザーのテンポラリなセキュリティ認証情報を使用して行われたかどうか。
- リクエストが別の AWS サービスによって行われたかどうか。

詳細については、 CloudTrail userIdentityElementを参照してください。

ごみ箱ログファイルエントリについて

証跡は、指定した S3 バケットにイベントをログファイルとして配信するように設定できます。CloudTrail ログファイルには、1 つ以上のログエントリがあります。イベントは、任意の出典からの単一のリクエストを表し、リクエストされたアクション、アクションの日時、リクエストパラメータなどに関する情報が含まれます。CloudTrail ログファイルは、パブリック API 呼び出しの順序付けられたスタックトレースではないため、特定の順序では表示されません。

以下に CloudTrail ログエントリの例を示します。

CreateRule


```
{
 "eventVersion": "1.08",
 "userIdentity": {
 "type": "AssumedRole",
 "principalId": "123456789012",
 "arn": "arn:aws:iam::123456789012:root",
 "accountId": "123456789012",
 "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
 "sessionContext": {
   "sessionIssuer": {
     "type": "Role",
```

```
    "principalId": "123456789012",
    "arn": "arn:aws:iam::123456789012:role/Admin",
    "accountId": "123456789012",
    "userName": "Admin"
  },
  "webIdFederationData": {},
  "attributes": {
    "mfaAuthenticated": "false",
    "creationDate": "2021-08-02T21:43:38Z"
  }
}
},
"eventTime": "2021-08-02T21:45:22Z",
"eventSource": "rbin.amazonaws.com",
"eventName": "CreateRule",
"awsRegion": "us-west-2",
"sourceIPAddress": "123.123.123.123",
"userAgent": "aws-cli/1.20.9 Python/3.6.14
Linux/4.9.230-0.1.ac.224.84.332.metal1.x86_64 botocore/1.21.9",
"requestParameters": {
"retentionPeriod": {
  "retentionPeriodValue": 7,
  "retentionPeriodUnit": "DAYS"
},
"description": "Match all snapshots",
"resourceType": "EBS_SNAPSHOT"
},
"responseElements": {
"identifier": "jkrnexample"
},
"requestID": "ex0577a5-amc4-pl4f-ef51-50fdexample",
"eventID": "714fafex-2eam-42pl-913e-926d4example",
"readOnly": false,
"eventType": "AwsApiCall",
"managementEvent": true,
"eventCategory": "Management",
"recipientAccountId": "123456789012",
"tlsDetails": {
"tlsVersion": "TLSv1.2",
"cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
"clientProvidedHostHeader": "rbin.us-west-2.amazonaws.com"
}
}
```

GetRule


```
{
 "eventVersion": "1.08",
 "userIdentity": {
 "type": "AssumedRole",
 "principalId": "123456789012",
 "arn": "arn:aws:iam::123456789012:root",
 "accountId": "123456789012",
 "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
 "sessionContext": {
   "sessionIssuer": {
     "type": "Role",
     "principalId": "123456789012",
     "arn": "arn:aws:iam::123456789012:role/Admin",
     "accountId": "123456789012",
     "userName": "Admin"
   },
   "webIdFederationData": {},
   "attributes": {
     "mfaAuthenticated": "false",
     "creationDate": "2021-08-02T21:43:38Z"
   }
 }
 },
 "eventTime": "2021-08-02T21:45:33Z",
 "eventSource": "rbin.amazonaws.com",
 "eventName": "GetRule",
 "awsRegion": "us-west-2",
 "sourceIPAddress": "123.123.123.123",
 "userAgent": "aws-cli/1.20.9 Python/3.6.14
 Linux/4.9.230-0.1.ac.224.84.332.metal1.x86_64 botocore/1.21.9",
 "requestParameters": {
 "identifier": "jkrnexample"
 },
 "responseElements": null,
 "requestID": "ex0577a5-amc4-pl4f-ef51-50fdexample",
 "eventID": "714fafex-2eam-42pl-913e-926d4example",
 "readOnly": true,
 "eventType": "AwsApiCall",
 "managementEvent": true,
 "eventCategory": "Management",
 "recipientAccountId": "123456789012",
 "tlsDetails": {
```

```
 "tlsVersion": "TLSv1.2",
 "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
 "clientProvidedHostHeader": "rbin.us-west-2.amazonaws.com"
 }
 }
```

ListRules

```
{
 "eventVersion": "1.08",
 "userIdentity": {
 "type": "AssumedRole",
 "principalId": "123456789012",
 "arn": "arn:aws:iam::123456789012:root",
 "accountId": "123456789012",
 "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
 "sessionContext": {
   "sessionIssuer": {
     "type": "Role",
     "principalId": "123456789012",
     "arn": "arn:aws:iam::123456789012:role/Admin",
     "accountId": "123456789012",
     "userName": "Admin"
   },
   "webIdFederationData": {},
   "attributes": {
     "mfaAuthenticated": "false",
     "creationDate": "2021-08-02T21:43:38Z"
   }
 }
 },
 "eventTime": "2021-08-02T21:44:37Z",
 "eventSource": "rbin.amazonaws.com",
 "eventName": "ListRules",
 "awsRegion": "us-west-2",
 "sourceIPAddress": "123.123.123.123",
 "userAgent": "aws-cli/1.20.9 Python/3.6.14
 Linux/4.9.230-0.1.ac.224.84.332.metal1.x86_64 botocore/1.21.9",
 "requestParameters": {
 "resourceTags": [
   {
     "resourceTagKey": "test",
     "resourceTagValue": "test"
```

```
    }
  ]
  },
  "responseElements": null,
  "requestID": "ex0577a5-amc4-pl4f-ef51-50fdexample",
  "eventID": "714fafex-2eam-42pl-913e-926d4example",
  "readOnly": true,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "eventCategory": "Management",
  "recipientAccountId": "123456789012",
  "tlsDetails": {
  "tlsVersion": "TLSv1.2",
  "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
  "clientProvidedHostHeader": "rbin.us-west-2.amazonaws.com"
  }
  }
```

UpdateRule

```
{
  "eventVersion": "1.08",
  "userIdentity": {
  "type": "AssumedRole",
  "principalId": "123456789012",
  "arn": "arn:aws:iam::123456789012:root",
  "accountId": "123456789012",
  "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
  "sessionContext": {
    "sessionIssuer": {
      "type": "Role",
      "principalId": "123456789012",
      "arn": "arn:aws:iam::123456789012:role/Admin",
      "accountId": "123456789012",
      "userName": "Admin"
    },
    "webIdFederationData": {},
    "attributes": {
      "mfaAuthenticated": "false",
      "creationDate": "2021-08-02T21:43:38Z"
    }
  }
  },
```

```
 "eventTime": "2021-08-02T21:46:03Z",
 "eventSource": "rbin.amazonaws.com",
 "eventName": "UpdateRule",
 "awsRegion": "us-west-2",
 "sourceIPAddress": "123.123.123.123",
 "userAgent": "aws-cli/1.20.9 Python/3.6.14
 Linux/4.9.230-0.1.ac.224.84.332.metal1.x86_64 botocore/1.21.9",
 "requestParameters": {
 "identifier": "jkrnexample",
 "retentionPeriod": {
   "retentionPeriodValue": 365,
   "retentionPeriodUnit": "DAYS"
 },
 "description": "Match all snapshots",
 "resourceType": "EBS_SNAPSHOT"
 },
 "responseElements": null,
 "requestID": "ex0577a5-amc4-pl4f-ef51-50fdexample",
 "eventID": "714fafex-2eam-42pl-913e-926d4example",
 "readOnly": false,
 "eventType": "AwsApiCall",
 "managementEvent": true,
 "eventCategory": "Management",
 "recipientAccountId": "123456789012",
 "tlsDetails": {
 "tlsVersion": "TLSv1.2",
 "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
 "clientProvidedHostHeader": "rbin.us-west-2.amazonaws.com"
 }
 }
```


DeleteRule


```
{
 "eventVersion": "1.08",
 "userIdentity": {
 "type": "AssumedRole",
 "principalId": "123456789012",
 "arn": "arn:aws:iam::123456789012:root",
 "accountId": "123456789012",
 "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
 "sessionContext": {
   "sessionIssuer": {
```

```
    "type": "Role",
    "principalId": "123456789012",
    "arn": "arn:aws:iam::123456789012:role/Admin",
    "accountId": "123456789012",
    "userName": "Admin"
  },
  "webIdFederationData": {},
  "attributes": {
    "mfaAuthenticated": "false",
    "creationDate": "2021-08-02T21:43:38Z"
  }
}
},
"eventTime": "2021-08-02T21:46:25Z",
"eventSource": "rbin.amazonaws.com",
"eventName": "DeleteRule",
"awsRegion": "us-west-2",
"sourceIPAddress": "123.123.123.123",
"userAgent": "aws-cli/1.20.9 Python/3.6.14
Linux/4.9.230-0.1.ac.224.84.332.metal1.x86_64 botocore/1.21.9",
"requestParameters": {
"identifier": "jkrnexample"
},
"responseElements": null,
"requestID": "ex0577a5-amc4-pl4f-ef51-50fdexample",
"eventID": "714fafex-2eam-42pl-913e-926d4example",
"readOnly": false,
"eventType": "AwsApiCall",
"managementEvent": true,
"eventCategory": "Management",
"recipientAccountId": "123456789012",
"tlsDetails": {
"tlsVersion": "TLSv1.2",
"cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
"clientProvidedHostHeader": "rbin.us-west-2.amazonaws.com"
}
}
```


TagResource


```
{
 "eventVersion": "1.08",
 "userIdentity": {
```

```
"type": "AssumedRole",
"principalId": "123456789012",
"arn": "arn:aws:iam::123456789012:root",
"accountId": "123456789012",
"accessKeyId": "AKIAIOSFODNN7EXAMPLE",
"sessionContext": {
  "sessionIssuer": {
    "type": "Role",
    "principalId": "123456789012",
    "arn": "arn:aws:iam::123456789012:role/Admin",
    "accountId": "123456789012",
    "userName": "Admin"
  },
  "webIdFederationData": {},
  "attributes": {
    "mfaAuthenticated": "false",
    "creationDate": "2021-10-22T21:38:34Z"
  }
}
},
"eventTime": "2021-10-22T21:43:15Z",
"eventSource": "rbin.amazonaws.com",
"eventName": "TagResource",
"awsRegion": "us-west-2",
"sourceIPAddress": "123.123.123.123",
"userAgent": "aws-cli/1.20.26 Python/3.6.14
Linux/4.9.273-0.1.ac.226.84.332.metal1.x86_64 botocore/1.21.26",
"requestParameters": {
"resourceArn": "arn:aws:rbin:us-west-2:123456789012:rule/ABCDEF01234",
"tags": [
  {
    "key": "purpose",
    "value": "production"
  }
]
},
"responseElements": null,
"requestID": "examplee-7962-49ec-8633-795efexample",
"eventID": "example4-6826-4c0a-bdec-0bab1example",
"readOnly": false,
"eventType": "AwsApiCall",
"managementEvent": true,
"eventCategory": "Management",
"recipientAccountId": "123456789012",
```

```
 "tlsDetails": {
 "tlsVersion": "TLSv1.2",
 "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
 "clientProvidedHostHeader": "rbin.us-west-2.amazonaws.com"
 }
 }
```

UntagResource

```
{
 "eventVersion": "1.08",
 "userIdentity": {
 "type": "AssumedRole",
 "principalId": "123456789012",
 "arn": "arn:aws:iam::123456789012:root",
 "accountId": "123456789012",
 "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
 "sessionContext": {
   "sessionIssuer": {
     "type": "Role",
     "principalId": "123456789012",
     "arn": "arn:aws:iam::123456789012:role/Admin",
     "accountId": "123456789012",
     "userName": "Admin"
   },
   "webIdFederationData": {},
   "attributes": {
     "mfaAuthenticated": "false",
     "creationDate": "2021-10-22T21:38:34Z"
   }
 }
 },
 "eventTime": "2021-10-22T21:44:16Z",
 "eventSource": "rbin.amazonaws.com",
 "eventName": "UntagResource",
 "awsRegion": "us-west-2",
 "sourceIPAddress": "123.123.123.123",
 "userAgent": "aws-cli/1.20.26 Python/3.6.14
 Linux/4.9.273-0.1.ac.226.84.332.metal1.x86_64 botocore/1.21.26",
 "requestParameters": {
 "resourceArn": "arn:aws:rbin:us-west-2:123456789012:rule/ABCDEF01234",
 "tagKeys": [
   "purpose"
```

```
]
},
"responseElements": null,
"requestID": "example7-6c1e-4f09-9e46-bb957example",
"eventID": "example6-75ff-4c94-a1cd-4d5f5example",
"readOnly": false,
"eventType": "AwsApiCall",
"managementEvent": true,
"eventCategory": "Management",
"recipientAccountId": "123456789012",
"tlsDetails": {
"tlsVersion": "TLSv1.2",
"cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
"clientProvidedHostHeader": "rbin.us-west-2.amazonaws.com"
}
}
```

ListTagsForResource

```
{
"eventVersion": "1.08",
"userIdentity": {
"type": "AssumedRole",
"principalId": "123456789012",
"arn": "arn:aws:iam::123456789012:root",
"accountId": "123456789012",
"accessKeyId": "AKIAIOSFODNN7EXAMPLE",
"sessionContext": {
  "sessionIssuer": {
    "type": "Role",
    "principalId": "123456789012",
    "arn": "arn:aws:iam::123456789012:role/Admin",
    "accountId": "123456789012",
    "userName": "Admin"
  },
  "webIdFederationData": {},
  "attributes": {
    "mfaAuthenticated": "false",
    "creationDate": "2021-10-22T21:38:34Z"
  }
}
},
"eventTime": "2021-10-22T21:42:31Z",
```

```
"eventSource": "rbin.amazonaws.com",
"eventName": "ListTagsForResource",
"awsRegion": "us-west-2",
"sourceIPAddress": "123.123.123.123",
"userAgent": "aws-cli/1.20.26 Python/3.6.14
Linux/4.9.273-0.1.ac.226.84.332.metal1.x86_64 botocore/1.21.26",
"requestParameters": {
"resourceArn": "arn:aws:rbin:us-west-2:123456789012:rule/ABCDEF01234"
},
"responseElements": null,
"requestID": "example8-10c7-43d4-b147-3d9d9example",
"eventID": "example2-24fc-4da7-a479-c9748example",
"readOnly": true,
"eventType": "AwsApiCall",
"managementEvent": true,
"eventCategory": "Management",
"recipientAccountId": "123456789012",
"tlsDetails": {
"tlsVersion": "TLSv1.2",
"cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
"clientProvidedHostHeader": "rbin.us-west-2.amazonaws.com"
}
}
```


LockRule


```
{
"eventVersion": "1.08",
"userIdentity": {
"type": "AssumedRole",
"principalId": "123456789012",
"arn": "arn:aws:iam::123456789012:root",
"accountId": "123456789012",
"accessKeyId": "AKIAIOSFODNN7EXAMPLE",
"sessionContext": {
  "sessionIssuer": {
    "type": "Role",
    "principalId": "123456789012",
    "arn": "arn:aws:iam::123456789012:role/Admin",
    "accountId": "123456789012",
    "userName": "Admin"
  },
  "webIdFederationData": {},
```

```
  "attributes": {
    "creationDate": "2022-10-25T00:45:11Z",
    "mfaAuthenticated": "false"
  }
}
},
"eventTime": "2022-10-25T00:45:19Z",
"eventSource": "rbin.amazonaws.com",
"eventName": "LockRule",
"awsRegion": "us-west-2",
"sourceIPAddress": "123.123.123.123",
"userAgent": "python-requests/2.25.1",
"requestParameters": {
"identifier": "jkrnexample",
"lockConfiguration": {
  "unlockDelay": {
    "unlockDelayValue": 7,
    "unlockDelayUnit": "DAYS"
  }
}
},
"responseElements": {
"identifier": "jkrnexample",
"description": "",
"resourceType": "EBS_SNAPSHOT",
"retentionPeriod": {
  "retentionPeriodValue": 7,
  "retentionPeriodUnit": "DAYS"
},
"resourceTags": [],
"status": "available",
"lockConfiguration": {
  "unlockDelay": {
    "unlockDelayValue": 7,
    "unlockDelayUnit": "DAYS"
  }
},
"lockState": "locked"
},
"requestID": "ex0577a5-amc4-pl4f-ef51-50fdexample",
"eventID": "714fafex-2eam-42pl-913e-926d4example",
"readOnly": false,
"eventType": "AwsApiCall",
"managementEvent": true,
```

```
 "recipientAccountId": "123456789012",
 "eventCategory": "Management",
 "tlsDetails": {
 "tlsVersion": "TLSv1.2",
 "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
 "clientProvidedHostHeader": "rbin.us-west-2.amazonaws.com"
 }
 }
```

UnlockRule

```
{
 "eventVersion": "1.08",
 "userIdentity": {
 "type": "AssumedRole",
 "principalId": "123456789012",
 "arn": "arn:aws:iam::123456789012:root",
 "accountId": "123456789012",
 "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
 "sessionContext": {
   "sessionIssuer": {
     "type": "Role",
     "principalId": "123456789012",
     "arn": "arn:aws:iam::123456789012:role/Admin",
     "accountId": "123456789012",
     "userName": "Admin"
   },
   "webIdFederationData": {},
   "attributes": {
     "creationDate": "2022-10-25T00:45:11Z",
     "mfaAuthenticated": "false"
   }
 }
 },
 "eventTime": "2022-10-25T00:46:17Z",
 "eventSource": "rbin.amazonaws.com",
 "eventName": "UnlockRule",
 "awsRegion": "us-west-2",
 "sourceIPAddress": "123.123.123.123",
 "userAgent": "python-requests/2.25.1",
 "requestParameters": {
 "identifier": "jkrnexample"
 },
```

```
"responseElements": {
"identifier": "jkrnexample",
"description": "",
"resourceType": "EC2_IMAGE",
"retentionPeriod": {
  "retentionPeriodValue": 7,
  "retentionPeriodUnit": "DAYS"
},
"resourceTags": [],
"status": "available",
"lockConfiguration": {
  "unlockDelay": {
    "unlockDelayValue": 7,
    "unlockDelayUnit": "DAYS"
  }
},
"lockState": "pending_unlock",
"lockEndTime": "Nov 1, 2022, 12:46:17 AM"
},
"requestID": "ex0577a5-amc4-pl4f-ef51-50fdexample",
"eventID": "714fafex-2eam-42pl-913e-926d4example",
"readOnly": false,
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "123456789012",
"eventCategory": "Management",
"tlsDetails": {
"tlsVersion": "TLSv1.2",
"cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
"clientProvidedHostHeader": "rbin.us-west-2.amazonaws.com"
}
}
```


ごみ箱のサービスエンドポイント

エンドポイントは、 AWS ウェブサービスのエントリポイントとして機能する URL です。ごみ箱は、次のエンドポイントタイプをサポートしています。

- IPv4 エンドポイント
- IPv4 と IPv6 の両方をサポートするデュアルスタックのエンドポイント
- FIPS エンドポイント

リクエストを行うと、使用するエンドポイントとリージョンを指定できます。エンドポイントを指定しない場合、デフォルトで IPv4 エンドポイントが使用されます。別のエンドポイントタイプを使用するには、リクエストで指定する必要があります。これを行う方法の例については、「エンドポイントの指定」を参照してください。

ごみ箱については、「Amazon Web Services 全般のリファレンス」の「Recycle Bin endpoints and quotas」を参照してください。

トピック

- IPv4 エンドポイント
- デュアルスタック (IPv4 および IPv6) エンドポイント
- FIPS エンドポイント
- エンドポイントの指定

IPv4 エンドポイント

IPv4 エンドポイントは IPv4 トラフィックのみをサポートします。IPv4 エンドポイントは、全リージョンで利用できます。

リージョンをエンドポイント名の一部として指定する必要があります。エンドポイント名には、次の命名規則が使用されます。

- rbin.*region*.amazonaws.com

例えば、米国東部 (バージニア北部) リージョンの IPv4 エンドポイントは `rbin.us-east-1.amazonaws.com` です。

デュアルスタック (IPv4 および IPv6) エンドポイント

デュアルスタックエンドポイントは、IPv4 と IPv6 トラフィックの両方をサポートします。デュアルスタックエンドポイントは、すべてのリージョンで利用できます。

IPv6 を使用するには、デュアルスタックエンドポイントを使用する必要があります。デュアルスタックエンドポイントにリクエストを行うと、エンドポイント URL は、ネットワークとクライアントが使用するプロトコルに応じて IPv6 または IPv4 アドレスに解決されます。

リージョンをエンドポイント名の一部として指定する必要があります。デュアルスタックエンドポイント名には、次の命名規則が使用されます。

- rbin.*region*.api.aws

例えば、米国東部 (バージニア北部) リージョンのデュアルスタックエンドポイントは rbin.us-east-1.api.aws です。

FIPS エンドポイント

ごみ箱は、次のリージョンの FIPS 検証済み IPv4 およびデュアルスタック (IPv4 および IPv6) エンドポイントを提供します。

- us-east-1 — 米国東部 (バージニア北部)
- us-east-2 — 米国東部 (オハイオ)
- us-west-1 — 米国西部 (北カリフォルニア)
- us-west-2 — 米国西部 (オレゴン)
- ca-central-1 — カナダ (中部)
- ca-west-1 — カナダ西部 (カルガリー)
- us-gov-east-1 — AWS GovCloud (US-East)
- us-gov-west-1 — AWS GovCloud (US-West)

FIPS IPv4 エンドポイントに使用される命名規則は rbin-fips.*region*.amazonaws.com です。例えば、米国東部 (バージニア北部) リージョンの FIPS IPv4 エンドポイントは rbin-fips.us-east-1.amazonaws.com です。

FIPS デュアルスタックエンドポイントの命名規則は rbin-fips.*region*.api.aws です。例えば、米国東部 (バージニア北部) リージョンの FIPS デュアルスタックのエンドポイントは rbin-fips.us-east-1.api.aws です。

エンドポイントの指定

次の例は、 AWS CLIを使用して us-east-2 リージョンのエンドポイントを指定する方法を示しています。

- デュアルスタック

```
aws rbin get-rule \
--identifier rule_id \
```

```
--endpoint-url https://rbin.us-east-2.api.aws
```

- IPv4

```
aws rbin get-rule \
--identifier rule_id \
--endpoint-url https://rbin.us-east-2.amazonaws.com
```

VPC とごみ箱の間にプライベート接続を作成する

AWS PrivateLink を利用したインターフェイス VPC エンドポイントを作成することで、VPC とごみ箱間にプライベート接続を設定できます。インターネットゲートウェイ、NAT デバイス、VPN 接続、または Direct Connect 接続を使用せずに、VPC 内にあるかのようにごみ箱にアクセスできます。VPC のインスタンスは、パブリック IP アドレスがなくてもごみ箱と通信できます。

インターフェイスエンドポイントに対して有効にする各サブネットにエンドポイントネットワークインターフェイスを作成します。

詳細については、「 AWS PrivateLink ガイド」の「 を通じて AWS サービスにアクセスする AWS PrivateLink」を参照してください。

ごみ箱のインターフェイス VPC エンドポイントを作成

Amazon VPC コンソールまたは AWS CLIを使用してごみ箱用の VPC エンドポイントを作成できます。詳細については、「AWS PrivateLink ガイド」の「Create a VPC endpoint」を参照してください。

以下のサービス名 `com.amazonaws.region.rbin` を使用して VPC エンドポイントを作成します。

エンドポイントに対してプライベート DNS を有効にすると、リージョンのデフォルト DNS 名 (`rbin.us-east-1.amazonaws.com` など) を使用して、ごみ箱へ API リクエストを実行できます。

ごみ箱用の VPC エンドポイントポリシーを作成

デフォルトで、エンドポイント経由でのごみ箱への完全なアクセスが許可されます。VPC エンドポイントポリシーを使用してインターフェイスエンドポイントへのアクセスを制御できます。VPC エ

ンドポイントには、ごみ箱へのアクセスを制御するエンドポイントポリシーをアタッチできます。このポリシーでは、以下の情報を指定します。

- アクションを実行できるプリンシパル。
- 実行可能なアクション。
- アクションを実行できるリソース。

詳細については、Amazon VPC ユーザーガイドのVPC エンドポイントによるサービスのアクセスコントロールを参照してください。

```
{
    "Statement": [
    {
        "Effect": "Allow",
        "Action": "rbin:*",
        "Resource": "*",
        "Principal": "*"
    },
    {
        "Effect": "Deny",
        "Action": "rbin:DeleteRule",
        "Resource": "*",
        "Principal": "*",
        "Condition": {
            "StringEquals" : {
                "rbin:Attribute/ResourceType": "EBS_SNAPSHOT"
            }
        }
    }]
}
```

Amazon EBS のセキュリティ

のクラウドセキュリティが最優先事項 AWS です。お客様は AWS 、セキュリティを最も重視する組織の要件を満たすように構築されたデータセンターとネットワークアーキテクチャを活用できます。

セキュリティは、 AWS とお客様の間の責任共有です。責任共有モデルではこれをクラウドのセキュリティおよびクラウド内のセキュリティと説明しています。

- クラウドのセキュリティ – AWS は、 で AWS サービスを実行するインフラストラクチャを保護する責任があります AWS クラウド。 AWS また、 では、安全に使用できるサービスも提供しています。サードパーティーの監査者は、AWS コンプライアンスプログラムコンプライアンスプログラムの一環として、当社のセキュリティの有効性を定期的にテストおよび検証。Amazon Elastic Block Store に適用されるコンプライアンスプログラムの詳細については、「コンプライアンスプログラムAWS による対象範囲内のサービスコンプライアンスプログラム」を参照してください。
- クラウド内のセキュリティ – お客様の責任は、使用する AWS サービスによって決まります。また、ユーザーは、データの機密性、会社の要件、適用される法律や規制など、その他の要因についても責任を負います。

このドキュメントは、Amazon EBS 使用時の責任共有モデルの適用方法を理解するのうえで役立ちます。次のトピックでは、セキュリティおよびコンプライアンスの目標を達成するため、Amazon EBS を構成する方法について説明します。また、Amazon EBS リソースのモニタリングや保護に役立つ他の AWS サービスの使用方法についても説明します。

トピック

- Amazon EBS でのデータ保護
- Amazon EBS 用の Identity and Access Management
- Amazon EBS のコンプライアンス検証
- Amazon EBS のデータ回復力

Amazon EBS でのデータ保護

責任共有モデル、Amazon AWS Elastic Block Store でのデータ保護に適用されます。このモデルで説明されているように、 AWS はすべての を実行するグローバルインフラストラクチャを保護する責任があります AWS クラウド。ユーザーは、このインフラストラクチャでホストされるコンテンツに対する管理を維持する責任があります。また、使用する「 AWS のサービス 」のセ

キュリティ設定と管理タスクもユーザーの責任となります。データプライバシーの詳細については、「Data Privacy FAQChina」を参照してください。 欧州におけるデータ保護に関する情報については、General Data Protection Regulation (GDPR) Center を参照してください。

データ保護の目的で、認証情報を保護し AWS アカウント 、 AWS IAM アイデンティティセンターまたは AWS Identity and Access Management (IAM) を使用して個々のユーザーを設定することをお勧めします。この方法により、それぞれのジョブを遂行するために必要な権限のみが各ユーザーに付与されます。また、次の方法でデータを保護することもお勧めします:

- 各アカウントで多要素認証 (MFA) を使用します。
- SSL/TLS を使用して AWS リソースと通信します。TLS 1.2 は必須ですが、TLS 1.3 を推奨します。
- で API とユーザーアクティビティのログ記録を設定します AWS CloudTrail。CloudTrail 証跡を使用して AWS アクティビティをキャプチャする方法については、「 AWS CloudTrail ユーザーガイド」のCloudTrail 証跡の使用」を参照してください。
- AWS 暗号化ソリューションと、 内のすべてのデフォルトのセキュリティコントロールを使用します AWS のサービス。
- Amazon Macie などの高度な管理されたセキュリティサービスを使用します。これらは、Amazon S3 に保存されている機密データの検出と保護を支援します。
- コマンドラインインターフェイスまたは API AWS を介して にアクセスするときに FIPS 140-3 検証済み暗号化モジュールが必要な場合は、FIPS エンドポイントを使用します。利用可能な FIPS エンドポイントの詳細については、「連邦情報処理規格 (FIPS) 140-3」を参照してください。

お客様の E メールアドレスなどの極秘または機密情報を、タグ、または [名前] フィールドなどの自由形式のテキストフィールドに含めないことを強くお勧めします。これは、コンソール、API、または SDK を使用して Amazon EBS AWS CLIまたは他の AWS のサービス を使用する場合も同様です。 AWS SDKs タグ、または名前に使用される自由記述のテキストフィールドに入力したデータは、請求または診断ログに使用される場合があります。外部サーバーに URL を提供する場合、そのサーバーへのリクエストを検証できるように、認証情報を URL に含めないことを強くお勧めします。

トピック

- Amazon EBS のデータセキュリティ
- 保管中と転送中の暗号化
- KMS キー管理

Amazon EBS のデータセキュリティ

Amazon EBS ボリュームは、初期化されていない raw ブロックデバイスとして表示されます。これらのデバイスは、EBS インフラストラクチャで作成された論理デバイスです。Amazon EBS サービスは、顧客が使用または再利用する前に、デバイスが論理的に空である (つまり、raw ブロックがゼロになっているか、暗号的に擬似ランダムデータが含まれている) ことを確認します。

DoD 5220.22-M (National Industrial Security Program Operating Manual) や NIST 800-88 (Guidelines for Media Sanitization) など、特定の方法を使用してすべてのデータを消去する必要がある場合は、Amazon EBS で行うことができます。使用前または使用後にデータを消去するか、手順の要件に応じて両方を消去できます。ブロックレベルのアクティビティは、Amazon EBS サービス内の基盤となるストレージメディアに反映されます。

保管中と転送中の暗号化

Amazon EBS 暗号化は、暗号化キーを使用して Amazon EBS ボリュームと Amazon EBS スナップショットを暗号化できるようにする AWS Key Management Service 暗号化ソリューションです。EBS 暗号化オペレーションは、Amazon EC2 インスタンスをホストするサーバーで行われます。これにより、インスタンスとそのアタッチされたボリュームとそれ以降のスナップショットとの間でdata-at-restdata-in-transitの両方のセキュリティが確保されます。詳細については、「Amazon EBS 暗号化」を参照してください。

KMS キー管理

暗号化された Amazon EBS ボリュームまたはスナップショットを作成するときは、 AWS Key Management Service キーを指定します。デフォルトでは、Amazon EBS はアカウントとリージョン() で Amazon EBS の AWS マネージド KMS キーを使用しますaws/ebs。ただし、ユーザーが作成および管理するカスタマーマネージド KMS キーを指定することができます。カスタマーマネージド KMS キー を使用すると、KMS キー を作成、更新、無効化する機能を含め、より柔軟性が得られます。

カスタマーマネージド KMS キーを使用するには、ユーザーに KMS キーを使用する許可を付与する必要があります。詳細については、「ユーザーのアクセス許可」を参照してください。

> **Important**
>
> Amazon EBS は、「対称 KMS キー」のみをサポートします。「非対称 KMS キー」を使用して Amazon EBS ボリュームおよびスナップショットを暗号化することはできませ

ん。KMS キーが対称か非対称かを判別する方法については、「Identify asymmetric KMS keys」を参照してください。

ボリュームごとに、Amazon EBS は指定した KMS キーで暗号化された一意のデータキーを生成する AWS KMS ように に求めます。Amazon EBS は、暗号化されたデータキーをボリュームとともに保存します。次に、ボリュームを Amazon EC2 インスタンスにアタッチすると、Amazon EBS はを呼び出し AWS KMS てデータキーを復号します。Amazon EBS は ハイパーバイザーメモリでプレーンテキストデータキーを使用し、すべての I/O をボリュームに暗号化します。詳細については、「Amazon EBS 暗号化の仕組み」を参照してください。

Amazon EBS 用の Identity and Access Management

AWS Identity and Access Management (IAM) は、管理者が AWS リソースへのアクセスを安全に制御 AWS のサービス するのに役立つ です。IAM 管理者は、誰を認証 (サインインを許可) し、誰に Amazon EBS リソースの使用を承認する (アクセス許可を付与する) かを制御します。IAM は、追加料金なしで使用できる AWS のサービス です。

トピック

- オーディエンス
- アイデンティティを使用した認証
- ポリシーを使用したアクセスの管理
- Amazon EBS と IAM が連携する仕組み
- Amazon EBS の IAM ポリシーの例
- Amazon EBS の認証問題のトラブルシューティング

オーディエンス

AWS Identity and Access Management (IAM) の使用方法は、ロールによって異なります。

- サービスユーザー - 機能にアクセスできない場合は、管理者にアクセス許可をリクエストします (「Amazon EBS の認証問題のトラブルシューティング」を参照)。
- サービス管理者 - ユーザーアクセスを決定し、アクセス許可リクエストを送信します (「Amazon EBS と IAM が連携する仕組み」を参照)
- IAM 管理者 - アクセスを管理するためのポリシーを作成します (「Amazon EBS の IAM ポリシーの例」を参照)

アイデンティティを使用した認証

認証とは、ID 認証情報 AWS を使用して にサインインする方法です。、IAM ユーザー AWS アカウントのルートユーザー、または IAM ロールを引き受けることで認証される必要があります。

(AWS IAM アイデンティティセンター IAM Identity Center)、シングルサインオン認証、Google/Facebook 認証情報などの ID ソースからの認証情報を使用して、フェデレーティッド ID としてサインインできます。サインインの詳細については、「AWS サインイン ユーザーガイド」の「AWS アカウントにサインインする方法」を参照してください。

プログラムによるアクセスの場合、 は SDK と CLI AWS を提供してリクエストを暗号化して署名します。詳細については、「IAM ユーザーガイド」の「API リクエストに対するAWS 署名バージョン 4」を参照してください。

AWS アカウント ルートユーザー

を作成するときは AWS アカウント、まず、すべての AWS のサービス および リソースへの完全なアクセス権を持つ AWS アカウント root ユーザーと呼ばれる 1 つのサインインアイデンティティから始めます。日常的なタスクには、ルートユーザーを使用しないことを強くお勧めします。ルートユーザー認証情報を必要とするタスクについては、「IAM ユーザーガイド」の「ルートユーザー認証情報が必要なタスク」を参照してください。

フェデレーテッドアイデンティティ

ベストプラクティスとして、人間のユーザーが一時的な認証情報 AWS のサービス を使用して にアクセスするには、ID プロバイダーとのフェデレーションを使用する必要があります。

フェデレーティッド ID は、エンタープライズディレクトリ、ウェブ ID プロバイダー、または ID Directory Service ソースの認証情報 AWS のサービス を使用して にアクセスするユーザーです。フェデレーテッドアイデンティティは、一時的な認証情報を提供するロールを引き受けます。

アクセスを一元管理する場合は、 AWS IAM アイデンティティセンターをお勧めします。詳細については、「AWS IAM アイデンティティセンター ユーザーガイド」の「IAM アイデンティティセンターとは」を参照してください。

IAM ユーザーとグループ

IAM ユーザーは、特定の個人やアプリケーションに対する特定のアクセス許可を持つアイデンティティです。長期認証情報を持つ IAM ユーザーの代わりに一時的な認証情報を使用することをお勧め

します。詳細については、IAM ユーザーガイドの「ID プロバイダーとのフェデレーションを使用して にアクセスする必要がある AWS」を参照してください。

IAM グループは、IAM ユーザーの集合を指定し、大量のユーザーに対するアクセス許可の管理を容易にします。詳細については、「IAM ユーザーガイド」の「IAM ユーザーに関するユースケース」を参照してください。

IAM ロール

IAM ロールは、特定のアクセス許可を持つアイデンティであり、一時的な認証情報を提供します。ユーザーから IAM ロール (コンソール) に切り替えるか、 または API オペレーションを呼び出すことで、ロールを引き受けることができます。 AWS CLI AWS 詳細については、「IAM ユーザーガイド」の「ロールを引き受けるための各種方法」を参照してください。

IAM ロールは、フェデレーションユーザーアクセス、一時的な IAM ユーザーのアクセス許可、クロスアカウントアクセス、クロスサービスアクセス、および Amazon EC2 で実行するアプリケーションに役立ちます。詳細については、IAM ユーザーガイド の IAM でのクロスアカウントリソースアクセス を参照してください。

ポリシーを使用したアクセスの管理

でアクセスを制御する AWS には、ポリシーを作成し、ID AWS またはリソースにアタッチします。ポリシーは、アイデンティティまたはリソースに関連付けられたときにアクセス許可を定義します。は、プリンシパルがリクエストを行うときにこれらのポリシー AWS を評価します。ほとんどのポリシーは JSON ドキュメント AWS として に保存されます。JSON ポリシードキュメントの詳細については、「IAM ユーザーガイド」の「JSON ポリシー概要」を参照してください。

管理者は、ポリシーを使用して、どのプリンシパルがどのリソースに対して、どのような条件でアクションを実行できるかを定義することで、誰が何にアクセスできるかを指定します。

デフォルトでは、ユーザーやロールにアクセス許可はありません。IAM 管理者は IAM ポリシーを作成してロールに追加し、このロールをユーザーが引き受けられるようにします。IAM ポリシーは、オペレーションの実行方法を問わず、アクセス許可を定義します。

アイデンティティベースのポリシー

アイデンティティベースのポリシーは、アイデンティティ (ユーザー、グループ、またはロール) にアタッチできる JSON アクセス許可ポリシードキュメントです。これらのポリシーは、アイデンティティがどのリソースに対してどのような条件下でどのようなアクションを実行できるかを制御し

ます。アイデンティティベースポリシーの作成方法については、IAM ユーザーガイド の カスタマー管理ポリシーでカスタム IAM アクセス許可を定義する を参照してください。

アイデンティティベースのポリシーは、インラインポリシー (単一の ID に直接埋め込む) または管理ポリシー (複数の ID にアタッチされたスタンドアロンポリシー) にすることができます。管理ポリシーとインラインポリシーのいずれかを選択する方法については、「IAM ユーザーガイド」の「管理ポリシーとインラインポリシーのいずれかを選択する」を参照してください。

リソースベースのポリシー

リソースベースのポリシーは、リソースに添付する JSON ポリシードキュメントです。例としては、IAM ロール信頼ポリシーや Amazon S3 バケットポリシーなどがあります。リソースベースのポリシーをサポートするサービスでは、サービス管理者はポリシーを使用して特定のリソースへのアクセスを制御できます。リソースベースのポリシーでは、プリンシパルを指定する必要があります。

リソースベースのポリシーは、そのサービス内にあるインラインポリシーです。リソースベースのポリシーでは、IAM の AWS マネージドポリシーを使用できません。

その他のポリシータイプ

AWS は、より一般的なポリシータイプによって付与されるアクセス許可の最大数を設定できる追加のポリシータイプをサポートしています。

- アクセス許可の境界 – アイデンティティベースのポリシーで IAM エンティティに付与することのできるアクセス許可の数の上限を設定します。詳細については、「IAM ユーザーガイド」の「IAM エンティティのアクセス許可境界」を参照してください。
- サービスコントロールポリシー (SCP) - AWS Organizations内の組織または組織単位の最大のアクセス許可を指定します。詳細については、「AWS Organizations ユーザーガイド」の「サービスコントロールポリシー」を参照してください。
- リソースコントロールポリシー (RCP) – は、アカウント内のリソースで利用できる最大数のアクセス許可を定義します。詳細については、「AWS Organizations ユーザーガイド」の「リソースコントロールポリシー (RCP)」を参照してください。
- セッションポリシー – ロールまたはフェデレーションユーザーの一時セッションを作成する際にパラメータとして渡される高度なポリシーです。詳細については、「IAM ユーザーガイド」の「セッションポリシー」を参照してください。

複数のポリシータイプ

1 つのリクエストに複数のタイプのポリシーが適用されると、結果として作成されるアクセス許可を理解するのがさらに難しくなります。が複数のポリシータイプが関与する場合にリクエストを許可するかどうか AWS を決定する方法については、「IAM ユーザーガイド」の「ポリシー評価ロジック」を参照してください。

Amazon EBS と IAM が連携する仕組み

IAM を使用して Amazon EBS へのアクセスを管理する前に、Amazon EBS で利用できる IAM 機能について説明します。

Amazon Elastic Block Store で使用できる IAM 機能

IAM 機能	Amazon EBS サポート
アイデンティティベースのポリシー	あり
リソースベースのポリシー	なし
ポリシーアクション	あり
ポリシーリソース	あり
ポリシー条件キー	あり
ACL	なし
ABAC (ポリシー内のタグ)	部分的
一時認証情報	あり
プリンシパルアクセス権限	あり
サービスロール	あり
サービスリンクロール	いいえ

Amazon EBS およびその他の AWS のサービスがほとんどの IAM 機能と連携する方法の概要については、「IAM ユーザーガイド」のAWS 「IAM と連携する のサービス」を参照してください。

Amazon EBS のアイデンティティベースのポリシー

アイデンティティベースのポリシーのサポート: あり

アイデンティティベースポリシーは、IAM ユーザー、ユーザーグループ、ロールなど、アイデンティティにアタッチできる JSON 許可ポリシードキュメントです。これらのポリシーは、ユーザーとロールが実行できるアクション、リソース、および条件をコントロールします。アイデンティティベースポリシーの作成方法については、「IAM ユーザーガイド」の「カスタマー管理ポリシーでカスタム IAM アクセス許可を定義する」を参照してください。

IAM アイデンティティベースのポリシーでは、許可または拒否するアクションとリソース、およびアクションを許可または拒否する条件を指定できます。JSON ポリシーで使用できるすべての要素について学ぶには、「IAM ユーザーガイド」の「IAM JSON ポリシーの要素のリファレンス」を参照してください。

Amazon EBS のアイデンティティベースのポリシー例

Amazon EBS のアイデンティティベースのポリシー例を確認するには、「Amazon EBS の IAM ポリシーの例」を参照してください。

Amazon EBS 内のリソースベースのポリシー

リソースベースのポリシーのサポート: なし

リソースベースのポリシーは、リソースに添付する JSON ポリシードキュメントです。リソースベースのポリシーには例として、IAM ロールの信頼ポリシー や Amazon S3 バケットポリシー があげられます。リソースベースのポリシーをサポートするサービスでは、サービス管理者はポリシーを使用して特定のリソースへのアクセスをコントロールできます。ポリシーがアタッチされているリソースの場合、指定されたプリンシパルがそのリソースに対して実行できるアクションと条件は、ポリシーによって定義されます。リソースベースのポリシーで、プリンシパルを指定する必要があります。プリンシパルには、アカウント、ユーザー、ロール、フェデレーティッドユーザー、または を含めることができます AWS のサービス。

クロスアカウントアクセスを有効にするには、全体のアカウント、または別のアカウントの IAM エンティティを、リソースベースのポリシーのプリンシパルとして指定します。詳細については、IAM ユーザーガイドのIAM でのクロスアカウントリソースアクセスを参照してください。

Amazon EBS のポリシーアクション

ポリシーアクションのサポート: あり

管理者は JSON AWS ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

JSON ポリシーの `Action` 要素にはポリシー内のアクセスを許可または拒否するために使用できるアクションが記述されます。このアクションは関連付けられたオペレーションを実行するためのアクセス許可を付与するポリシーで使用されます。

Amazon EBS アクションのリストを確認するには、「Service Authorization Reference」の「Actions, resources, and condition keys for Amazon Elastic Block Store」および「Actions, resources, and condition keys for Amazon EBS」を参照してください。

Amazon EBS のポリシーアクションは、アクションの前に `ec2` または `ebs` のプレフィックスを使用します。

単一のステートメントで複数のアクションを指定するには、アクションをカンマで区切ります。

```
"Action": [
  "ec2:action1",
  "ec2:action2"
]
```

Amazon EBS のアイデンティティベースのポリシー例を確認するには、「Amazon EBS の IAM ポリシーの例」を参照してください。

Amazon EBS のポリシーリソース

ポリシーリソースのサポート: あり

管理者は JSON AWS ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

`Resource` JSON ポリシー要素はアクションが適用されるオブジェクトを指定します。ベストプラクティスとして、Amazon リソースネーム (ARN) を使用してリソースを指定します。リソースレベルのアクセス許可をサポートしないアクションの場合は、ステートメントがすべてのリソースに適用されることを示すために、ワイルドカード (*) を使用します。

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

一部の Amazon EBS API アクションは複数のリソースをサポートします。複数リソースを単一ステートメントで指定するには、ARN をカンマで区切ります。たとえば、DescribeVolumes は vol-01234567890abcdef および vol-09876543210fedcba にアクセスするため、プリンシパルには両方のリソースにアクセスする許可が必要です。

```
"Resource": [
  "arn:aws:ec2:us-east-1:123456789012:volume/vol-01234567890abcdef",
  "arn:aws:ec2:us-east-1:123456789012:volume/vol-09876543210fedcba"
]
```

Amazon EBS のポリシー条件キー

サービス固有のポリシー条件キーのサポート: あり

管理者は JSON AWS ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

Condition 要素は、定義された基準に基づいてステートメントが実行される時期を指定します。イコールや未満などの条件演算子を使用して条件式を作成して、ポリシーの条件とリクエスト内の値を一致させることができます。すべての AWS グローバル条件キーを確認するには、「IAM ユーザーガイド」のAWS 「グローバル条件コンテキストキー」を参照してください。

たとえば、次の条件では、ボリュームタイプが gp2 の場合にのみ、プリンシパルがボリュームに対してアクションを実行できます。

```
"Condition":{
  "StringLikeIfExists":{
    "ec2:VolumeType":"gp2"
  }
}
```

Amazon EBS の条件キーのリストを確認するには、「サービス認証リファレンス」の「アクション、リソース、および条件キー」を参照してください。

Amazon EBS の ACL

ACL のサポート: なし

アクセスコントロールリスト (ACL) は、どのプリンシパル (アカウントメンバー、ユーザー、または ロール) がリソースにアクセスするためのアクセス許可を持つかを制御します。ACL はリソースベースのポリシーに似ていますが、JSON ポリシードキュメント形式は使用しません。

Amazon EBS の ABAC

ABAC (ポリシー内のタグ) のサポート: 一部

属性ベースのアクセス制御 (ABAC) は、タグと呼ばれる属性に基づいてアクセス許可を定義する認可戦略です。IAM エンティティと AWS リソースにタグをアタッチし、プリンシパルのタグがリソースのタグと一致するときにオペレーションを許可するように ABAC ポリシーを設計できます。

タグに基づいてアクセスを管理するには、`aws:ResourceTag/key-name`、`aws:RequestTag/key-name`、または `aws:TagKeys` の条件キーを使用して、ポリシーの条件要素でタグ情報を提供します。

サービスがすべてのリソースタイプに対して 3 つの条件キーすべてをサポートする場合、そのサービスの値はありです。サービスが一部のリソースタイプに対してのみ 3 つの条件キーのすべてをサポートする場合、値は「部分的」になります。

ABAC の詳細については、「IAM ユーザーガイド」の「ABAC 認可でアクセス許可を定義する」を参照してください。ABAC をセットアップする手順を説明するチュートリアルについては、「IAM ユーザーガイド」の「属性ベースのアクセス制御 (ABAC) を使用する」を参照してください。

Amazon EBS での一時的な認証情報の使用

一時的な認証情報のサポート: あり

一時的な認証情報は、 AWS リソースへの短期的なアクセスを提供し、フェデレーションまたはスイッチロールの使用時に自動的に作成されます。 AWS では、長期的なアクセスキーを使用する代わりに、一時的な認証情報を動的に生成することをお勧めします。詳細については、「IAM ユーザーガイド」の「IAM の一時的な認証情報」および「AWS のサービス と IAM との連携」を参照してください。

Amazon EBS 用クロスサービスプリンシパルの許可

転送アクセスセッション (FAS) のサポート: あり

転送アクセスセッション (FAS) は、 を呼び出すプリンシパルのアクセス許可と AWS のサービス、ダウンストリームサービス AWS のサービス へのリクエストをリクエストする を使用します。FAS

リクエストを行う際のポリシーの詳細については、「転送アクセスセッション」を参照してください。

Amazon EBS 用サービスロール

サービスロールのサポート: あり

サービスロールとは、サービスがユーザーに代わってアクションを実行するために引き受ける IAM ロールです。IAM 管理者は、IAM 内からサービスロールを作成、変更、削除できます。詳細については、IAM ユーザーガイドの AWS のサービスに許可を委任するロールを作成するを参照してください。

> **Warning**
>
> サービスロールの許可を変更すると、Amazon EBS の機能が破損する恐れがあります。Amazon EBS が指示するときにのみ、サービスロールを編集してください。

Amazon EBS 用サービスリンクロール

サービスにリンクされたロールのサポート: なし

サービスにリンクされたロールは、 にリンクされたサービスロールの一種です AWS のサービス。サービスは、ユーザーに代わってアクションを実行するロールを引き受けることができます。サービスにリンクされたロールは に表示され AWS アカウント 、サービスによって所有されます。IAM 管理者は、サービスにリンクされたロールのアクセス許可を表示できますが、編集することはできません。

サービスにリンクされたロールの作成または管理の詳細については、「IAM と提携するAWS のサービス」を参照してください。表の「サービスリンクロール」列に Yes と記載されたサービスを見つけます。サービスにリンクされたロールに関するドキュメントをサービスで表示するには、[はい] リンクを選択します。

Amazon EBS の IAM ポリシーの例

デフォルトでは、ユーザーおよびロールには Amazon EBS リソースを作成または変更する許可がありません。また、、 AWS Command Line Interface (AWS CLI) AWS マネジメントコンソール、または AWS API を使用してタスクを実行することはできません。IAM 管理者は、リソースで必要なアクションを実行するための権限をユーザーに付与する IAM ポリシーを作成できます。その後、管理者はロールに IAM ポリシーを追加し、ユーザーはロールを引き受けることができます。

これらサンプルの JSON ポリシードキュメントを使用して、IAM アイデンティティベースのポリシーを作成する方法については、IAM ユーザーガイドの IAM ポリシーの作成 を参照してください。

トピック

- ポリシーに関するベストプラクティス
- ユーザーに Amazon EBS コンソールの使用を許可
- 自分の権限の表示をユーザーに許可する
- ユーザーにボリュームの操作を許可
- ユーザーにスナップショットの操作を許可

ポリシーに関するベストプラクティス

アイデンティティベースのポリシーは、あるユーザーがアカウントの Amazon EBS リソースを作成、アクセス、削除できるどうかを決定します。これらのアクションでは、 AWS アカウントに費用が発生する場合があります。アイデンティティベースポリシーを作成したり編集したりする際には、以下のガイドラインと推奨事項に従ってください:

- AWS 管理ポリシーを開始し、最小特権のアクセス許可に移行する – ユーザーとワークロードにアクセス許可の付与を開始するには、多くの一般的なユースケースにアクセス許可を付与するAWS 管理ポリシーを使用します。これらは で使用できます AWS アカウント。ユースケースに固有の AWS カスタマー管理ポリシーを定義することで、アクセス許可をさらに減らすことをお勧めします。詳細については、IAM ユーザーガイド の AWS マネージドポリシー または ジョブ機能のAWS マネージドポリシー を参照してください。
- 最小特権を適用する – IAM ポリシーでアクセス許可を設定する場合は、タスクの実行に必要な許可のみを付与します。これを行うには、特定の条件下で特定のリソースに対して実行できるアクションを定義します。これは、最小特権アクセス許可とも呼ばれています。IAM を使用して許可を適用する方法の詳細については、IAM ユーザーガイド の IAM でのポリシーとアクセス許可 を参照してください。
- IAM ポリシーで条件を使用してアクセスをさらに制限する - ポリシーに条件を追加して、アクションやリソースへのアクセスを制限できます。たとえば、ポリシー条件を記述して、すべてのリクエストを SSL を使用して送信するように指定できます。条件を使用して、サービスアクションが などの特定の を通じて使用されている場合に AWS のサービス、サービスアクションへのアクセスを許可することもできます CloudFormation。詳細については、IAM ユーザーガイド の IAM JSON ポリシー要素:条件 を参照してください。
- IAM アクセスアナライザー を使用して IAM ポリシーを検証し、安全で機能的な権限を確保する - IAM アクセスアナライザー は、新規および既存のポリシーを検証して、ポリシーが IAM ポリシー

言語 (JSON) および IAM のベストプラクティスに準拠するようにします。IAM アクセスアナライザーは 100 を超えるポリシーチェックと実用的な推奨事項を提供し、安全で機能的なポリシーの作成をサポートします。詳細については、IAM ユーザーガイド の IAM Access Analyzer でポリシーを検証する を参照してください。

- 多要素認証 (MFA) を要求する – で IAM ユーザーまたはルートユーザーを必要とするシナリオがある場合は AWS アカウント、MFA をオンにしてセキュリティを強化します。API オペレーションが呼び出されるときに MFA を必須にするには、ポリシーに MFA 条件を追加します。詳細については、IAM ユーザーガイド の MFA を使用した安全な API アクセス を参照してください。

IAM でのベストプラクティスの詳細については、IAM ユーザーガイド の IAM でのセキュリティのベストプラクティス を参照してください。

ユーザーに Amazon EBS コンソールの使用を許可

Amazon Elastic Block Store コンソールにアクセスするには、最小限の許可セットが必要です。これらのアクセス許可により、 の Amazon EBS リソースの詳細を一覧表示および表示できます AWS アカウント。最小限必要な許可よりも制限が厳しいアイデンティティベースのポリシーを作成すると、そのポリシーを持つエンティティ (ユーザーまたはロール) に対してコンソールが意図したとおりに機能しません。

AWS CLI または AWS API のみを呼び出すユーザーには、最小限のコンソールアクセス許可を付与する必要はありません。代わりに、実行しようとしている API オペレーションに一致するアクションのみへのアクセスが許可されます。

ユーザーとロールが引き続き Amazon EBS コンソールを使用できるようにするには、エンティティに Amazon EBS *ConsoleAccess*または *ReadOnly* AWS 管理ポリシーもアタッチします。詳細については、「IAM ユーザーガイド」の「ユーザーへのアクセス許可の追加」を参照してください。

自分の権限の表示をユーザーに許可する

この例では、ユーザーアイデンティティにアタッチされたインラインおよびマネージドポリシーの表示を IAM ユーザーに許可するポリシーの作成方法を示します。このポリシーには、コンソールで、または AWS CLI または AWS API を使用してプログラムでこのアクションを実行するアクセス許可が含まれています。

```
{
    "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": "*"
        }
    ]
}
```

ユーザーにボリュームの操作を許可

例

- 例: ボリュームのアタッチとデタッチ
- 例: ボリュームの作成
- 例: タグ付きのボリュームの作成
- 例: Amazon EC2 コンソールを使用したボリュームの操作

例: ボリュームのアタッチとデタッチ

API アクションが複数のリソースを指定するために発信者を必要とする場合、ユーザーがすべての必要なリソースにアクセスできるようにポリシーステートメントを作成する必要があります。1 つ以上

のリソースで Condition エレメントを使用する必要がある場合、この例のとおり複数のステートメントを作成する必要があります。

以下のポリシーでは、ユーザーがタグvolume_user=iam-user-name」の付いたボリュームを、タグdepartment=devの付いたインスタンスにアタッチしたり、またインスタンスからボリュームをデタッチしたりできるようにします。このポリシーを IAM グループにアタッチする場合、aws:username ポリシー変数によってグループのユーザーに、値としてユーザー名を持つタグ名が volume_user のインスタンスからボリュームをアタッチまたはデタッチするための許可が付与されます。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AttachVolume",
                "ec2:DetachVolume"
            ],
            "Resource": "arn:aws:ec2:us-east-1:111122223333:instance/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/department": "dev"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AttachVolume",
                "ec2:DetachVolume"
            ],
            "Resource": "arn:aws:ec2:us-east-1:111122223333:volume/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/volume_user": "${aws:username}"
                }
            }
        }
    ]
```

```
}
```

例: ボリュームの作成

次のポリシーでは、ユーザーが CreateVolume API アクションを使用することができます。ユーザーは、ボリュームが暗号化されていて、ボリューム サイズが 20 GiB 未満の場合にのみボリュームの作成を許可されます。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateVolume"
            ],
            "Resource": "arn:aws:ec2:us-east-1:111122223333:volume/*",
            "Condition": {
                "NumericLessThan": {
                    "ec2:VolumeSize": "20"
                },
                "Bool": {
                    "ec2:Encrypted": "true"
                }
            }
        }
    ]
}
```

例: タグ付きのボリュームの作成

次のポリシーには、タグ aws:RequestTag および costcenter=115 を使用して作成したすべてのボリュームへのタグ付けをユーザーに求める stack=prod 条件キーが含まれています。ユーザーがこれらのタグを渡さないか、タグをまったく指定しない場合、リクエストは失敗します。

タグを適用するリソース作成アクションでは、ユーザーが CreateTags アクションを使用するアクセス権限を持っていることも必要です。2 番目のステートメントは、ec2:CreateAction 条件キー

を使用して、ユーザーが CreateVolume のコンテキストでみタグを使用できるようにします。ユーザーは、既存のボリュームにも他のリソースにもタグ付けできません。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCreateTaggedVolumes",
            "Effect": "Allow",
            "Action": "ec2:CreateVolume",
            "Resource": "arn:aws:ec2:us-east-1:111122223333:volume/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/costcenter": "115",
                    "aws:RequestTag/stack": "prod"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:us-east-1:111122223333:volume/*",
            "Condition": {
                "StringEquals": {
                    "ec2:CreateAction": "CreateVolume"
                }
            }
        }
    ]
}
```

次のポリシーでは、ユーザーがタグを指定しなくてもボリュームを作成することができます。CreateTags アクションは、タグが CreateVolume リクエストで指定されている場合にのみ評価されます。ユーザーがタグを指定する場合、purpose=test タグを指定する必要があります。リクエストでは他のタグは許可されません。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:CreateVolume",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:us-east-1:111122223333:volume/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/purpose": "test",
                    "ec2:CreateAction": "CreateVolume"
                },
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": "purpose"
                }
            }
        }
    ]
}
```

例: Amazon EC2 コンソールを使用したボリュームの操作

次のポリシーは、ボリュームの表示および作成、ならびに Amazon EC2 コンソールを使用して特定のインスタンスにボリュームのアタッチおよびデタッチする許可をユーザーに付与します。

ユーザーは、「purpose=test」というタグを含むインスタンスに対してどのボリュームもアタッチできます。同様に、それらのインスタンスからボリュームをデタッチすることもできます。Amazon EC2 コンソールを使用してボリュームをアタッチするには、ユーザーに ec2:DescribeInstances アクションを使用するアクセス許可があると、[Attach Volume] ダイアログボックスのあらかじめ用意されたリストからインスタンスを選択できるため、役立ちます。ただ

し、これにより、コンソールの [Instances] ページでもすべてのインスタンスが表示されるため、このアクションを省略することもできます。

最初のステートメントでは、 ec2:DescribeAvailabilityZones アクションは、ボリュームの作成時にユーザーがアベイラビリティーゾーンを選択できるようにするために必要です。

ユーザーは、作成したボリュームをタグ付けできません (ボリュームの作成中も作成後も)。

ユーザーにスナップショットの操作を許可

以下に、CreateSnapshot (EBS ボリュームのポイントインタイムスナップショット) と CreateSnapshots (マルチボリュームスナップショット) の両方のポリシーの例を示しています。

例

- 例: スナップショットの作成
- 例: スナップショットの作成
- 例: タグ付きのスナップショットの作成
- 例: タグを使用してマルチボリュームスナップショットを作成する
- 例: スナップショットのコピー
- 例: スナップショットのアクセス許可設定の変更

例: スナップショットの作成

次のポリシーでは、お客様が CreateSnapshot API アクションを使用することができます。お客様は、ボリュームが暗号化されていて、ボリューム サイズが 20 GiB 未満の場合にのみスナップショットを作成できます。

JSON

```
{
    "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-1:111122223333:volume/*",
            "Condition": {
                "NumericLessThan": {
                    "ec2:VolumeSize": "20"
                },
                "Bool": {
                    "ec2:Encrypted": "true"
                }
            }
        }
    ]
}
```

例: スナップショットの作成

次のポリシーでは、お客様が CreateSnapshot API アクションを使用することができます。インスタンス上のすべてのボリュームがタイプ GP2 の場合にのみ、お客様はスナップショットを作成できます。

JSON

```
{
   "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"
```

```
            }
        }

         }
    ]
}
```

例: タグ付きのスナップショットの作成

次のポリシーには、タグ aws:RequestTag および costcenter=115 をすべての新しいリクエストに適用することをお客様に求める stack=prod 条件キーが含まれています。ユーザーがこれらのタグを渡さないか、タグをまったく指定しない場合、リクエストは失敗します。

タグを適用するリソース作成アクションでは、CreateTags アクションを使用するアクセス権限も持っていることが求められます。3 番目のステートメントは、ec2:CreateAction 条件キーを使用して、お客様が CreateSnapshot のコンテキストでみタグを使用できるようにします。お客様は、既存のボリュームにも他のリソースにもタグ付けできません。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:CreateSnapshot",
            "Resource": "arn:aws:ec2:us-east-1:111122223333: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"
                }
            }
        }
    ]
}
```

例: タグを使用してマルチボリュームスナップショットを作成する

次のポリシーには、マルチボリュームスナップショットセットを作成するときにタグ costcenter=115 および stack=prod を適用することをお客様に要求する aws:RequestTag 条件キーが含まれています。ユーザーがこれらのタグを渡さないか、タグをまったく指定しない場合、リクエストは失敗します。

JSON

```
{
   "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"
            }
         }
      }
   ]
}
```

次のポリシーでは、お客様がタグを指定しなくてもスナップショットを作成することができます。CreateTags アクションは、タグが CreateSnapshot または CreateSnapshots リクエストで指定されている場合にのみ評価されます。リクエストでは、タグを省略できます。タグを指定する場合、タグは purpose=test である必要があります。リクエストでは他のタグは許可されません。

JSON

```
{
   "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"
            }
         }
      }
   ]
}
```

次のポリシーでは、お客様がタグを指定しなくても、マルチボリュームスナップショットセットを作成することができます。CreateTags アクションは、タグが CreateSnapshot または CreateSnapshots リクエストで指定されている場合にのみ評価されます。リクエストでは、タグを省略できます。タグを指定する場合、タグは purpose=test である必要があります。リクエストでは他のタグは許可されません。

JSON

```
{
   "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"
            }
         }
      }
   ]
```

```
}
```

次のポリシーでは、ソースボリュームにお客様の User:*username* がタグ付けされていて、スナップショット自体に Environment:Dev と User:*username* がタグ付けされている場合にのみスナップショットの作成を許可します。お客様は、スナップショットにタグを追加できます。

次の CreateSnapshots のポリシーでは、ソースボリュームにお客様用の User:*username* がタグ付けされ、スナップショット自体に Environment:Dev と User:*username* のタグ付けがされている場合にのみスナップショットを作成できます。

JSON

```
{
    "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-1:111122223333: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/*"
        }
    ]
}
```

次のポリシーでは、スナップショットにお客様の User:username がタグ付けされている場合のみスナップショットの削除を許可します。

JSON

```
{
   "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}"
            }
         }
      }
   ]
}
```

次のポリシーでは、お客様はスナップショットを作成できますが、作成されるスナップショットにタグキー `value=stack` が付いている場合はアクションが拒否されます。

JSON

```
{
   "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"
            }
         }
      }
   ]
}
```

次のポリシーでは、お客様はスナップショットを作成できますが、作成されるスナップショットにタグキー value=stack が付いている場合はアクションが拒否されます。

JSON

```
{
   "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"
            }
         }
      }
   ]
}
```

次のポリシーでは、複数のアクションを単一のポリシーにまとめることができます。スナップショットがリージョン CreateSnapshots で作成された場合にのみ、(us-east-1 のコンテキスト内で) スナップショットを作成できます。スナップショットがリージョン CreateSnapshots に作成されている場合、およびインスタンスタイプが us-east-1 の場合にのみ、スナップショットを作成できます (t2* のコンテキスト内で)。

JSON

```
{
   "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.*"]
            }
         }
      }
   ]
```

```
}
```

例: スナップショットのコピー

CopySnapshot アクション用に指定されたリソースレベルのアクセス許可は、新しいスナップショットとソーススナップショップの両方に適用されます。

以下のポリシーの例では、新しいスナップショットがタグキー purpose、タグ値 production (purpose=production) を使用して作成された場合にのみ、プリンシパルがスナップショットをコピーすることを許可します。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCopySnapshotWithTags",
            "Effect": "Allow",
            "Action": "ec2:CopySnapshot",
            "Resource": "arn:aws:ec2:*:111122223333:snapshot/*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/purpose": "production"
                }
            }
        }
    ]
}
```

次のポリシー例では、ソーススナップショットが AWS アカウント によって所有されている場合にのみ、プリンシパルがスナップショットをコピーすることを許可します123456789012。

以下のポリシーの例では、ソーススナップショットが CSISnapshotName のタグキーを使用して作成された場合にのみ、プリンシパルがスナップショットをコピーすることを許可します。

```
{
    "Effect": "Allow",
    "Action": "ec2:CopySnapshot",
    "Resource": "arn:aws:ec2:*::snapshot/${*}",
```

```
    "Condition": {
        "StringLike": {
            "aws:RequestTag/CSISnapshotName": "*"
         }
     }
},

{
    "Effect": "Allow",
    "Action": "ec2:CopySnapshot",
    "Resource": "arn:aws:ec2:*::snapshot/snap-*"
}
```

例: スナップショットのアクセス許可設定の変更

次のポリシーでは、スナップショットが でタグ付けされている場合にのみスナップショットの変更を許可します。User:*username#####*はお客様の AWS アカウントのユーザー名です。この条件が満たされない場合、リクエストは失敗します。

JSON

```
{
   "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}"
            }
         }
      }
   ]
}
```

Amazon EBS の認証問題のトラブルシューティング

次の情報を使用し、Amazon EBS および IAM の操作時に発生する可能性がある一般的な問題の診断や解決に役立ててください。

問題

- Amazon EBS でアクションを実行する権限がありません
- iam:PassRole を実行する権限がありません
- 自分の 以外のユーザーに Amazon EBS リソース AWS アカウント へのアクセスを許可したい

Amazon EBS でアクションを実行する権限がありません

にアクションを実行する権限がないと AWS マネジメントコンソール 通知された場合は、管理者に連絡してサポートを依頼する必要があります。管理者とは、サインイン認証情報を提供した担当者です。

次のエラー例は、mateojackson IAM ユーザーがコンソールを使用してボリュームに関する詳細を表示しようとしても、ec2:DescribeVolumes 許可がない場合に発生します。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 ec2:DescribeVolumes on resource: volume-id
```

この場合、Mateo は AWS 管理者にボリュームの説明を許可するように依頼します。

iam:PassRole を実行する権限がありません

iam:PassRole アクションを実行する権限がないというエラーが表示された場合、ポリシーを更新して Amazon EBS にロールを渡せるようにする必要があります。

一部の AWS のサービス では、新しいサービスロールまたはサービスにリンクされたロールを作成する代わりに、そのサービスに既存のロールを渡すことができます。そのためには、サービスにロールを渡すアクセス許可が必要です。

次のエラー例は、marymajor という IAM ユーザーがコンソールを使用して Amazon EBS でアクションを実行しようとする場合に発生します。ただし、このアクションをサービスが実行するには、サービスロールから付与されたアクセス許可が必要です。Mary には、ロールをサービスに渡すアクセス許可がありません。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

この場合、Mary のポリシーを更新してメアリーに iam:PassRole アクションの実行を許可する必要があります。

サポートが必要な場合は、 AWS 管理者にお問い合わせください。サインイン資格情報を提供した担当者が管理者です。

自分の 以外のユーザーに Amazon EBS リソース AWS アカウント へのアクセスを許可したい

他のアカウントのユーザーや組織外の人が、リソースにアクセスするために使用できるロールを作成できます。ロールの引き受けを委託するユーザーを指定できます。リソースベースのポリシーまたはアクセスコントロールリスト (ACL) をサポートするサービスの場合、それらのポリシーを使用して、リソースへのアクセスを付与できます。

詳細については、以下を参照してください:

- Amazon EBS がこれらの機能をサポートしているかどうかを確認するには、「Amazon EBS と IAM が連携する仕組み」を参照してください。
- 所有 AWS アカウント している のリソースへのアクセスを提供する方法については、「IAM ユーザーガイド」の「所有 AWS アカウント している別の の IAM ユーザーへのアクセスを提供する」を参照してください。
- リソースへのアクセスをサードパーティーに提供する方法については AWS アカウント、IAM ユーザーガイドの「サードパーティー AWS アカウント が所有する へのアクセスを提供する」を参照してください。
- ID フェデレーションを介してアクセスを提供する方法については、IAM ユーザーガイド の 外部で認証されたユーザー (ID フェデレーション) へのアクセスの許可 を参照してください。
- クロスアカウントアクセスにおけるロールとリソースベースのポリシーの使用方法の違いについては、「IAM ユーザーガイド」の「IAM でのクロスアカウントのリソースへのアクセス」を参照してください。

Amazon EBS のコンプライアンス検証

新しい AWS サインアップエクスペリエンスは、規制されたワークロード向けには設計されていません。新しい AWS サインアップエクスペリエンスを使用しているが、規制対象のワークロード AWS に を使用する場合は、 AWS (アドバンスド) にサインアップするか、 AWS 環境の高度な機能をアクティブ化できます。

AWS のサービス が特定のコンプライアンスプログラムの範囲内にあるかどうかを確認するには、「コンプライアンスAWS のサービス プログラムによるスコープ」の「コンプライアンス」を参照し

て、関心のあるコンプライアンスプログラムを選択します。一般的な情報については、AWS「コンプライアンスプログラム」を参照してください。

を使用して、サードパーティーの監査レポートをダウンロードできます AWS Artifact。詳細については、「Downloading Reports in AWS Artifact」を参照してください。

を使用する際のお客様のコンプライアンス責任 AWS のサービス は、お客様のデータの機密性、貴社のコンプライアンス目的、適用可能な法律および規制によって決まります。を使用する際のコンプライアンス責任の詳細については AWS のサービス、AWS「 セキュリティドキュメント」を参照してください。

Amazon EBS のデータ回復力

AWS グローバルインフラストラクチャは、 AWS リージョン およびアベイラビリティーゾーンを中心に構築されています。 は、低レイテンシー、高スループット、高度に冗長なネットワークで接続された、物理的に分離および分離された複数のアベイラビリティーゾーン AWS リージョン を提供します。アベイラビリティーゾーンでは、ゾーン間で中断することなく自動的にフェールオーバーするアプリケーションとデータベースを設計および運用することができます。アベイラビリティーゾーンは、従来の単一または複数のデータセンターインフラストラクチャよりも可用性、フォールトトレランス、および拡張性が優れています。

AWS リージョン およびアベイラビリティーゾーンの詳細については、AWS「 グローバルインフラストラクチャ」を参照してください。

Amazon EBS は、 AWS グローバルインフラストラクチャに加えて、データの耐障害性とバックアップのニーズをサポートするのに役立ついくつかの機能を提供しています。

- Amazon Data Lifecycle Managerを使用して EBS スナップショットを自動化する機能
- リージョン間の EBS スナップショットをコピーする機能

Amazon EBS のモニタリングツール

モニタリングは、Amazon Elastic Block Store およびその他の AWS ソリューションの信頼性、可用性、パフォーマンスを維持する上で重要な部分です。 には、Amazon EBS を監視し、問題が発生したときに報告し、必要に応じて自動アクションを実行するための以下のモニタリングツール AWS が用意されています。

- AWS CloudTrail は、 によって、または に代わって行われた API コールおよび関連イベントをキャプチャ AWS アカウント し、指定した Amazon S3 バケットにログファイルを配信します。呼び出し元のユーザーとアカウント AWS、呼び出し元の送信元 IP アドレス、呼び出しの発生日時を特定できます。EBS ボリュームとスナップショットを管理する API は、Amazon EC2 API の一部です。CloudTrail と Amazon EC2 API の詳細については、「Amazon EC2 ユーザーガイド」の「AWS CloudTrailを使用して Amazon EC2 API コールをログに記録」を参照してください。
- Amazon CloudWatch は、 AWS リソースと で実行されるアプリケーションを AWS リアルタイムでモニタリングします。メトリクスの収集と追跡、カスタマイズしたダッシュボードの作成、および指定したメトリクスが指定したしきい値に達したときに通知またはアクションを実行するアラームの設定を行うことができます。例えば、CloudWatch で Amazon EC2 インスタンスの CPU 使用率などのメトリクスを追跡し、必要に応じて新しいインスタンスを自動的に起動できます。詳細については、「the section called “Amazon CloudWatch”」を参照してください。
- Amazon EventBridge を使用して AWS サービスを自動化し、アプリケーションの可用性の問題やリソースの変更などのシステムイベントに自動的に対応できます。 AWS サービスからのイベントは、ほぼリアルタイムで EventBridge に配信されます。簡単なルールを記述して、注目するイベントと、イベントがルールに一致した場合に自動的に実行するアクションを指定できます。詳細については、「the section called “Amazon EventBridge”」を参照してください。
- Amazon EBS の詳細なパフォーマンス統計は、Nitro ベースの Amazon EC2 インスタンスにアタッチされた Amazon EBS ボリュームに対するリアルタイムの I/O パフォーマンス統計を提供します。詳細については、Amazon EBS の詳細なパフォーマンス統計 を参照してください。
- Amazon GuardDuty は、EC2 インスタンスで悪意の可能性のあるアクティビティを検出するのに役立ちます。GuardDuty Malware Protection for EC2 は、EC2 インスタンスにアタッチされた EBS ボリュームをスキャンします。詳細については、「the section called “Amazon GuardDuty”」を参照してください。

Amazon EBS の Amazon CloudWatch メトリクス

Amazon CloudWatch メトリクスは、ボリュームの実行動作を表示または分析したり、それらの実行動作についてのアラームを設定したりするために使用できる統計データです。

データは無料で 1 分間隔で自動的に取得されます。

CloudWatch からデータを取得したときに、`Period` リクエストパラメータを含めて、返されるデータの詳細程度を指定できます。これは、EBS がデータを収集するために使用する期間 (1 分) とは異なります。リクエストで収集期間以上の期間を指定して、返されたデータが有効であることを確認することをお勧めします。

データは、CloudWatch API または Amazon EC2 コンソールのいずれかを使用して取得できます。コンソールは CloudWatch API から未加工データを取得し、そのデータに基づいて一連のグラフを表示します。必要に応じて、API のデータまたはコンソールのグラフのいずれかを使用できます。

トピック

- Amazon EBS ボリュームのメトリクス
- Amazon EBS スナップショットのメトリクス
- Nitro インスタンスのメトリクス
- 高速スナップショット復元のメトリクス
- Amazon EC2 コンソールのグラフ

Amazon EBS ボリュームのメトリクス

`AWS/EBS` 名前空間には、すべてのインスタンスタイプにアタッチされている EBS ボリュームの、次のメトリクスが含まれます。すべての Amazon EBS ボリュームタイプは、ボリュームがインスタンスにアタッチされている場合にのみ、1 分間のメトリクスを自動的に CloudWatch に送信します。

インスタンスのオペレーティングシステムから使用可能なディスク領域に関する情報を取得するには、空きディスク容量の表示を参照してください。

> **Note**
>
> いくつかのメトリクスでは、Nitro System 上に構築されたインスタンスによって違いが生じます。これらのインスタンスタイプのリストについては、「Nitro System 上に構築されたインスタンス」を参照してください。

メトリクス	説明	単位	ディメンション	有意義な統計
VolumeAvgIOPS *	**Note** Nitro インスタンスにアタッチされたすべての EBS ボリュームタイプでサポートされています。Amazon ECS および AWS Fargate タスクにアタッチされたボリュームについては公開されていません。 1 分間にボリュームに駆動される読み取りおよび書き込み IOPS の平均値。過去 1 分間にボリュームに駆動されたオペレーションがない場合、メトリクスの値はゼロ (0) になります。詳細については、「CloudWatch を使用して I/O 特性を監視する」を参照してください。 マルチアタッチが有効なボリュームの場合、InstanceId ディメンションを使用して、	OPS/秒	VolumeId | InstanceId	• Sum • Average • Minimum | Maximum

メトリクス	説明	単位	ディメンション	有意義な統計
	特定のボリュームインスタンスアタッチメントの平均 IOPS を表示します。			

メトリクス	説明	単位	ディメンション	有意義な統計
VolumeAvgThroughput *	**Note** Nitro インスタンスにアタッチされたすべての EBS ボリュームタイプでサポートされています。Amazon ECS および AWS Fargate タスクにアタッチされたボリュームについては公開されていません。 1 分間にボリュームに駆動される読み取りおよび書き込みスループットの平均値。過去 1 分間にボリュームに駆動されたオペレーションがない場合、メトリクスの値はゼロ (0) になります。詳細については、「CloudWatch を使用して I/O 特性を監視する」を参照してください。 マルチアタッチが有効なボリュームの場合、InstanceId ディメンションを使用して、	KiB/秒	VolumeId | InstanceId	• Sum • Average • Minimum | Maximum

メトリクス	説明	単位	ディメンション	有意義な統計
	特定のボリュームインスタンスアタッチメントの平均スループットを表示します。			

メトリクス	説明	単位	ディメンション	有意義な統計
VolumeAvgReadLatency *	**Note** Nitro インスタンスにアタッチされたすべてのボリュームタイプでサポートされています。Amazon ECS および AWS Fargate タスクにアタッチされたボリュームについては公開されていません。 1 分間に読み取りオペレーションを完了するのにかかる平均時間。このメトリクスを使用して、Amazon EC2 インスタンスにアタッチされた EBS ボリュームの平均 I/O レイテンシーをモニタリングします。平均値は、過去 1 分間に完了した I/O オペレーションに基づいて計算されます。過去 1 分以内に完了したオペレーションがない場合、メトリクスの値はゼロになります。	ミリ秒	VolumeId | InstanceId	Minimum | Maximum

メトリクス	説明	単位	ディメンション	有意義な統計
	マルチアタッチが有効なボリュームの場合、InstanceId ディメンションを使用して、特定のボリュームインスタンスアタッチメントの平均レイテンシーを表示します。			

メトリクス	説明	単位	ディメンション	有意義な統計
VolumeAvgWriteLatency *	**Note** Nitro インスタンスにアタッチされたすべてのボリュームタイプでサポートされています。Amazon ECS および AWS Fargate タスクにアタッチされたボリュームについては公開されていません。 1 分間に書き込みオペレーションを完了するのにかかる平均時間。このメトリクスを使用して、Amazon EC2 インスタンスにアタッチされた EBS ボリュームの平均 I/O レイテンシーをモニタリングします。平均値は、過去 1 分間に完了した I/O オペレーションに基づいて計算されます。過去 1 分以内に完了したオペレーションがない場合、メトリクスの値はゼロになります。	ミリ秒	VolumeId | InstanceId	Minimum | Maximum

メトリクス	説明	単位	ディメンション	有意義な統計
	マルチアタッチが有効なボリュームの場合、InstanceId ディメンションを使用して、特定のボリュームインスタンスアタッチメントの平均レイテンシーを表示します。			

メトリクス	説明	単位	ディメンション	有意義な統計
VolumeIOPSExceededCheck *	**Note** Nitro インスタンスにアタッチされたマグネティック (standard) を除くすべてのボリュームタイプでサポートされています。マルチアタッチが有効なボリュームではサポートされません。Amazon ECS および AWS Fargate タスクにアタッチされたボリュームについては公開されていません。 アプリケーションが過去 1 分以内にボリュームのプロビジョンド IOPS パフォーマンスを超える IOPS を一貫して駆動しようとしたかどうかをレポートします。このメトリクスは、0 (プロビジョンド IOPS を超えていない) または 1 (プロビジョンド IOPS を超えて	なし	VolumeId | InstanceId	• Sum • Average • Minimum | Maximum

メトリクス	説明	単位	ディメンション	有意義な統計
	いない) のいずれかの値をとります。詳細については、「CloudWatch を使用して I/O 特性を監視する」を参照してください。			

メトリクス	説明	単位	ディメンション	有意義な統計
VolumeThroughputExceededCheck *	**Note** Nitro インスタンスにアタッチされたマグネティック (standard) を除くすべてのボリュームタイプでサポートされています。マルチアタッチが有効なボリュームではサポートされません。Amazon ECS および AWS Fargate タスクにアタッチされたボリュームについては公開されていません。 アプリケーションが過去 1 分以内にボリュームのプロビジョンドスループットのパフォーマンスを超えるスループットを一貫して駆動しようとしたかどうかをレポートします。このメトリクスは、0 (プロビジョンドスループットを超えていない) または 1 (プロビジョ	なし	VolumeId | InstanceId	• Sum • Average • Minimum | Maximum

メトリクス	説明	単位	ディメンション	有意義な統計
	ンドスループットを超えている) のいずれかの値をとります。詳細については、「CloudWatch を使用して I/O 特性を監視する」を参照してください。			
VolumeReadBytes	指定された期間の読み取りオペレーションに関する情報を提供します。 • Sum 統計は、期間内に転送されたバイトの総数をレポートします。 • Average 統計は、指定した期間に読み込まれたバイト数の平均をレポートします。 • SampleCount 統計は、統計計算で使用されるデータポイントの数を表します。 Note Xen インスタンスでは、ボリュームに読み取りアクティビティがある場合にのみデータが報告されます。	バイト	VolumeId	• Average • Sum • SampleCount • Minimum | Maximum — Nitro ベースのインスタンスにアタッチされたボリュームのみ

メトリクス	説明	単位	ディメンション	有意義な統計
VolumeWriteBytes	指定された期間の書き込みオペレーションに関する情報を提供します。 • Sum 統計は、期間内に転送されたバイトの総数をレポートします。 • Average 統計は、指定した期間に書き込まれたバイト数の平均です。 • SampleCount 統計は、統計計算で使用されるデータポイントの数を表します。 Note Xen インスタンスでは、ボリュームに書き込みアクティビティがある場合にのみデータが報告されます。	バイト	VolumeId	• Average • Sum • SampleCount • Minimum | Maximum — Nitro ベースのインスタンスにアタッチされたボリュームのみ

メトリクス	説明	単位	ディメンション	有意義な統計
VolumeReadOps	指定期間内の読み取りオペレーションの総数。読み取りオペレーションは、完了時にカウントされます。その期間の 1 秒あたりの読み込み I/O 操作回数 (読み取り IOPS) の平均を算出するには、その期間の読み取りオペレーション回数の合計をその期間の秒数で割ります。	カウント	VolumeId	• Average • Sum • Minimum | Maximum — Nitro ベースのインスタンスにアタッチされたボリュームのみ
VolumeWriteOps	指定期間内の書き込みオペレーションの総数。書き込みオペレーションは、完了時にカウントされます。その期間の 1 秒あたりの書き込み I/O 操作回数 (書き込み IOPS) の平均を算出するには、その期間の書き込みオペレーション回数の合計をその期間の秒数で割ります。	カウント	VolumeId	• Average • Sum • Minimum | Maximum — Nitro ベースのインスタンスにアタッチされたボリュームのみ

メトリクス	説明	単位	ディメンション	有意義な統計
VolumeTotalReadTime	**Note** マルチアタッチが有効なボリュームではサポートされません。Xen インスタンスでは、ボリュームに読み取りアクティビティがある場合にのみデータが報告されます。 指定期間内に完了した操作すべての読み取りオペレーションに要した時間 (秒) の合計。複数のリクエストが同時に送信された場合は、この合計が期間の長さを超えることがあります。例えば、期間が 1 分間 (60 秒) で、その期間内に完了した操作の数が 150 あり、1 つの操作に 1 秒かかるとすれば、この値は 150 秒となります。	秒	VolumeId	• Average — Nitro ベースのインスタンスにアタッチされたボリュームは対象外 • Sum • Minimum | Maximum — Nitro ベースのインスタンスにアタッチされたボリュームのみ

メトリクス	説明	単位	ディメンション	有意義な統計
VolumeTotalWriteTime	**Note** マルチアタッチが有効なボリュームではサポートされません。Xen インスタンスでは、ボリュームに書き込みアクティビティがある場合にのみデータが報告されます。 指定期間内に完了した操作すべての、書き込みオペレーションに要した時間 (秒) の合計。複数のリクエストが同時に送信された場合は、この合計が期間の長さを超えることがあります。例えば、期間が 1 分間 (60 秒) で、その期間内に完了した操作の数が 150 あり、1 つの操作に 1 秒かかるとすれば、この値は 150 秒となります。	秒	VolumeId	• Average — Nitro ベースのインスタンスにアタッチされたボリュームは対象外 • Sum • Minimum | Maximum — Nitro ベースのインスタンスにアタッチされたボリュームのみ

メトリクス	説明	単位	ディメンション	有意義な統計
VolumeIdleTime	**Note** マルチアタッチが有効なボリュームではサポートされません。 指定期間内に、読み取りと書き込みのどちらの操作も行われなかった時間 (秒) の合計。	秒	VolumeId	• Average — Nitro ベースのインスタンスにアタッチされたボリュームは対象外 • Sum • Minimum | Maximum — Nitro ベースのインスタンスにアタッチされたボリュームのみ

メトリクス	説明	単位	ディメンション	有意義な統計
VolumeQueueLength	指定期間内に完了を待っていた読み取りおよび書き込みの操作リクエストの数。	カウント	VolumeId	• Average • Sum — Nitro インスタンスにアタッチされたボリュームは対象外 • Minimum | Maximum — Nitro インスタンスにアタッチされたボリュームのみ

メトリクス	説明	単位	ディメンション	有意義な統計
VolumeStalledIOCheck *	**Note** Nitro インスタンスのみが対象です。Amazon ECS と AWS Fargate タスクにアタッチされたボリュームについては公開されていません。 過去 1 分間にボリュームが停止した IO チェックに合格または失敗したかについてレポートします。このメトリクスは、0 (合格) または 1 (失敗) のいずれかの値をとります。詳細については、「CloudWatch を使用して I/O 特性を監視する」を参照してください。	なし	VolumeId | InstanceId	• 合計 • 平均 • 最小値 • 最大値

メトリクス	説明	単位	ディメンション	有意義な統計
VolumeThroughputPercentage	**Note** プロビジョンド IOPS SSD ボリュームのみが対象です。マルチアタッチが有効なボリュームではサポートされません。 Amazon EBS ボリュームにプロビジョニングされた合計 IOPS (1 秒間あたりの I/O 操作回数) に対する、配信された IOPS の割合 (パーセント)。プロビジョンド IOPS SSD ボリュームは、期間の 99.9 パーセントにわたり、プロビジョニングされたパフォーマンスを実現します。書き込みの間、他に保留中の I/O リクエストが 1 分以内になければ、メトリクス値は 100% となります。また、お客様が行ったアクション (例えば使用率ピーク時にボリュームのスナップショットを作成する、EBS 最適化イン	割合 (%)	VolumeId	• Average • Minimum | Maximum

メトリクス	説明	単位	ディメンション	有意義な統計
	スタンス以外でボリュームを実行する、そのボリュームのデータに初めてアクセスするなど) によってボリュームの I/O 性能が一時的に低下する場合があります。			
VolumeConsumedReadWriteOps	**Note** プロビジョンド IOPS SSD ボリュームのみが対象です。 指定された期間内に消費された読み書き操作の合計数 (256K キャパシティーユニットに標準化) 。それぞれ 256K より小さい I/O 操作は、1 消費 IOPS とカウントされます。256K より大きい I/O 操作は、256K キャパシティーユニットでカウントされます。例えば、1024K I/O は 4 消費 IOPS としてカウントされます。	カウント	VolumeId	• Average • Sum • Minimum | Maximum

メトリクス	説明	単位	ディメンション	有意義な統計
BurstBalance	**Note** gp2、st1、および sc1 ボリュームのみ。 バーストバケットに残っている I/O クレジット (gp2 用) またはスループットクレジット (st1 と sc1 用) の割合に関する情報を提供します。データは、ボリュームがアクティブな場合にのみ CloudWatch に報告されます。ボリュームがアタッチされていない場合、データは報告されません。ボリュームのベースラインパフォーマンスが最大バーストパフォーマンスを超える場合、クレジットは消費されません。その他のインスタンスの場合、報告されるバーストバランスは 100% です。詳細については、「gp2 ボリュームのパフォーマンス」を参照してください。	割合 (%)	VolumeId	• Average • Sum — Nitro インスタンスにアタッチされたボリュームは対象外です。 • Minimum | Maximum

* これらのメトリクスは、Outposts、Local Zones、または Wavelength Zones では使用できません。

Amazon EBS スナップショットのメトリクス

AWS/EBS 名前空間には、Amazon EBS スナップショットの次のメトリクスが含まれます。

メトリクス	説明	単位	ディメンション	有意義な統計
SnapshotCopyBytesTransferred	AWS リージョンにコピーされたスナップショットデータの量。	バイト	sourceRegion	Sum

Nitro インスタンスのメトリクス

AWS/EC2 名前空間にはベアメタルインスタンスではない、Nitro ベースのインスタンスにアタッチされているボリュームに関する、追加の Amazon EBS メトリクスが含まれます。

メトリクス	説明	Unit	有意義な統計
InstanceEBSIOPSExceededCheck *	過去 1 分以内にアプリケーションがインスタンスの最大 EBS IOPS 制限を超える IOPS を駆動しようとしたかどうかをレポートします。このメトリクスは、0 (IOPS 制限を超えていない) または 1 (IOPS 制限を超えている) のいずれの値をとります。	なし	• 合計 • 平均 • 最小値 • 最大値
InstanceEBSThroughputExceededCheck *	過去 1 分以内にアプリケーションがインスタンスの最大 EBS スループット制限を超えるスループットを駆動しようとしたかどうかをレポートします。このメトリクスは、0 (スループット制限を超えていない) または 1 (スループット制限を超えている) のいずれかの値をとります。	なし	• 合計 • 平均 • 最小値 • 最大値

メトリクス	説明	Unit	有意義な統計
EBSRead0ps	指定された期間にインスタンスに接続されたすべての Amazon EBS ボリュームからの、完了した読み込みオペレーション。その期間の 1 秒あたりの読み込み I/O 操作回数 (読み込み IOPS) の平均を算出するにはその期間の操作回数の合計をその期間の秒数で割ります。基本 (5 分) モニタリングを使用している場合、この数を 300 で除算して読み込み IOPS を計算することができます。詳細 (1 分) モニタリングを使用している場合はこの数を 60 で除算します。CloudWatch メトリクスの計算関数 DIFF_TIME を使用して、1 秒あたりのオペレーション数を求めることもできます。例えば、CloudWatch で EBSRead0ps のグラフを m1 として作成した場合、メトリクスの計算関数 m1/(DIFF_TIME(m1)) はメトリクスをオペレーション/秒単位で返します。DIFF_TIME およびその他のメトリクス計算関数の詳細については、「Amazon CloudWatch ユーザーガイド」の「CloudWatch メトリクスでの数式の使用」を参照してください。	カウント	• 合計 • 平均 • 最小値 • 最大値

メトリクス	説明	Unit	有意義な統計
EBSWriteOps	指定された期間にインスタンスに接続されたすべての EBS ボリュームからの、完了した書き込み操作。その期間の 1 秒あたりの書き込み I/O 操作回数 (書き込み IOPS) の平均を算出するにはその期間の操作回数の合計をその期間の秒数で割ります。基本 (5 分) モニタリングを使用している場合、この数を 300 で除算して書き込み IOPS を計算することができます。詳細 (1 分) モニタリングを使用している場合はこの数を 60 で除算します。CloudWatch メトリクスの計算関数 DIFF_TIME を使用して、1 秒あたりのオペレーション数を求めることもできます。例えば、CloudWatch で EBSWriteOps のグラフを m1 として作成した場合、メトリクスの計算関数 m1/(DIFF_TIME(m1)) はメトリクスをオペレーション/秒単位で返します。DIFF_TIME およびその他のメトリクス計算関数の詳細については、「Amazon CloudWatch ユーザーガイド」の「CloudWatch メトリクスでの数式の使用」を参照してください。	カウント	• 合計 • 平均 • 最小値 • 最大値

メトリクス	説明	Unit	有意義な統計
EBSReadBytes	指定した期間内にインスタンスに接続されたすべての EBS ボリュームから読み取られたバイト数。報告された数は期間中に読み取られたバイト数です。基本 (5 分) モニタリングを使用している場合、この数を 300 で除算して読み込みバイト/秒を求めることができます。詳細 (1 分) モニタリングを使用している場合はこの数を 60 で除算します。CloudWatch メトリクスの計算関数 DIFF_TIME を使用して、1 秒あたりのバイト数を求めることもできます。例えば、CloudWatch で EBSReadBytes のグラフを m1 として作成した場合、メトリクスの数式 m1/(DIFF_TIME(m1)) はメトリクスをバイト/秒単位で返します。DIFF_TIME およびメトリクス計算関数の詳細については、「Amazon CloudWatch ユーザーガイド」の「メトリクス数式の使用」を参照してください。	バイト	• 合計 • 平均 • 最小値 • 最大値

メトリクス	説明	Unit	有意義な統計
EBSWriteBytes	指定した期間内にインスタンスに接続されたすべての EBS ボリュームに書き込まれたバイト数。報告された数は期間中に書き込まれたバイト数です。基本 (5 分) モニタリングを使用している場合、この数を 300 で除算して書き込みバイト/秒を求めることができます。詳細 (1 分) モニタリングを使用している場合はこの数を 60 で除算します。CloudWatch メトリクスの計算関数 DIFF_TIME を使用して、1 秒あたりのバイト数を求めることもできます。例えば、CloudWatch で EBSWriteBytes のグラフを m1 として作成した場合、メトリクスの数式 m1/(DIFF_TIME(m1)) はメトリクスをバイト/秒単位で返します。DIFF_TIME およびメトリクス計算関数の詳細については、「Amazon CloudWatch ユーザーガイド」の「メトリクス数式の使用」を参照してください。	バイト	• 合計 • 平均 • 最小値 • 最大値
EBSIOBalance%	バーストバケットの I/O 残りクレジットの割合に関する情報を提供します。このメトリクスは基本モニタリング専用です。このメトリクスは少なくとも 24 時間に 1 回、30 分間だけ最大パフォーマンスにバーストする一部の *.4xlarge インスタンスサイズ以下でのみ使用できます。詳細については、「EBS 最適化 (デフォルト)」を参照してください。 Sum 統計はこのメトリクスに該当しません。	割合 (%)	• 最小値 • 最大値

メトリクス	説明	Unit	有意義な統計
EBSByteBalance%	バーストバケットのスループット残りクレジットの割合に関する情報を提供します。このメトリクスは基本モニタリング専用です。このメトリクスは少なくとも 24 時間に 1 回、30 分間だけ最大パフォーマンスにバーストする一部の *.4xlarge インスタンスサイズ以下でのみ使用できます。詳細については、「EBS 最適化 (デフォルト)」を参照してください。 Sum 統計はこのメトリクスに該当しません。	割合 (%)	• 最小値 • 最大値

* これらのメトリクスは、Outposts、Local Zones、または Wavelength Zones では使用できません。

高速スナップショット復元のメトリクス

AWS/EBS の名前空間には、高速スナップショット復元に関する次のメトリクスが含まれています。

メトリクス	説明	単位	ディメンション	有意義な統計
FastSnapshotRestoreCreditsBucketSize	蓄積できるボリューム作成クレジットの最大数。このメトリクスは、アベイラビリティーゾーンごとにスナップショット単位で報告されます。	なし	SnapshotId | AvailabilityZone	• Average • Minimum | Maximum **Note** 最も有益な統計は Average です。Minimum 統計と Maximum 統計の結果は、Average と同じであり、代わ

メトリクス	説明	単位	ディメンション	有意義な統計
				りに使用できます。
FastSnapshotRestoreCreditsBalance	使用可能なボリューム作成クレジットの数。このメトリクスは、アベイラビリティーゾーンごとにスナップショット単位で報告されます。	なし	SnapshotId | AvailabilityZone	• Average • Minimum | Maximum Note 最も有益な統計は Average です。Minimum 統計と Maximum 統計の結果は、Average と同じであり、代わりに使用できます。

Amazon EC2 コンソールのグラフ

ボリュームを作成したら、Amazon EC2 コンソールでボリュームのモニタリンググラフを確認できます。コンソールの [Volumes] ページでボリュームを選択し、[Monitoring] を選択します。次の表は、表示されるグラフをまとめたものです。右側の欄は、各グラフを作成するために CloudWatch API の未加工データメトリクスがどのように使用されるかを示しています。すべてのグラフの期間は 5分です。

グラフ	未加工メトリクスの使用に関する説明
読み取りスループット (KiB/s)	Sum(VolumeReadBytes) / Period / 1024
書き込みスループット (KiB/s)	Sum(VolumeWriteBytes) / Period / 1024

グラフ	未加工メトリクスの使用に関する説明
読み取り操作 (OPS/秒)	`Sum(VolumeReadOps) / Period`
書き込み操作 (OPS/秒)	`Sum(VolumeWriteOps) / Period`
平均キュー長 (オペレーション)	`Avg(VolumeQueueLength)`
アイドル時間 (%)	`Sum(VolumeIdleTime) / Period × 100`

平均読み取りサイズ (KiB/OP)	`Avg(VolumeReadBytes) / 1024`
	
	Nitro ベースのインスタンスの場合、CloudWatch Metric Math で次の公式を使用して平均読み込みサイズを算出します。
	
	`(Sum(VolumeReadBytes) / Sum(Volum eReadOps)) / 1024`
	
	`VolumeReadBytes` および `VolumeReadOps` メトリクスはEBS CloudWatch コンソールで使用できます。

平均書き込みサイズ (KiB/OP)	`Avg(VolumeWriteBytes) / 1024`
	
	Nitro ベースのインスタンスの場合、CloudWatch Metric Math で次の公式を使用して平均書き込みサイズを算出します。
	
	`(Sum(VolumeWriteBytes) / Sum(Volum eWriteOps)) / 1024`
	
	`VolumeWriteBytes` および `VolumeWriteOps` メトリクスはEBS CloudWatch コンソールで使用できます。

グラフ	未加工メトリクスの使用に関する説明

平均読み取りレイテンシー (ms/op)	`Avg(VolumeTotalReadTime) × 1000`
	
	Nitro ベースのインスタンスの場合、CloudWatch Metric Math で次の公式を使用して平均レイテンシーを算出します。
	
	`(Sum(VolumeTotalReadTime) / Sum(VolumeReadOps)) × 1000`
	
	`VolumeTotalReadTime` および `VolumeReadOps` メトリクスはEBS CloudWatch コンソールで使用できます。

平均書き込みレイテンシー (ms/op)	`Avg(VolumeTotalWriteTime) × 1000`
	
	Nitro ベースのインスタンスの場合、CloudWatch Metric Math で次の公式を使用して平均書き込み待ち時間を算出します。
	
	`(Sum(VolumeTotalWriteTime) / Sum(VolumeWriteOps)) * 1000`
	
	`VolumeTotalWriteTime` および `VolumeWriteOps` メトリクスはEBS CloudWatch コンソールで使用できます。

平均レイテンシーグラフおよび平均サイズグラフでは、期間中に完了したオペレーション (読み込みまたは書き込みのうち、いずれかグラフに該当する方) の合計数に基づいて平均が計算されます。

Amazon EBS 用 Amazon EventBridge イベント

Amazon EBS は、ボリュームとスナップショットに対して実行されたアクションのイベントを Amazon EventBridge に送信します。EventBridge を使用することで、これらのイベントに対応するプログラム的なアクションをトリガーするルールを設定できます。例えば、スナップショットの高速復元が有効になったときに電子メールに通知を送信するルールを作成できます。

EventBridge でのイベントは、JSON オブジェクトとして表されます。イベント固有のフィールドは、JSON オブジェクトの「detail (詳細)」セクションに表示されます。「event」 フィールドにはイベント名が入ります。「result」 フィールドには、イベントをトリガーしたアクションの完了し

たステータスが入ります。詳細については、「Amazon EventBridge ユーザーガイド」の「Amazon EventBridge のイベントパターン」を参照してください。

詳細については、Amazon EventBridge ユーザーガイドの「What Is Amazon EventBridge?」を参照してください。

イベント

- EBS ボリュームイベント
- EBS ボリューム初期化イベント
- EBS ボリュームの変更イベント
- EBS スナップショットイベント
- EBS スナップショットのアーカイブイベント
- EBS 高速スナップショット復元イベント
- AWS Lambda を使用して EventBridge イベントを処理する

EBS ボリュームイベント

Amazon EBS は、次のボリュームイベントが発生したときに、EventBridge にイベントを送信します。

イベント

- ボリュームの作成 (createVolume)
- ボリュームの削除 (deleteVolume)
- ボリュームのアタッチまたは再アタッチ (attachVolume、reattachVolume)
- デタッチボリューム (detachVolume)

ボリュームの作成 (createVolume)

`createVolume` イベントは、ボリュームを作成するアクションが完了すると AWS アカウントに送信されます。ただし、保存、ログ記録、アーカイブはされません。このイベントの結果は、`available` または `failed` のいずれかです。以下の例に示すように、無効な AWS KMS key が指定されている場合、作成は失敗します。

イベントデータ

以下に示すのは、createVolume イベントが正常に完了したときに EBS から出力される JSON オブジェクトの例です。

```
{
   "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"
   }
}
```

以下に示すのは、createVolume が失敗したときに EBS から出力される JSON オブジェクトの例です。失敗の原因は無効な KMS キー です。

```
{
  "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",
  }
```

```
}
```

以下に示すのは、createVolume イベントが失敗した後で EBS から出力される JSON オブジェクトの例です。失敗の原因は、KMS キー の保留中のインポートです。

```
{
  "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",
  }
}
```

ボリュームの削除 (deleteVolume)

ボリュームを削除するアクションが完了すると、deleteVolumeイベントが AWS アカウントに送信されます。ただし、保存、ログ記録、アーカイブはされません。このイベントの結果は deleted です。削除が完了しない場合、イベントは送信されません。

イベントデータ

以下に示すのは、deleteVolume イベントが正常に完了したときに EBS から出力される JSON オブジェクトの例です。

```
{
   "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"
    }
}
```

ボリュームのアタッチまたは再アタッチ (attachVolume、reattachVolume)

インスタンスにボリュームがアタッチまたは再アタッチされたときに、attachVolume または reattachVolume イベントが AWS アカウントに送信されます。ただし、保存、ログ記録、アーカイブはされません。次の例に示すように、KMS キー を使用して EBS ボリュームを暗号化し、KMS キー が無効になった場合、インスタンスへのアタッチまたは再アタッチにその KMS キー が後で使用されると、EBS はイベントを出力します。

イベントデータ

以下に示すのは、attachVolume が失敗したときに EBS から出力される JSON オブジェクトの例です。失敗の原因は、KMS キー の保留中の削除です。

> **Note**
>
> AWS は、定期的なサーバーメンテナンス後にボリュームへの再アタッチを試みることがあります。

```
{
  "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": ""
  }
}
```

以下に示すのは、reattachVolume が失敗したときに EBS から出力される JSON オブジェクトの例です。失敗の原因は、KMS キー の保留中の削除です。

```
{
  "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": ""
  }
}
```

デタッチボリューム (detachVolume)

detachVolume イベントは、ボリュームが Amazon EC2 インスタンスからデタッチされると、AWS アカウントに送信されます。

イベントデータ

以下に、正常な detachVolume イベントの例を示します。


```
{
  "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"
    }
  }
}
```

EBS ボリューム初期化イベント

スナップショットから Amazon EBS ボリュームを作成し、デフォルトのボリューム初期化レートまたはボリューム初期化の Amazon EBS プロビジョンドレートを使用すると、ボリュームの初期化が完了してから 5 分以内にinitializeVolumeイベントが AWS アカウントに送信されます。詳細については、「ボリューム初期化に Amazon EBS プロビジョンドレートを使用する」を参照してください。

高速スナップショット復元を使用して作成されたボリュームの場合、イベントは送信されません。

> **Important**
>
> completionTime は、イベントが生成された日時を反映します。イベントは初期化の完了後 5 分以内に生成されるため、completionTime は初期化の完了後 5 分以内の日時となります。

以下に示しているのは、イベントの例です。

```
{
   "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-01234567890abcdef"
    ],
    "detail": {
       "event": " initializeVolume",
       "result": "succeeded",
       "completionTime": "yyyy-mm-ddThh:mm:ssZ",
       "request-id": "01234567-0123-0123-0123-0123456789ab"
    }
}
```

EBS ボリュームの変更イベント

Amazon EBS は、ボリュームが変更されると、EventBridge に modifyVolume イベントを送信します。ただし、保存、ログ記録、アーカイブはされません。

```
{
    "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"
    }
}
```

EBS スナップショットイベント

Amazon EBS は、次のボリュームイベントが発生したときに、EventBridge にイベントを送信します。

イベント

- スナップショットの作成 (createSnapshot)
- スナップショットの作成 (createSnapshots)
- スナップショットのコピー (copySnapshot)
- スナップショットを共有 (shareSnapshot)

スナップショットの作成 (createSnapshot)

スナップショットを作成するアクションが完了すると、createSnapshotイベントが AWS アカウントに送信されます。ただし、保存、ログ記録、アーカイブはされません。このイベントの結果は、succeeded または failed のいずれかです。

イベントデータ

以下に示すのは、createSnapshot イベントが正常に完了したときに EBS から出力される JSON オブジェクトの例です。detail セクションで、source フィールドにはソースボリュームの ARN が入ります。startTime フィールドと endTime フィールドは、スナップショット作成の開始時間と終了時間を示します。

```
{
  "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"  }
```

```
}
```

スナップショットの作成 (createSnapshots)

マルチボリュームスナップショットを作成するアクションが完了すると、createSnapshotsイベントが AWS アカウントに送信されます。このイベントの結果は、succeeded または failed のいずれかです。

イベントデータ

以下に示すのは、createSnapshots イベントが正常に完了したときに EBS から出力される JSON オブジェクトの例です。detail セクションで、source フィールドには、マルチボリュームスナップショットセットのソースボリュームの ARN が入ります。startTime フィールドと endTime フィールドは、スナップショット作成の開始時間と終了時間を示します。

```
{
  "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"
      }
    ]
  }
}
```

以下に示すのは、createSnapshots が失敗したときに EBS から出力される JSON オブジェクトの例です。失敗の原因は、マルチボリュームのスナップショットセットの 1 つ以上のスナップショットが完了しなかったことです。snapshot_id の値は、失敗したスナップショットの ARN です。startTime と endTime は、スナップショットを作成するアクションの開始時間と終了時間を表します。

```
{
  "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"
      }
```

```
    ]
  }
}
```

スナップショットのコピー (copySnapshot)

スナップショットをコピーするアクションが完了すると、copySnapshotイベントが AWS アカウントに送信されます。ただし、保存、ログ記録、アーカイブはされません。このイベントの結果は、succeeded または failed のいずれかです。

detail セクションでは、source はソーススナップショットの ARN であり、snapshot_id はスナップショットコピーの ARN です。startTime と endTime は、コピーオペレーションがいつ開始および終了したかを示します。incremental は、スナップショットコピーが増分スナップショット (true) であるか、完全なスナップショット (false) であるかを示します。transferType は、スナップショットコピーオペレーションが標準のコピーオペレーションであるか、時間ベースのコピーオペレーションであるかを示します。詳細については、「Amazon EBS スナップショットと EBS-backed AMI の時間ベースのコピー」を参照してください。

スナップショットをリージョン間でコピーしている場合、イベントは送信先のリージョンで発生します。

シナリオ 1: 標準のスナップショットコピーオペレーションが完了した

以下は、標準のスナップショットコピーオペレーションが正常に完了したときにアカウントに送信されるイベントの例です。transferType は standard に設定されています。

```
{
  "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",
    "transferType": "standard"
  }
}
```

シナリオ 2: 時間ベースのスナップショットコピーオペレーションが完了期間内に完了した

以下は、時間ベースのスナップショットコピーオペレーションが完了期間内に完了したときにアカウントに送信されるイベントの例です。transferType が time-based になっており、これは時間ベースのスナップショットコピーオペレーションであることを示します。completionDurationStartTime は完了期間がいつ開始されたかを示します。

```
{
  "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": "",
    "startTime": "yyyy-mm-ddThh:mm:ssZ",
    "endTime": "yyyy-mm-ddThh:mm:ssZ",
    "snapshot_id": "arn:aws:ec2::us-west-2:snapshot/snap-01234567",
    "source": "arn:aws:ec2::eu-west-1:snapshot/snap-76543210",
    "incremental": "true",
    "completionDurationStartTime":"2024-11-16T06:27:33.816Z",
    "transferType": "time-based"
  }
}
```

シナリオ 3: 時間ベースのスナップショットコピーオペレーションは完了したが、リクエストされた完了期間内に完了しなかった

時間ベースのスナップショットコピーオペレーションが完了したが、リクエストされた完了期間内に完了しなかった場合、CloudWatch は 2 つのイベントをアカウントに送信します。以下に、これらのイベントの例を示します。

- 1 つ目のイベントは、コピーオペレーションがまだ進行中であっても、完了期間が経過するとすぐにアカウントに送信されます。このイベントの場合、detail-type は EBS Copy Snapshot Missed Completion Duration であり、missedCompletionDurationCause はその理由を示します。

```
{
 "version":"0",
 "id":"fd90eb95-0938-e02c-cf55-b81363b8ac12",
 "detail-type":"EBS Copy Snapshot Missed Completion Duration",
 "source":"aws.ec2",
 "account":"123456789012",
 "time":"2024-11-19T18:17:08Z",
 "region":"us-east-1",
 "resources": ["arn:aws:ec2:us-east-1:123456789012:snapshot/snap-01234567890abcedf"],
 "detail":{
   "event":"copySnapshot",
   "missedCompletionDurationCause":"Snapshot copy was not able to meet the specified
 completion duration because your snapshot copy operation throughput quota was
 exceeded.",
   "snapshot_id":"arn:aws:ec2:us-
east-1:123456789012:snapshot/snap-01234567890abcedf",
   "source":"arn:aws:ec2:us-east-1:123456789012:snapshot/snap-00987654321fedcba",
   "startTime": "Sun Nov 24 22:32:55 UTC 2024",
   "transferType": "time-based"
   }
}
```

- 2 つ目のイベントは、スナップショットが完了した後にのみアカウントに送信されます。このイベントに含まれる missedCompletionDurationCause は理由を示します。

```
{
  "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": "",
    "startTime": "yyyy-mm-ddThh:mm:ssZ",
    "endTime": "yyyy-mm-ddThh:mm:ssZ",
    "snapshot_id": "arn:aws:ec2::us-west-2:snapshot/snap-01234567",
    "source": "arn:aws:ec2::eu-west-1:snapshot/snap-76543210",
    "incremental": "true",
    "completionDurationStartTime":"2024-11-16T06:27:33.816Z",
    "missedCompletionDurationCause":"Snapshot copy was not able to meet the specified
 completion duration because your snapshot copy operation throughput quota was
 exceeded.",
    "transferType": "time-based"
  }
}
```

シナリオ 4: スナップショットコピーオペレーションが失敗した

以下は、スナップショットコピーオペレーションが失敗したときにアカウントに送信されるイベントの例です。result が failed になっており、これはオペレーションが失敗したことを示します。

```
{
  "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"
  }
}
```

スナップショットを共有 (shareSnapshot)

別の AWS アカウントがスナップショットを共有すると、shareSnapshotイベントがアカウントに送信されます。ただし、保存、ログ記録、アーカイブはされません。結果は常に succeeded です。

イベントデータ

shareSnapshot イベントが完了したときに EBS から出力される JSON オブジェクトの例を以下に示します。detail セクションの の値は、スナップショットを共有したユーザーの AWS アカウント番号sourceです。 startTimeと は、スナップショット共有アクションが開始および終了した日時endTimeを表します。shareSnapshot イベントは、プライベートスナップショットが別のユーザーと共有された場合にのみ発生します。パブリックスナップショットを共有しても、イベントはトリガーされません。

```
{
  "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"
  }
}
```

EBS スナップショットのアーカイブイベント

Amazon EBS は、スナップショットアーカイブアクションに関連するイベントを発行します。詳細については、「CloudWatch Events を使用して Amazon EBS スナップショットアーカイブをモニタリング」を参照してください。

EBS 高速スナップショット復元イベント

スナップショットの高速スナップショット復元の状態が変わると、Amazon EBS はイベントを EventBridge に送信します。イベントは、ベストエフォートベースで出力されます。

以下はこのイベントのサンプルデータです。

```
{
   "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",
   }
}
```

`state` の想定される値は、`enabling`、`optimizing`、`enabled`、`disabling`、および `disabled` です。

`message` の有効な値は次のとおりです。

`Client.InvalidSnapshot.InvalidState - The requested snapshot transitioned to an invalid state (Error)`

高速スナップショット復元を有効にするリクエストが失敗し、状態は disabling または disabled に移行しました。このスナップショットに対しては、高速スナップショット復元を有効にすることができません。

`Client.UserInitiated`

状態は、正常に enabling または disabling に移行しました。

`Client.UserInitiated - Lifecycle state transition`

状態は、正常に optimizing、enabled、または disabled に移行しました。

`Server.InsufficientCapacity - There was insufficient capacity available to satisfy the request`

高速スナップショット復元を有効にするリクエストが容量不足のために失敗し、状態は disabling または disabled に移行しました。しばらく待ってから、もう一度試してください。

`Server.InternalError - An internal error caused the operation to fail`

高速スナップショット復元を有効にするリクエストが内部エラーのために失敗し、状態は disabling または disabled に移行しました。しばらく待ってから、もう一度試してください。

`Client.InvalidSnapshot.InvalidState - The requested snapshot was deleted or access permissions were revoked`

スナップショットが削除されたか、スナップショット所有者によって共有解除されたため、スナップショットに対する高速スナップショット復元の状態が disabling または disabled に移行しました。削除されたか共有しなくなったスナップショットに対して、高速スナップショット復元を有効にすることはできません。

AWS Lambda を使用して EventBridge イベントを処理する

Amazon EBS と Amazon EventBridge を使用して、データのバックアップのワークフローを自動化できます。そのためには、IAM ポリシー、イベントを処理する AWS Lambda 関数、受信イベントを照合して Lambda 関数にルーティングする EventBridge ルールを作成する必要があります。

次の手順では、createSnapshot イベントを使用して完成したスナップショットを災害対策の目的で自動的に別のリージョンにコピーします。

完了したスナップショットを別のリージョンにコピーするには

1. 次の例に示すような IAM ポリシーを作成し、CopySnapshot アクションを使用して EventBridge ログに書き込むためのアクセス許可を提供します。このポリシーを EventBridge イベントを処理するユーザーに割り当てます。

   JSON

```
{
  "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. EventBridge コンソールから利用できる関数を Lambda で定義します。以下の Lambda 関数は、Node.js で記述したサンプルであり、該当する createSnapshot イベント (スナップショットの完成を示す) が Amazon EBS から出力されたときに EventBridge から呼び出されます。この関数は、呼び出されると、スナップショットを us-east-2 から us-east-1 にコピーします。

```
// 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);
```

```
        }
    });
};
```

Lambda 関数が EventBridge コンソールから使用可能であることを確認するには、EventBridge イベントが発生するリージョンに関数を作成します。詳細については、AWS Lambda デベロッパーガイドを参照してください。

3. Amazon EventBridge コンソール のhttps://console.aws.amazon.com/events/ を開いてください。
4. ナビゲーションペインで、[Rules (ルール)] を選択し、[Create rule (ルールの作成)] を選択してください。
5. [Step 1: Define rule detail] (ステップ 1: ルールの詳細を定義する) で、次の操作を行います。
   a. [Name] (名前) と [Description] (説明) の値を入力します。
   b. [Event bus] (イベントバス) は [default] (デフォルト) のままにします。
   c. 選択したイベントバスでルールを有効にする がオンになっていることを確認します。
   d. [Event type] (イベントタイプ) で、[Rule with an event pattern] (イベントパターンを使用するルール) を選択します。
   e. [次へ] を選択します。
6. [Step 2: Build event pattern] (ステップ 2: イベントパターンを作成する) で、次の操作を行います。
   a. [イベントソース] で、[AWS イベントまたは EventBridge パートナーイベント] を選択します。
   b. 「イベントパターン」セクションの「イベントソース」では、AWS サービスが選択されていることを確認します。AWS 「サービス」では、「EC2」を選択します。
   c. [Event type] (イベントタイプ) で、[EBS Snapshot Notification] (EBS スナップショット通知) を選択し、[Specific event(s)] (特定のイベント) を選択してから、[createSnapshot] を選択します。
   d. [Specific result(s)] (特定の結果) を選択してから、[succeeded] (成功) を選択します。
   e. [次へ] を選択します。
7. [Step 3: Select targets] (ステップ 3: ターゲットを選択する) で、次を実行します。
   a. [ターゲットタイプ] で、[AWS サービス] を選択します。

b. [Select target] (ターゲットを選択) で [Lambda function] (Lambda 関数) を選択し、前に作成した関数を [Function] (関数) で選択します。

c. [Next] (次へ) を選択します。

8. [Step 4: Configure tags] (ステップ 4: タグを設定する) で、必要に応じてルールのタグを指定し、[Next] (次へ) を選択します。

9. [Step 5: Review and create] (ステップ 5: 確認および作成する) でルールを確認し、[Create rule] (ルールを作成) を選択します。

作成したルールが、[Rules] タブに表示されます。上の例で、設定したイベントは次回にスナップショットをコピーすると EBS から出力されます。

Amazon EBS の詳細なパフォーマンス統計

Amazon EBS NVMe ブロックデバイスは、Nitro ベースの Amazon EC2 インスタンスにアタッチされた Amazon EBS ボリュームに関するリアルタイムかつ高解像度の I/O パフォーマンス統計を提供します。これらの統計は集計カウンターとして表示され、インスタンスへのボリュームのアタッチメントの存続期間にわたって保持されます。この統計は、オペレーションの累積数、送受信されたバイト数、読み取り/書き込み I/O オペレーションに費やされた時間に関する詳細を提供します。さらに、統計には読み取りおよび書き込み I/O オペレーションのヒストグラムが含まれます。また、アプリケーションが EBS ボリュームまたはアタッチされたインスタンスのプロビジョニングされた IOPS またはスループット制限を超えた合計時間もレポートします。

これらの統計は最大 1 秒間隔で詳細に収集できます。リクエストが 1 秒間隔よりも頻繁に行われると、NVMe ドライバーが、後で処理されるリクエストを他の管理コマンドとともにキューに入れることがあります。

考慮事項

- 統計は、すべての Amazon EBS ボリュームタイプでサポートされています。
- 統計は、AWS Nitro System 上に構築されたインスタンスにアタッチされたボリュームでのみサポートされます。
- 統計は、マルチアタッチが有効なボリュームで使用できます。マルチアタッチが有効なボリュームの統計を表示する場合、統計はそのインスタンスのアタッチメントに固有であり、そのインスタンスの使用状況のみが反映されます。
- この統計は追加料金なしでご利用いただけます。

統計

Amazon EBS NVMe ブロックデバイスは次の統計を提供します。

統計名	フルネーム	型	説明
total_rea d_ops	読み取りオペレーションの合計回数	Counter	完了した読み取りオペレーションの合計回数。
total_wri te_ops	書き込みオペレーションの合計数	Counter	完了した書き込みオペレーションの合計回数。
total_rea d_bytes	読み取りバイトの合計数	Counter	転送された読み取りバイトの合計数。
total_wri te_bytes	書き込みバイトの合計数	Counter	転送された書き込みバイトの合計数。
total_rea d_time	読み取り時間の合計数	Counter	完了したすべての読み取りオペレーションで費やされた合計時間 (マイクロ秒単位)。
total_wri te_time	書き込み時間の合計数	Counter	完了したすべての書き込みオペレーションで費やされた合計時間 (マイクロ秒単位)。
ebs_volur e_perforr ance_exce eded_iops	需要がボリュームのプロビジョンド IOPS を超えた合計時間	Counter	IOPS 需要がボリュームのプロビジョンド IOPS パフォーマンスを超えた合計時間 (マイクロ秒単位)。
ebs_volur e_perforr ance_exce eded_tp	需要がボリュームのプロビジョンドスループットを越えた合計時間	Counter	スループット需要がボリュームのプロビジョンドスループットパフォーマンスを超えた合計時間 (マイクロ秒単位)。
ec2_insta nce_ebs_p erformanc	需要が EC2 インスタンスの IOPS パフォーマンスを超えた合計時間	Counter	EBS ボリュームがアタッチされた Amazon EC2 インスタンスの最大 IOPS パフォーマンスを超えた合計時間 (マイクロ秒単位)。

統計名	フルネーム	型	説明
e_exceede d_iops			
ec2_insta nce_ebs_p erformanc e_exceede d_tp	需要が EC2 インスタンスのスループットパフォーマンスを超えた合計時間	Counter	EBS ボリュームがアタッチされた Amazon EC2 インスタンスの最大スループットパフォーマンスを超えた合計時間 (マイクロ秒単位)。
volume_qu eue_lengt h	ボリュームキューの長さ	ポイントインタイム	完了を待機している読み取りおよび書き込みオペレーションの数。
read_io_l atency_hi stogram	I/O ヒストグラムの読み取り	ヒストグラム*	各レイテンシービン内で完了した読み取りオペレーションの数 (単位: マイクロ秒)。
write_io_ latency_h istogram	I/O ヒストグラムの書き込み	ヒストグラム*	各レイテンシービン内で完了した書き込みオペレーションの数 (単位: マイクロ秒)。

> **Note**
>
> * ヒストグラム統計は正常に完了した I/O オペレーションのみを表します。停止している、あるいは正常でない I/O オペレーションは含まれませんが、ポイントインタイム統計として volume_queue_length 統計に表示されます。

統計へのアクセス

統計には、Amazon EBS ボリュームがアタッチされているインスタンスから、直接アクセスする必要があります。統計には、次のいずれかの方法を使用してアクセスできます。

Linux インスタンス

Amazon CloudWatch

インスタンスから統計を収集し Amazon CloudWatch のカスタムメトリクスとして利用できるように、Amazon CloudWatch エージェントを設定できます。その後、CloudWatch のメトリクスを使用して、I/O パターンの分析、パフォーマンスの傾向の追跡、カスタムダッシュボードの作成、パフォーマンスのしきい値に基づく自動アラームの設定を行うことができます。

CloudWatch エージェントの設定の詳細については、次のトピックを参照してください。

- CloudWatch エージェント設定ファイルを手動で作成または編集する
- Amazon EBS NVMe ドライバーメトリクスを収集する

Amazon CloudWatch Observability EKS アドオンバージョン `4.1.0` 以降では、Amazon EBS CSI ドライバーメトリクスが有効になっていると自動的に統計が収集されます。詳細については、「Amazon EBS NVMe ドライバーメトリクス」を参照してください。

ebsnvme script

`ebsnvme` スクリプトは amazon-ec2-utils Github リポジトリにあります。

統計にアクセスするには

1. ボリュームがアタッチされているインスタンスに接続します。
2. `amazon-ec2-utils` GitHub リポジトリから `ebsnvme` スクリプトをダウンロードします。

```
wget https://raw.githubusercontent.com/amazonlinux/amazon-ec2-utils/refs/heads/main/ebsnvme
```

3. スクリプトのアクセス許可を変更して、実行可能にします。

```
sudo chmod +x ./ebsnvme
```

4. `ebsnvme` スクリプトを実行し、ボリュームのデバイス名を指定します。

```
sudo ./ebsnvme stats /dev/nvme0n1
```

nvme-cli tool

統計にアクセスするには

1. ボリュームがアタッチされているインスタンスに接続します。
2. 2024 年 11 月 12 日以降にリリースされた Amazon Linux AMI には、`nvme-cli` ツールの最新バージョンが含まれています。以前の Amazon Linux AMI を使用している場合は `nvme-cli` ツールを更新します。

```
sudo yum install nvme-cli
```

3. 次のコマンドを実行し、ボリュームのデバイス名を指定します。

```
nvme amzn stats /dev/nvme0n1
```

Prometheus

統計は、オープンソースのモニタリングアプリケーションである Prometheus、および Amazon Managed Service for Prometheus でモニタリングできます。これにより、コンテナと Kubernetes 環境全体で Amazon EBS ボリュームの大規模なモニタリングを簡単に行えるようになります。Amazon EBS CSI ドライバーバージョン v1.37.0 以降では、詳細なパフォーマンス統計は Prometheus 互換 `/metrics` エンドポイントとして公開され、Prometheus にエクスポートできます。

詳細については、「Amazon Managed Service for Prometheus User Guide」の「Ingest metrics to your Amazon Managed Service for Prometheus workspace」を参照してください。

Windows インスタンス

nvme_amzn.exe tool

統計にアクセスするには

1. ボリュームがアタッチされているインスタンスに接続します。
2. AWSNVMe ドライバーのバージョン `1.7.0` 以降を使用していることを確認します。AWSNVMe ドライバーの更新の詳細については、「AWS NVMe ドライバー」を参照してください。

3. EBS ボリュームのディスク番号を取得します。詳細については、「Map Amazon EBS volumes to NVMe device names」を参照してください。
4. 次のコマンドを管理者として実行し、ボリュームのディスク番号を指定します。

```
.\nvme_amzn.exe stats disk_number
```

Amazon EBS 用 Amazon GuardDuty

Amazon GuardDuty は、 AWS 環境内のアカウント、コンテナ、ワークロード、およびデータを保護するのに役立つ脅威検出サービスです。GuardDuty は、機械学習 (ML) モデル、異常および脅威検出機能を使用して、さまざまなログソースとランタイムアクティビティを継続的に監視し、環境内の潜在的なセキュリティリスクと悪意のあるアクティビティを特定して優先順位を付けます。

GuardDuty 内の Malware Protection 機能は、Amazon EC2 インスタンスとコンテナワークロードに関連付けられた Amazon EBS ボリュームをスキャンして、潜在的な脅威を検出します。GuardDuty は、2 つの方法で検出を行います。

- マルウェア保護を有効にする — GuardDuty が Amazon EC2 インスタンスまたはコンテナワークロードにマルウェアが存在する可能性を示す検出結果を生成すると、侵害された可能性のあるリソースに対してマルウェアスキャンが自動的に開始されます。
- Malware Protection を有効にせずにオンデマンドのマルウェアスキャンを使用する — Amazon EC2 インスタンスの Amazon リソースネーム (ARN) を指定して、オンデマンドスキャンを開始します。

詳細については、「Amazon GuardDuty ユーザーガイド」を参照してください。

請求および使用状況レポートに表示される Amazon EBS のコードを理解する

Amazon EBS を使用すると、 AWS 請求レポートと使用状況レポートに関連コードが含まれます。これらのコードを確認することは、Amazon EBS のコストと使用パターンを理解するのに役立ちます。コストを最適化するには、費用の追跡と管理が不可欠です。

次の表は、請求および使用状況レポートに表示される Amazon EBS のコードを示しています。請求および使用状況レポートで使用されるリージョンコードのリストについては、「AWS Region billing codes」を参照してください。

請求コードの対象

- スナップショット
- ボリュームストレージ
- プロビジョンドパフォーマンス
- EBS direct API
- 高速スナップショット復元 (FSR)

関連リソース

- Amazon EBS の料金
- the section called “料金”

スナップショット

コード	説明	単位	詳細度
region-EBS:SnapshotUsage	標準階層のスナップショット (増分コピー)。	GB-月	毎時
region-EBS:SnapshotArchiveStorage	アーカイブ階層のスナップショット (フルコピー)。	GB-月	毎時
region-EBS:SnapshotArchiveEarlyDelete	最小保持期間が終了する前にアーカイブされたスナップショットを削除します。	GB	削除あたり
region-EBS:SnapshotArchi	アーカイブされたスナップショットを取得します。	GB	GB あたり

コード	説明	単位	詳細度
veRetrieval			

ボリュームストレージ

コード	説明	単位	詳細度
region-EBS:VolumeUsage.gp3	汎用 SSD (gp3) ボリュームのストレージ。	GB-月	毎時
region-EBS:VolumeUsage.gp2	汎用 SSD (g2) ボリュームのストレージ。	GB-月	毎時
region-EBS:VolumeUsage.io2	プロビジョンド IOPS SSD (io2) ボリュームのストレージ。	GB-月	毎時
region-EBS:VolumeUsage.piops	プロビジョンド IOPS SSD (io1) ボリュームのストレージ	GB-月	毎時
region-EBS:VolumeUsage.st1	スループット最適化 HDD ボリュームのストレージ。	GB-月	毎時
region-EBS:Volu	コールド HDD ボリュームのストレージ。	GB-月	毎時

コード	説明	単位	詳細度
meUsage.sc1			
region-EBS:VolumeUsage	マグネティックボリュームのストレージ。	GB-月	毎時

プロビジョンドパフォーマンス

コード	説明	単位	詳細度
region-EBS:VolumeP-Throughput.gp3	125 MB/秒を超える gp3 ボリュームのプロビジョンドスループット。	MB/秒 - 月	毎時
region-EBS:VolumeP-IOPS.gp3	3,000 を超える gp3 ボリュームのプロビジョンド IOPS。	IOPS - 月	毎時
region-EBS:VolumeP-IOPS.io2	32,000 までの io2 ボリュームのプロビジョンド IOPS。	IOPS - 月	毎時
region-EBS:VolumeP-IOPS.io2.tier2	32,001 から 64,000 の io2 ボリュームのプロビジョンド IOPS。	IOPS - 月	毎時
region-EBS:Volu	64,000 を超える io2 ボリュームのプロビジョンド IOPS。	IOPS - 月	毎時

コード	説明	単位	詳細度
meP-IOPS.io2.tier3			
region-EBS:VolumeP-IOPS.piops	io1 ボリュームのプロビジョンド IOPS。	IOPS - 月	毎時
region-EBS:VolumeIOUsage	レガシー IOPS。	IOPS - 月	毎時

EBS direct API

コード	説明	単位	詳細度
region-EBS:directAPI.snapshot.List	ListChangedBlocks および ListSnapshotBlocks API アクションを呼び出します。	1,000 件のリクエストあたり	リクエストあたり
region-EBS:directAPI.snapshot.Get	GetSnapshotBlock API アクションによって返されたブロック。	返された 1,000 ブロックあたり	リクエストあたり
region-EBS:directAPI.snapshot.Put	PutSnapshotBlock API アクションによって書き込まれたブロック。	書き込まれた 1,000 ブロックあたり	リクエストあたり

高速スナップショット復元 (FSR)

コード	説明	単位	詳細度
region-EBS:FastSnapshotRestore	高速スナップショット復元 (AZ あたり) が有効になっているスナップショットのデータサービスの単位時間 (DSU 時)。	DSU 時	1 分あたり (最低 1 時間)

EBS ボリュームのインベントリを作成する

Amazon EBS は、EC2 インスタンスで使用できるスケーラブルで高性能なブロックストレージリソースを提供します。EC2 インスタンスに EBS ボリュームをアタッチすることも、EC2 インスタンスから EBS ボリュームをデタッチすることもできます。

EBS ストレージの主な特性は次のとおりです。システム管理者は、EBS リソースのこれらの特性に関する情報を取得し、それを使用して、オンプレミスで実行するサーバーまたは別のクラウドプロバイダーのサーバーと機能的に同等のストレージを設定できます。システム管理者は、EBS リソースに保存されているデータを取得し、そのデータをオンプレミスまたは別のクラウドプロバイダーのストレージに保存することもできます。

特性	説明
ボリューム	EBS ボリュームは、インスタンスへのアタッチ、フォーマット、マウントが可能な、耐久性に優れたブロックレベルストレージデバイスです。
I/O パフォーマンス	Amazon EBS には、パフォーマンス特性と価格の点で異なる複数のボリュームタイプが用意されています。
EBS データボリュームの内容	ボリュームのデータに直接アクセスする唯一の方法は、インスタンスからアクセスすることです。

特性	説明
スナップショット	EBS スナップショットは、EBS ボリューム上のデータのポイントインタイムコピーです。スナップショットは Amazon S3 のバケット (直接アクセスできない) に保存されます。
EBS スナップショットの内容	スナップショットからデータを読み取ることも、スナップショットを使用してボリュームを作成することもできます。

ボリューム

EBS ボリュームを作成するときは、ボリュームのアベイラビリティーゾーンを選択します。EBS ボリュームは、同じアベイラビリティーゾーン内の EC2 インスタンスにアタッチする必要があります。EC2 インスタンスを起動するときにボリュームを作成するか、最初にボリュームを作成してからインスタンスにアタッチできます。ボリュームがインスタンスにアタッチされている場合、そのステータスは `in-use` になります。ボリュームがインスタンスにアタッチされていない場合、そのステータスは `available` になります。

これらが重要な理由

EBS ルートボリュームには、EC2 インスタンスのオペレーティングシステムが含まれています。EBS データボリュームには、ビジネスクリティカルなデータを含めることができます。すべてのリージョンの EBS ボリュームを一覧表示し、EC2 インスタンスにアタッチされているかどうかを確認できます。

すべてのリージョンの EBS ボリュームの概要を表示するには

Amazon EC2 Global View を使用すると、すべてのリージョンの EBS ボリュームを一覧表示できます。

1. Amazon EC2 Global View コンソール (https://console.aws.amazon.com/ec2globalview/home) を開きます。
2. [リージョンエクスプローラー] タブの [概要] で、[ボリューム] のリソース数を確認します。これには、ボリューム数とリージョン数が含まれており、下線付きのテキストを選択すると、ボリューム数がリージョン間でどのように分散されるかが表示されます。

3. [グローバル検索] タブで、クライアントフィルター [リソースタイプ = ボリューム] を選択します。リージョンまたはタグを指定することで、結果をさらにフィルタリングできます。

使用中の EBS ボリュームを記述するには

describe-volumes コマンドを使用します。このコマンドは、EBS ボリュームが存在する各リージョンで実行する必要があります。--filters パラメータは、結果を in-use のボリュームにスコープします。--query パラメータは、出力内の指定されたフィールドのみを表示します。必要に応じて追加のフィールドを含めることができます。

```
aws ec2 describe-volumes \
    --filters Name=status,Values=in-use \
    --query "Volumes[].[VolumeId,Size,Attachments[0].InstanceId]" \
    --output table
```

以下は出力の例です。各列は、ボリューム ID、ボリュームサイズ、アタッチされているインスタンスの ID、デバイス名を示しています。

```
----------------------------------------------------------------------
|                          DescribeVolumes                           |
+-------------------------+-----+-----------------------+------------+
|  vol-0992ee65bec96de19  |  8  |  i-00a7d9ec76a46a49f  |  /dev/xvda |
|  vol-04d631d22047db1cb  |  30 |  i-0b1bf24fd4f297ab9  |  /dev/sda1 |
   ...
   ...
|  vol-0b811be5cbd1425cc | 100 |  i-0a8d998154b320257  |  /dev/xvde |
+-------------------------+-----+-----------------------+------------+
```

使用されていない EBS ボリュームを記述するには

前のコマンドのフィルターを次のように変更して、ステータスが available のボリュームを検索します。

```
--filters Name=status,Values=available
```

ボリュームが使用されていない場合は、必要なデータがあるかどうかを判断する必要があります。例えば、テスト目的で作成されたボリュームを削除したり、問題をトラブルシューティングしたりできます。ボリュームをインスタンスにアタッチしたブロックデバイスが、インスタンスの終了後も永続するように設定されている場合も、ボリュームが使用可能になることがあります。このような場合は、ボリューム上のデータがまだ必要かどうかを確認します。

I/O パフォーマンス

EBS ボリュームに指定するボリュームタイプによって、そのパフォーマンスが決まります。ボリュームタイプに応じて、ボリュームのサイズ、IOPS、スループットの組み合わせを指定できます。各ボリュームタイプの最大プロビジョンド IOPS とスループットの詳細については、「Amazon EBS volume types」を参照してください。

トラフィック管理が重要である理由

EBS ボリュームのパフォーマンスを決定したら、機能的に同等のストレージに必要な最小仕様を決定できます。

ボリュームのパフォーマンスに関する情報を取得するには

describe-volumes コマンドを使用して、使用中のボリュームのパフォーマンスを記述します。--query パラメータは、出力内の指定されたフィールドのみを表示します。必要に応じて追加のフィールドを含めることができます。EBS に最適化されたパフォーマンスの詳細については、「Amazon EC2 Instance Types Guide」の「Amazon EC2 instance type specifications」を参照してください。

```
aws ec2 describe-volumes \
    --filters Name=status,Values=in-use \
    --query "Volumes[].[VolumeId,VolumeType,Iops,Throughput]" \
    --output table
```

以下は出力の例です。各列は、ボリューム ID、ボリュームタイプ、IOPS、スループット (MiB/秒) を示しています。

```
-------------------------------------------------
|                DescribeVolumes                |
+------------------------+------+-------+-------+
|  vol-0992ee65bec96de19 |  gp3 |  3000 |  125  |
|  vol-04d631d22047db1cb |  gp2 |  100  |  None |
   ...
   ...
|  vol-0b811be5cbd1425cc |  gp3 |  3000 |  125  |
+------------------------+------+-------+-------+
```

EBS データボリュームの内容

Amazon EBS は、EBS ボリュームに保存されているデータにアクセスできません。データボリュームをバックアップする責任はお客様にあります。例えば、EBS スナップショットと呼ばれるポイントインタイムスナップショットを作成できます。

EC2 インスタンスに接続し、EBS ボリュームからご自身のコンピュータにファイルを転送することもできます。これを行うのに役立つツールが多数用意されています。以下は例です。

トラフィック管理が重要である理由

EBS ボリュームにビジネスクリティカルなデータがある場合は、機能的に同等のストレージにコピーできます。

Linux インスタンスにアタッチされたボリュームからファイルを転送するには

Linux インスタンスに接続して SCP を使用します。詳細については、「SCP を使用して Linux インスタンスにファイルを転送する」を参照してください。

Windows インスタンスにアタッチされたボリュームからファイルを転送するには

Windows インスタンスに接続して RDP を使用します。詳細については、「RDP を使用して Windows インスタンスにファイルを転送する」を参照してください。

スナップショット

EBS スナップショットを作成して、EBS データボリュームをバックアップできます。また、Amazon EBS、Amazon Data Lifecycle Manager、または AWS Backup を使用して、EBS スナップショットを作成できます。

トラフィック管理が重要である理由

EBS ボリュームにビジネスクリティカルなデータがある場合、通常は EBS スナップショットの形式でバックアップを作成します。ボリュームの最新のスナップショットを検索し、現在のデータがボリュームに反映されていることを確認できます。

特定のボリュームのスナップショットを記述するには

describe-snapshots コマンドを使用します。`--filters` パラメータは、指定されたボリュームのスナップショットに結果をスコープします。`--query` パラメータは、出力内の指定されたフィールドのみを表示します。必要に応じて追加のフィールドを含めることができます。

```
aws ec2 describe-snapshots \
    --filters Name=volume-id,Values=vol-00622ef8c2ac8b762 \
    --query Snapshots[*].[SnapshotId,CompletionTime,Description] --output table
```

以下は出力の例です。各列は、スナップショット ID、完了日、ボリュームが暗号化されているかどうかを示しています。

```
-------------------------------------------------------------------------
|                            DescribeSnapshots                          |
+-------------------------+--------------------------------------+--------+
|  snap-0ad439c50efabb47c |  2024-07-12T03:36:27.952000+00:00  |  False |
+-------------------------+--------------------------------------+--------+
```

EBS スナップショットの内容

EBS スナップショットの内容を別の形式でダウンロードまたはエクスポートすることはできません。ただし、EBS direct API を使用するか、スナップショットから EBS ボリュームを作成してファイルを転送することで、スナップショット内のデータにアクセスできます。

トラフィック管理が重要である理由

EBS ボリュームにビジネスクリティカルなデータがある場合、通常は EBS スナップショットの形式でバックアップを作成します。このデータを取得し、機能的に同等のストレージにコピーできます。

オプション 1: スナップショットからデータを読み取るには

EBS direct API を使用します。

オプション 2: スナップショットからボリュームを作成し、データを転送するには

1. create-volume コマンドを使用して、スナップショットからボリュームを作成します。

   ```
   aws ec2 create-volume \
      --volume-type gp3 \
      --snapshot-id snap-0ad439c50efabb47c \
      --iops 6000 \
      --availability-zone us-east-2b
   ```

2. attach-volume コマンドを使用して、インスタンスにボリュームをアタッチします。

   ```
   aws ec2 attach-volume \
   ```

```
    --device /dev/sdf \
    --instance-id i-0b1bf24fd4f297ab9 \
    --volume-id vol-1234567890abcdef0
```

3. インスタンスに接続し、ボリュームをフォーマットおよびマウントします。
4. ご自身のコンピュータにボリューム上のファイルを転送します。

Amazon EBS のクォータ

AWS アカウント には、それぞれ の制限と呼ばれるデフォルトのクォータがあります AWS のサービス。特に明記されていない限り、クォータは地域固有です。一部のクォータについては引き上げをリクエストできますが、その他のクォータについては引き上げることはできません。

Amazon EBS のクォータを表示するには、「Service Quotas コンソール」を開きます。ナビゲーションペインで、[AWS サービス] を選択し、[Amazon Elastic Block Store (Amazon EBS)] を選択します。クォータの引き上げをリクエストするには、Service Quotas ユーザーガイド の「クォータ引き上げリクエスト」を参照してください。

AWS アカウント には、Amazon EBS に関連する次のクォータがあります。

名前	デフォルト	引き上げ可能	説明
ボリュームあたりのアーカイブされたスナップショット	サポートされている各リージョン: 25	可能	ボリュームあたりのアーカイブされたスナップショットの最大数。
アカウントあたりの CompleteSnapshot	サポートされている各リージョン : 10/秒	いいえ	アカウントあたりの許容される CompleteSnapshot リクエストの最大数。
送信先リージョンあたりの同時スナップショットコピー	サポートされている各リージョン: 20	いいえ	1 つの送信先リージョンへの同時スナップショットコピーの最大数。
Cold HDD (sc1) ボリュームあたりの同時スナップショット	サポートされている各リージョン: 1	[いいえ]	このリージョンの Cold HDD (sc1) ボリュームあたりの同時スナップショットの最大数。

名前	デフォルト	引き上げ可能	説明
汎用 SSD (gp2) ボリュームあたりの同時スナップショット	サポートされている各リージョン：5	いいえ	このリージョンの汎用 SSD (gp2) ボリュームあたりの同時スナップショットの最大数。
汎用 SSD (gp3) ボリュームあたりの同時スナップショット	サポートされている各リージョン：5	いいえ	このリージョンの汎用 SSD (gp3) ボリュームあたりの同時スナップショットの最大数。
マグネティック (スタンダード) ボリュームあたりの同時スナップショット	サポートされている各リージョン：5	いいえ	このリージョンのマグネティック (スタンダード) ボリュームあたりの同時スナップショットの最大数。
プロビジョンド IOPS SSD (io1) ボリュームあたりの同時スナップショット	サポートされている各リージョン：5	いいえ	このリージョンのプロビジョンド IOPS SSD (io1) ボリュームあたりの同時スナップショットの最大数。
プロビジョンド IOPS SSD (io2) ボリュームあたりの同時スナップショット	サポートされている各リージョン：5	いいえ	このリージョンのプロビジョンド IOPS SSD (io2) ボリュームあたりの同時スナップショットの最大数。

名前	デフォルト	引き上げ可能	説明
スループット最適化 HDD (st1) ボリュームあたりの同時スナップショット	サポートされている各リージョン: 1	[いいえ]	このリージョンのスループット最適化 HDD (st1) ボリュームあたりの同時スナップショットの最大数。
高速スナップショット復元	サポートされている各リージョン：5	あり	このリージョンで高速スナップショット復元を有効化できるスナップショットの最大数。
アカウントあたりの GetSnapshotBlock リクエスト	us-east-1: 5,000/秒 us-east-2: 5,000/秒 us-west-2: 5,000/秒 ap-southeast-1: 5,000/秒 eu-west-1: 5,000/秒 他のサポートされている各リージョン: 1,000/秒	あり	アカウントあたりの許容される GetSnapshotBlock リクエストの最大数。

名前	デフォルト	引き上げ可能	説明
スナップショットごとの GetSnapshotBlock リクエスト数	サポートされている各リージョン: 1,000/秒	いいえ	スナップショットごとの許容される GetSnapshotBlock リクエストの最大数。
プロビジョンド IOPS SSD (io1) ボリュームの IOPS	サポートされている各リージョン: 300,000	あり	このリージョンのプロビジョンド IOPS SSD (io1) ボリューム間でプロビジョニングできる IOPS の最大集計数。
プロビジョンド IOPS SSD (io2) ボリュームの IOPS	サポートされている各リージョン: 100,000	あり	このリージョンのプロビジョンド IOPS SSD (io2) ボリューム間でプロビジョニングできる IOPS の最大集計数。
プロビジョンド IOPS SSD (io1) ボリュームの IOPS 変更	サポートされている各リージョン: 500,000	あり	このリージョンのプロビジョンド IOPS SSD (io1) ボリューム全体のボリューム変更でリクエストできる IOPS の最大集計数。
プロビジョンド IOPS SSD (io2) ボリュームの IOPS 変更	サポートされている各リージョン: 100,000	あり	このリージョンのプロビジョンド IOPS SSD (io2) ボリューム全体のボリューム変更リクエストでリクエストできる IOPS の最大更新数。

名前	デフォルト	引き上げ可能	説明
アカウントごとの進行中のスナップショットアーカイブ数	サポートされている各リージョン: 25	可能	アカウントごとの進行中のスナップショットアーカイブの最大数。
アカウントごとの進行中のスナップショットのアーカイブからの復元数	サポートされている各リージョン：5	あり	アカウントごとの進行中のスナップショットのアーカイブからの復元の最大数。
アカウントあたりの ListChangedBlocks リクエスト数	サポートされている各リージョン: 50/秒	いいえ	アカウントあたりの許容される ListChangedBlocks リクエストの最大数。
アカウントあたりの ListSnapshotBlocks リクエスト数	サポートされている各リージョン: 50/秒	いいえ	アカウントあたりの許容される ListSnapshotBlocks リクエストの最大数。
アカウントあたりの「ボリュームのコピー」オペレーションの最大同時実行数	サポートされている各リージョン：5	あり	このリージョンでアカウントごとに許可される「ボリュームのコピー」オペレーションの最大同時実行数。

名前	デフォルト	引き上げ可能	説明
リージョンごとの同時ボリューム作成リクエスト全体にわたるボリューム初期化のプロビジョンドレート	サポートされている各リージョン: 5,000	あり	このリージョンの同時ボリューム作成リクエスト全体にわたってリクエストできる、ボリューム初期化の最大集計プロビジョンドレート (MiB/秒)。
アカウントあたりの PutSnapshotBlock リクエスト	us-east-1: 5,000/秒 us-east-2: 5,000/秒 us-west-2: 5,000/秒 ap-southeast-1: 5,000/秒 eu-west-1: 5,000/秒 他のサポートされている各リージョン: 1,000/秒	あり	アカウントあたりの許容される PutSnapshotBlock リクエストの最大数。
スナップショットごとの PutSnapshotBlock リクエスト数	サポートされている各リージョン: 1,000/秒	いいえ	スナップショットごとの許容される PutSnapshotBlock リクエストの最大数。

名前	デフォルト	引き上げ可能	説明
リージョンあたりのスナップショット	サポートされている各リージョン: 100,000	あり	リージョンあたりのスナップショットの最大数。
アカウントあたりの保留中の StartSnapshot スナップショット	サポートされている各リージョン: 100	いいえ	StartSnapshot API を使用して作成できる、アカウントあたりの保留中のスナップショットの最大数。
アカウントごとの StartSnapshot リクエスト数	サポートされている各リージョン：10/秒	いいえ	アカウントあたりの許容される StartSnapshot リクエストの最大数。

名前	デフォルト	引き上げ可能	説明
コールド HDD (sc1) ボリュームのストレージ (TiB)	us-east-1: 50 us-east-2: 50 us-west-1: 50 us-west-2: 50 ap-northeast-1: 50 ap-northeast-2: 50 ap-south-1: 50 ap-southeast-1: 50 ap-southeast-2: 50 ca-central-1: 50 eu-central-1: 50 eu-north-1: 50 eu-west-1: 50 eu-west-2: 50 eu-west-3: 50 sa-east-1: 50	あり	このリージョンの Cold HDD (sc1) ボリューム全体でプロビジョニングできるストレージの最大集計量 (TiB)。

名前	デフォルト	引き上げ可能	説明
	他のサポートされている各リージョン: 300		

名前	デフォルト	引き上げ可能	説明
汎用 SSD (gp2) ボリュームのストレージ (TiB)	us-east-1: 50 us-east-2: 50 us-west-1: 50 us-west-2: 50 ap-northeast-1: 50 ap-northeast-2: 50 ap-south-1: 50 ap-southeast-1: 50 ap-southeast-2: 50 ca-central-1: 50 eu-central-1: 50 eu-north-1: 50 eu-west-1: 50 eu-west-2: 50 eu-west-3: 50 sa-east-1: 50	あり	このリージョンの汎用 SSD (gp2) ボリューム全体でプロビジョニングできるストレージの最大集計量 (TiB)。

名前	デフォルト	引き上げ可能	説明
	他のサポートされている各リージョン: 300		

名前	デフォルト	引き上げ可能	説明
汎用 SSD (gp3) ボリュームのストレージ (TiB)	af-south-1: 300 ap-east-1: 300 ap-east-2: 300 ap-northeast-3: 300 ap-south-2: 300 ap-southeast-4: 300 ap-southeast-5: 300 ap-southeast-6: 300 ap-southeast-7: 300 ca-west-1: 300 eu-central-2: 300 eu-south-1: 300 eu-south-2: 300 il-central-1: 300 mx-central-1: 300	あり	このリージョンの汎用 SSD (gp3) ボリューム全体でプロビジョニングできるストレージの最大集計量 (TiB)。

名前	デフォルト	引き上げ可能	説明
	他のサポートされている各リージョン: 50		

名前	デフォルト	引き上げ可能	説明
マグネティック (standard) ボリュームのストレージ (TiB)	us-east-1: 50 us-east-2: 50 us-west-1: 50 us-west-2: 50 ap-northeast-1: 50 ap-northeast-2: 50 ap-south-1: 50 ap-southeast-1: 50 ap-southeast-2: 50 ca-central-1: 50 eu-central-1: 50 eu-north-1: 50 eu-west-1: 50 eu-west-2: 50 eu-west-3: 50 sa-east-1: 50	あり	このリージョンのマグネティック (スタンダード) ボリューム全体でプロビジョニングできるストレージの最大集計量 (TiB)。

名前	デフォルト	引き上げ可能	説明
	他のサポートされている各リージョン: 300		

名前	デフォルト	引き上げ可能	説明
プロビジョンド IOPS SSD (io1) ボリュームのストレージ (TiB)	us-east-1: 50 us-east-2: 50 us-west-1: 50 us-west-2: 50 ap-northeast-1: 50 ap-northeast-2: 50 ap-south-1: 50 ap-southeast-1: 50 ap-southeast-2: 50 ca-central-1: 50 eu-central-1: 50 eu-north-1: 50 eu-west-1: 50 eu-west-2: 50 eu-west-3: 50 sa-east-1: 50	あり	このリージョンのプロビジョンド IOPS SSD (io1) ボリューム全体でプロビジョニングできるストレージの最大集計量 (TiB)。

名前	デフォルト	引き上げ可能	説明
	他のサポートされている各リージョン: 300		
プロビジョンド IOPS SSD (io2) ボリュームのストレージ (TiB)	サポートされている各リージョン: 20	可能	このリージョンのプロビジョンド IOPS SSD (io2) ボリューム全体でプロビジョニングできるストレージの最大集計量 (TiB)。

名前	デフォルト	引き上げ可能	説明
スループット最適化 HDD (st1) ボリュームのストレージ (TiB)	us-east-1: 50 us-east-2: 50 us-west-1: 50 us-west-2: 50 ap-northeast-1: 50 ap-northeast-2: 50 ap-south-1: 50 ap-southeast-1: 50 ap-southeast-2: 50 ca-central-1: 50 eu-central-1: 50 eu-north-1: 50 eu-west-1: 50 eu-west-2: 50 eu-west-3: 50 sa-east-1: 50	あり	このリージョンのスループット最適化 HDD (st1) ボリューム全体でプロビジョニングできるストレージの最大集計量 (TiB)。

名前	デフォルト	引き上げ可能	説明
	他のサポートされている各リージョン: 300		
コールド HDD (sc1) ボリュームのストレージ変更 (TiB)	サポートされている各リージョン: 500	あり	このリージョンの Cold HDD (sc1) ボリューム全体のボリューム変更でリクエストできるストレージの最大集計量 (TiB)。
汎用 SSD (gp2) ボリュームのストレージ変更 (TiB)	サポートされている各リージョン: 500	あり	このリージョンの汎用 SSD (gp2) ボリューム全体のボリューム変更でリクエストできるストレージの最大集計量 (TiB)。
汎用 SSD (gp3) ボリュームのストレージ変更 (TiB)	サポートされている各リージョン: 500	あり	このリージョンの汎用 SSD (gp3) ボリューム全体のボリューム変更でリクエストできるストレージの最大集計量 (TiB)。
マグネティック (standard) ボリュームのストレージ変更 (TiB)	サポートされている各リージョン: 500	あり	このリージョンのマグネティック (スタンダード) ボリューム全体のボリューム変更でリクエストできるストレージの最大集計量 (TiB)。

名前	デフォルト	引き上げ可能	説明
プロビジョンド IOPS SSD (io1) ボリュームのストレージ変更 (TiB)	サポートされている各リージョン: 500	あり	このリージョンのプロビジョンド IOPS SSD (io1) ボリューム全体のボリューム変更でリクエストできるストレージの最大集計量 (TiB)。
プロビジョンド IOPS SSD (io2) ボリュームのストレージ変更 (TiB)	サポートされている各リージョン: 20	可能	このリージョンのプロビジョンド IOPS SSD (io2) ボリューム全体のボリューム変更でリクエストできるストレージの最大集計量 (TiB)。
スループット最適化 HDD (st1) ボリュームのストレージ変更 (TiB)	サポートされている各リージョン: 500	あり	このリージョンのスループット最適化 HDD (st1) ボリューム全体のボリューム変更でリクエストできるストレージの最大集計量 (TiB)。
コピー先リージョンあたりの時間ベースのスナップショットコピースループット	サポートされている各リージョン: 2,000	あり	送信先リージョンあたりの時間ベースのスナップショットコピーオペレーションのアカウントレベルの最大スループット (MiB/秒)。

考慮事項

- クォータは時間の経過とともに変化する可能性があります。Amazon EBS は、各リージョン内のプロビジョニングされたストレージと IOPS の使用状況を常にモニタリングしており、使用状況に応じて、リージョンごとにクオータを自動的に引き上げる場合があります。使用状況に基づいて Amazon EBS がクオータを自動的に引き上げますが、必要に応じてクォータの引き上げをリクエストすることもできます。たとえば、米国東部 (バージニア北部) で現在のクォータよりも多くの gp3 ストレージを使用する場合、使用予定より前にそのリージョンのボリュームタイプのクォータ引き上げをリクエストできます。
- 送信先リージョンあたりの同時スナップショットコピーのクォータは Service Quotas では調整できません。ただし、このクォータの引き上げをリクエストするには、 AWS サポートにお問い合わせください。
- IOPS 変更とストレージ変更クォータは、同時に変更できるボリュームの現在の合計値 (クォータに応じてサイズまたは IOPS) に適用されます。現在の値 (サイズまたは IOPS) を合計してクォータまでのボリュームに対して、同時に変更リクエストを行うことができます。たとえば、プロビジョンド IOPS SSD (io1) ボリュームの IOPS 変更 クォータが `50,000` の場合、現在の IOPS の合計が `50,000` 以下であれば、任意の数の `io1` ボリュームに対して同時 IOPS 変更リクエストを行うことができます。3 つの `io1` ボリュームにそれぞれ `20,000` IOPS がプロビジョニングされている場合、2 つのボリュームの IOPS 変更を同時にリクエストできます (`20,000 * 2 < 50,000`)。3 番目のボリュームに対して同時 IOPS 変更リクエストを送信すると、クォータを超過してリクエストは失敗します (`20,000 * 3 > 50,000`)。
- Amazon EBS には、インスタンス起動リクエストあたりの EBS ボリューム数に対して、以下の調整不可能な制限があります。
  - `2500` – `us-east-1`、`us-west-2`、`eu-west-1`、`ap-northeast-1`
  - `500` — その他のすべてのリージョン

  この制限は、ユーザーが行うインスタンス起動リクエスト、およびユーザーに代わって Amazon EMR などの AWS のサービスによって行われるインスタンス起動リクエストに適用されます。この制限を超えたためにインスタンス起動リクエストが失敗した場合は、起動リクエストの EBS ボリューム設定を調整して、ボリュームの数を制限未満に減らします。または、テクニカルアカウントマネージャー (TAM) と協力して、クラスターを起動するための他のオプションを確認してください。

Amazon EBS ユーザーガイドのドキュメント履歴

次の表は、Amazon EBS のドキュメントのリリースについての説明をまとめたものです。

変更	説明	日付
Elastic Volumes の変更の更新	前の Elastic Volumes の変更が完了するとすぐに、24 時間以内に最大 4 回、新しい Elastic Volumes の変更を開始できるようになりました。	2026 年 1 月 15 日
EBS ボリュームのごみ箱	ごみ箱を使用すると、Amazon EBS ボリュームを偶発的な削除から保護できます。	2025 年 11 月 20 日
EBS ボリュームのコピー	同じアベイラビリティーゾーン内に Amazon EBS ボリュームのインスタントなポイントインタイムコピーを作成できます。	2025 年 10 月 14 日
gp3 ボリュームに関する更新	サイズが最大 64 TiB、IOPS が最大 80,000、スループットが最大 2,000 MiB/秒の汎用 SSD (gp3) ボリュームをプロビジョニングできるようになりました。	2025 年 9 月 26 日
io2 ボリュームのレイテンシーに関するガイダンスの更新	io2 Block Express ボリュームは、16KiB の I/O オペレーションで平均 500 マイクロ秒未満のレイテンシーを実現するように設計されています。io2Block Express ボリュームでは、汎用ボリュームと比較して外れ値のレイテ	2025 年 9 月 26 日

	ンシーも改善され、800 マイクロ秒を超える I/O の頻度が 10 分の 1 以下に減少します。	
レイテンシーインジェクション	AWS FIS レイテンシーインジェクションアクションを使用して、Amazon EBS ボリュームの I/O レイテンシーの増加をシミュレートします。	2025 年 9 月 16 日
io2 リージョンの拡張	プロビジョンド IOPS SSD (io2) ボリュームが、すべての商用および AWS GovCloud (US) リージョンで利用可能になりました。	2025 年 7 月 22 日
ローカルゾーン内のローカルスナップショット	ローカルゾーンにローカルスナップショットを作成できるようになりました。	2025 年 5 月 15 日
ボリューム初期化の Amazon EBS プロビジョンドレート	Amazon EBS ボリュームの作成時にボリュームの初期化レートを指定して、予測可能な時間内に完全に初期化されるようにできるようになりました。	2025 年 5 月 6 日
Amazon Data Lifecycle Manager の VPC エンドポイント	VPC と Amazon Data Lifecycle Manager 間のプライベート接続を確立するには、インターフェイス VPC エンドポイントを作成します。	2025 年 2 月 28 日
時間ベースの AMI コピー	EBS-backed AMI コピーオペレーションの完了期間をリクエストして、AMI コピーが特定の期間内に完了するようにできるようになりました。	2025 年 2 月 25 日

フルスナップショットサイズ	Amazon EC2 コンソールと を使用して、Amazon EBS スナップショットのフルサイズを表示できるようになりました AWS CLI。	2025 年 2 月 11 日
Amazon Data Lifecycle Manager での IPv6 のサポート	Amazon Data Lifecycle Manager で、IPv4 トラフィックと IPv6 トラフィックの両方をサポートするデュアルスタックエンドポイントが提供されるようになりました。	2025 年 2 月 7 日
ごみ箱での IPv6 のサポート	ごみ箱で、IPv4 と IPv6 トラフィックの両方をサポートするデュアルスタックエンドポイントが提供されるようになりました。	2024 年 12 月 19 日
専用ローカルゾーン内のローカルスナップショット	専用ローカルゾーンにローカルスナップショットを作成できるようになりました。	2024 年 12 月 16 日
AWSDataLifecycleManagerServiceRole AWS 管理ポリシーが更新されました	AWSDataLifecycleManagerServiceRole AWS 管理ポリシーが更新され、`ec2:DescribeAvailabilityZones` アクションのアクセス許可が追加されました。	2024 年 12 月 16 日

EBS スナップショットのブロックパブリックアクセスに関する宣言型ポリシー	宣言型ポリシーを使用して、スナップショットのブロックパブリックアクセスに関するアカウントレベルの設定を、複数のリージョンとアカウントに同時に適用できるようになりました。詳細については、「AWS Organizations IAM ユーザーガイド」の「管理されたポリシー」を参照してください。	2024 年 12 月 1 日
時間ベースのスナップショットコピー	スナップショットコピーオペレーションの完了期間をリクエストして、スナップショットコピーが特定の期間内に完了するようにできるようになりました。	2024 年 11 月 26 日
ごみ箱の除外タグ	リージョンレベルの保持ルールに除外タグを追加して、特定のタグを持つリソースを除外できるようになりました。	2024 年 11 月 19 日
CloudFormation ごみ箱のサポート	を使用してごみ箱の保持ルールを作成および管理できるようになりました AWS CloudFormation。	2024 年 11 月 18 日
Amazon EBS の詳細なパフォーマンス統計	Amazon EBS NVMe ブロックデバイスは、Nitro ベースの Amazon EC2 インスタンスにアタッチされた Amazon EBS ボリュームに関するリアルタイムかつ高解像度の I/O パフォーマンス統計を提供します。	2024 年 11 月 12 日

Amazon EBS ボリューム用の新しい CloudWatch メトリクス	VolumeAvgReadLatency 、VolumeAvgWriteLatency 、VolumeIOPSExceededCheck 、および VolumeThroughputExceededCheck Amazon CloudWatch メトリクスを使用して、ボリュームのパフォーマンスをモニタリングできるようになりました。	2024 年 10 月 30 日
Amazon Data Lifecycle Manager のデフォルトポリシーをアカウント全体で有効化	CloudFormation StackSets を使用して、 AWS 組織全体または特定の AWS アカウントで Amazon Data Lifecycle Manager のデフォルトポリシーを有効にできます。	2024 年 4 月 26 日
AWSDataLifecycleManagerSSMFullAccess AWS 管理ポリシー	AWSSystemsManagerSAP-CreateDLMSnapshotForSAPHANA SSM ドキュメントを使用した SAP HANA で、アプリケーション整合性のあるスナップショットをサポートするようにポリシーを更新しました。	2023 年 11 月 17 日
VolumeStalledIOCheck メトリクス	過去 1 分間にストールした IO チェックに、ボリュームが合格していたか不合格であったかは、VolumeStalledIOCheck メトリクスを使用して確認できます。	2023 年 11 月 16 日

Amazon Data Lifecycle Manager のデフォルトポリシー	Amazon EBS スナップショットと EBS で動作する AMI 用に、Amazon Data Lifecycle Manager のデフォルトポリシーを作成し、リージョン内のすべてのボリュームとインスタンスをバックアップできるようになりました。	2023 年 11 月 16 日
Amazon EBS スナップショットのロック	Amazon EBS スナップショットをロックして、偶発的または悪意のある削除から保護したり、WORM 形式で一定期間保存したりできます。	2023 年 11 月 15 日
スナップショットのパブリックアクセスのブロック	パブリックに共有されることを防止するために、スナップショットのパブリックアクセスをブロックできるようになりました。	2023 年 11 月 9 日
Amazon Data Lifecycle Manager の事前スクリプトと事後スクリプト	Amazon Data Lifecycle Manager のスナップショットポリシーで、事前スクリプトと事後スクリプトを使用して、アプリケーション整合性のあるスナップショットのライフサイクルを自動化できるようになりました。	2023 年 11 月 7 日
NVMe 予約	マルチアタッチ対応の io2 ボリュームは、業界標準のストレージフェンシングプロトコルのセットである NVMe 予約をサポートします。	2023 年 9 月 18 日

Amazon EBS での障害テスト	AWS FIS を使用して、EBS ボリュームとそれがアタッチされているインスタンス間の I/O を一時的に停止し、ワークロードが I/O 中断を処理する方法をテストします。	2023 年 1 月 27 日
ごみ箱の保持ルールのロック	保持ルールをロックすることで、偶発的な、あるいは悪意のある変更や削除から保護できます。	2022 年 11 月 23 日
[Condition keys for Recycle Bin] (ごみ箱の条件キー)	ごみ箱リクエストのアクセスをフィルタリングするために `rbin:RequestResourceType` と `rbin:ResourceTypeの条件キーを使用することができます。	2022 年 6 月 14 日
io2 Block Express ボリューム	io2 Block Express ボリュームのサイズとプロビジョンド IOPS を変更し、高速スナップショット復元のために有効にできます。	2022 年 5 月 31 日
AMI のごみ箱	ごみ箱を使用すると、誤って削除した AMI を復元できます。	2022 年 2 月 3 日
Amazon EBS スナップショットのごみ箱	Amazon EBS スナップショットのごみ箱は、誤って削除したスナップショットを復元できるスナップショット復元機能です。	2021 年 11 月 29 日

Amazon EBS Snapshots Archive	Amazon EBS Snapshots Archive は、アクセス頻度の低いスナップショットを低コストで長期保存するために使用できる新しいストレージ階層です。	2021 年 11 月 29 日
Amazon Data Lifecycle Manager の AMI の廃止サポート	Amazon Data Lifecycle Manager EBS-backed AMI ポリシーは、AMI を非推奨にすることができます。AWS DataLifecycleManagerServiceRoleForAMIManagement AWS 管理ポリシーが更新され、この機能がサポートされました。	2021 年 8 月 23 日
Amazon Data Lifecycle Manager の CloudWatch メトリクス	Amazon CloudWatch を使用して、Amazon Data Lifecycle Manager のポリシーをモニタリングできます。	2021 年 7 月 28 日
EBS ダイレクト API の CloudTrail データイベント	ListSnapshotBlocks、ListChangedBlocks、GetSnapshotBlockおよびPutSnapshotBlock の API は、CloudTrail 内のデータイベントをログに記録することができます。	2021 年 7 月 27 日
io2 Block Express ボリューム	io2 Block Express ボリュームが一般利用可能になりました。	2021 年 7 月 19 日

Amazon EBS local snapshots on Outposts	Outposts の Amazon EBS ローカルスナップショットを使用して、Outpost のボリュームのスナップショットを、その Outpost に置かれた Amazon S3 内でローカルに保存できるようになりました。	2021 年 2 月 4 日
io2 ボリューム用マルチアタッチのサポート	Amazon EBS マルチアタッチのプロビジョンド IOPS SSD (io2) ボリュームを有効にできるようになりました。	2020 年 12 月 18 日
Amazon Data Lifecycle Manager	Amazon Data Lifecycle Manager を使用して、スナップショットを共有し、 AWS アカウント間でコピーするプロセスを自動化します。	2020 年 12 月 17 日
gp3 ボリューム	新しい Amazon EBS 汎用 SSD ボリュームタイプ。ボリュームを作成または変更するときに、プロビジョンド IOPS とスループットを指定できます。	2020 年 12 月 1 日
スループット最適化 HDD ボリュームサイズと Cold HDD ボリュームサイズ	スループット最適化 HDD (st1) ボリュームおよび Cold HDD (sc1) ボリュームのサイズは 125 GiB から 16 TiB です。	2020 年 11 月 30 日
Amazon Data Lifecycle Manager	Amazon Data Lifecycle Manager を使用して、EBS-backed AMI の作成、保持、削除を自動化できます。	2020 年 11 月 9 日

Amazon Data Lifecycle Manager	Amazon Data Lifecycle Manager ポリシーは、最大 4 つのスケジュールで設定できます。	2020 年 9 月 17 日
Amazon EBS 用プロビジョンド IOPS SSD (io2) ボリューム	プロビジョンド IOPS SSD (io2) ボリュームは、0.001% 以下の AFR で 99.999% のボリューム耐久性を提供するように設計されています。	2020 年 8 月 24 日
高速スナップショット復元	共有しているスナップショットに対して高速スナップショット復元を有効にすることができます。	2020 年 7 月 21 日
Amazon EBS マルチアタッチ	単一のプロビジョンド IOPS SSD (io1) ボリュームを、同じアベイラビリティーゾーンにある最大 16 の Nitro ベースのインスタンスにアタッチできるようになりました。	2020 年 2 月 14 日
Amazon EBS 高速スナップショット復元	EBS スナップショットで高速スナップショット復元を有効にして、スナップショットから作成された EBS ボリュームが作成時に完全に初期化され、プロビジョニングされたすべてのパフォーマンスが即座に提供されるようにすることができます。	2019 年 11 月 20 日

Amazon EBS マルチボリュームスナップショット	EC2 インスタンスにアタッチされている複数の EBS ボリューム間で、正確なポイントインタイムで、データ調整済みの Crash-consistent スナップショットを取得できます。	2019 年 5 月 29 日
デフォルトでの Amazon EBS 暗号化	リージョンでデフォルトで暗号化を有効にすると、そのリージョンで作成するすべての新しい EBS ボリュームは EBS 暗号化のデフォルトの KMS キー を使用して暗号化されます。	2019 年 5 月 23 日
スナップショットライフサイクルの自動化	Amazon Data Lifecycle Manager を使用して、EBS ボリュームをバックアップするスナップショットの作成と削除を自動化できます。	2018 年 7 月 12 日
アタッチされた EBS ボリュームに変更を行う	ほとんどの EC2 インスタンスにアタッチされたほとんどの EBS ボリュームでは、ボリュームをデタッチしたりインスタンスを停止したりせずに、ボリュームのサイズ、タイプ、IOPS を変更できます。	2017 年 2 月 13 日
間で暗号化された Amazon EBS スナップショットをコピーする AWS アカウント	AWS アカウント間で暗号化された EBS スナップショットをコピーできるようになりました。	2016 年 6 月 21 日

スループット最適化 HDD ボリュームタイプと Cold HDD ボリュームタイプ	スループット最適化 HDD (st1) と Cold HDD (sc1) ボリュームを作成できるようになりました。	2016 年 4 月 19 日
汎用 SSD ボリュームタイプ	汎用 SSD ボリュームは、さまざまなワークロードに対応できるコスト効率の高いストレージとして使用できます。これらのボリュームは、1 桁ミリ秒のレイテンシー、長時間にわたる 3,000 IOPS へのバースト機能、最大 3 IOPS/GiB のベースパフォーマンスを実現しています。汎用 SSD ボリュームのサイズ範囲は、1 GiB ~ 1 TiB です。	2014 年 6 月 16 日
Amazon EBS 暗号化	Amazon EBS 暗号化により、EBS データボリュームとスナップショットがシームレスに暗号化されるため、セキュアキー管理インフラストラクチャを構築および維持する必要がなくなります。EBS 暗号化サービスは、 AWS マネージドキーを使用してデータを暗号化することにより、保管中のデータのセキュリティを確保します。EC2 インスタンスをホストするサーバーで暗号化が行われるため、EC2 インスタンスと EBS ストレージとの間を移動するデータが暗号化されます。	2014 年 5 月 21 日

増分スナップショットコピー	インクリメンタルスナップショットコピーを実行できるようになりました。	2013 年 6 月 11 日
EBS スナップショットのコピー	スナップショットのコピーを使用して、データのバックアップを作成したり、新しい Amazon EBS ボリュームを作成したり、Amazon マシンイメージ (AMI) を作成したりすることができます。	2012 年 12 月 17 日

翻訳は機械翻訳により提供されています。提供された翻訳内容と英語版の間で齟齬、不一致または矛盾がある場合、英語版が優先します。